ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 155-160

Representation of obstacles in a Neural Network based Classifier System

N.R. Ball

Engineering Design Centre, Department of Engineering
University of Cambridge, Trumpington Street, Cambridge, U.K.

Abstract

This paper describes the application of a Neural Network based classifier system to the control of a simulated organism or animat in a simple obstacle avoidance task. A mechanism for concurrent exploration and exploitation of an environment based on Kohonen Feature Maps is described and some initial results in a simulated domain presented.

1. Introduction

The Hybrid Learning System (HLS) is an extension of the Classifier System Architecture that embeds neural networks within each classifier to enable environmental feedback to impinge directly upon the classifier population [2]. Experiments with HLS have focussed on the control of artificial organisms called animats that interact with simulated environments by activation of external classifiers [7]. The goal of the system is to enable the animat to achieve global problem domain objectives by learning how classifiers behave during the exploration of the domain and then evolving new classifiers that can optimize search in the domain state space to discover goal states. Earlier work focused on the associative learning of global objectives and unsupervised calibration of classifiers that emerged when the system was used to control animats exploring open mazes [3][4]. More recent experiments have introduced obstacles into these mazes and studied the resulting behaviour patterns.

2. The Hybrid Learning System

2.1 Embedding Neural Networks in a Classifier System

Classifier Systems are message-passing, rule-based, self-organizing systems employing a Genetic Algorithm (GA) as their main learning process [5]. The architecture of the Hybrid Learning System HLS (figure 1) is based on a classifier system with embedded Kohonen Feature Maps [6]. These maps are self-organizing

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 155-160

network structures that provide the system with its internal 'world model'. This model adapts with experience and the nature of that experience is changed through the activation of external and internal classifiers. The key concept is to use neural networks to represent the behaviour of each classifier in terms of its effects on the external environment in a similar way to Arkin's motor schemas [1]. These effects are encoded within the networks as state vectors with each vector representing the environment state before and after classifier activation. HLS's networks are self-organizing since no external critic is available to provide training examples. The self-organizing process calibrates classifiers by adapting their pre/post activation state vectors and eliminating redundant state data unaffected by classifier activation.

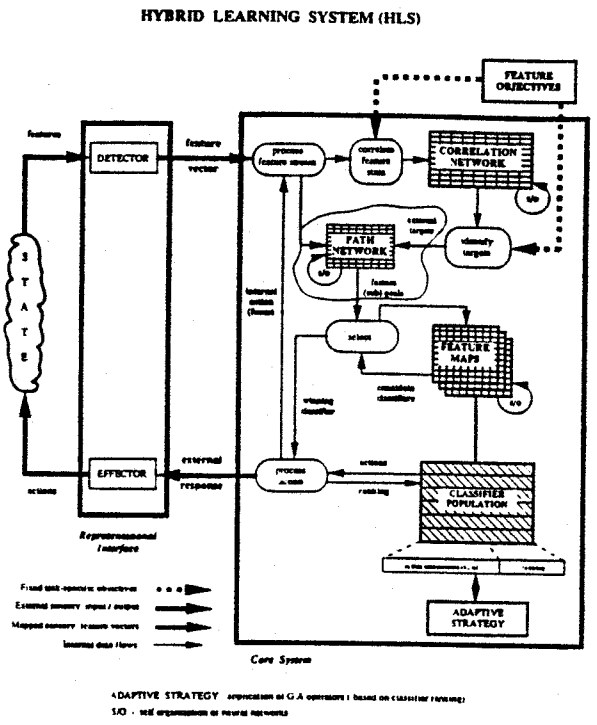

Figure 1. The top level HLS architecture

2.2 Exploring mazes

A simple simulated control problem is presented here to illustrate the execution cycle of HLS. An XY maze (32 * 30) is to be explored by an animat to reach a fixed food source at X=27, Y=23. In this scenario the animat is configured with two features - X location {X}, Y location {Y} - and five classifiers - +X, -X, +Y, -Y, <null> (implying no movement).

The basic execution cycle of the system is -

(1) detection of current feature signals {X,Y} for all locations within sensory range.
(2) selection of candidate classifiers for each external target {X=27, Y=23}.
(3) application of classifiers to problem environment to generate a new object state.
(4) self organization of classsifier feature maps based on the object state pre and post classifier application in (3).

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 155-160

The resulting behaviour is shown in figure 2 with the parts of the maze visited by the animat marked with gridlines. The system has calibrated the classifier population to move the object from the start point (S) to the target (G) despite the presence of obstacles. Calibration in this context means that the X/Y classifiers have been correctly associated with the dimensions that they effect.

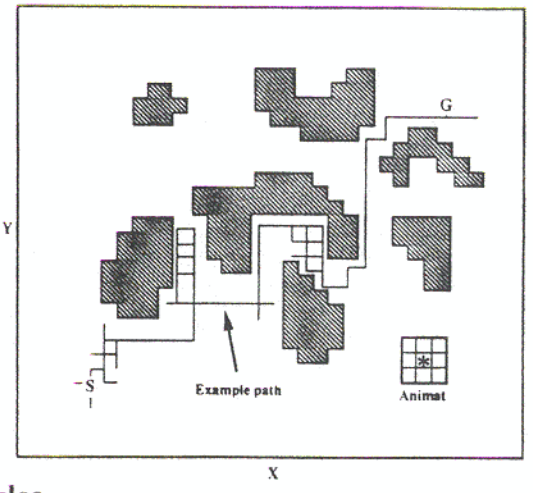

Figure 2 . Maze with 7 obstacles

2.3 Coping with large obstacles

The introduction of large obstacles into the domain can result in the animat becoming stuck in local minima since the classifier selection process is essentially a noisy hill climber working towards a fixed goal using strictly local information. The noise in this context is introduced by inter-classifier competition which is a measure of how efficient the system is in achieving its goals. Typical behaviour in a maze is shown in figure 3. Open mazes (no obstacles) permit accurate classifier calibration with rapidly decreasing competition indicating progress towards the goal The introduction of obstacles into the maze does not prevent the system achieving the goal but greatly increases the search time. A more serious hidden cost is the level of inter-classifier competition which decreases initially but then rises as obstacles are encountered. This in turn degrades each classifier's fitness ranking and reduces the effectiveness of the GA-based Adaptive Strategy (see figure 1).

One approach to overcoming this problem is the online decomposition of the problem domain space into a series of continous spaces that can be exploited by hill climbing. A path network has been introduced into HLS to capture obstacle boundary data and apply local search mechanisms to generate a series of achievable subgoals from this structure.

3. Representation of obstacle boundaries

3.1 Architecture

The architecture of the path network (figure 4) consists of a fixed size, 3 layer, 1D Kohonen Feature Map with a standard input / output layer configuration and an additional rehearsal layer. The purpose of the network is to capture nodes adjacent to

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 155-160

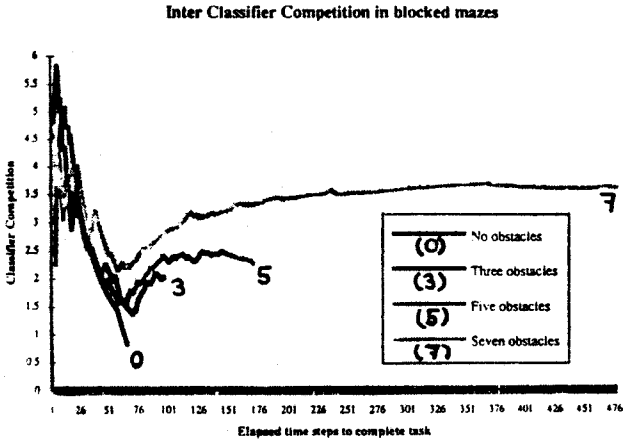

Figure 3. Inter-classifier competition dependency on obstacle configuration

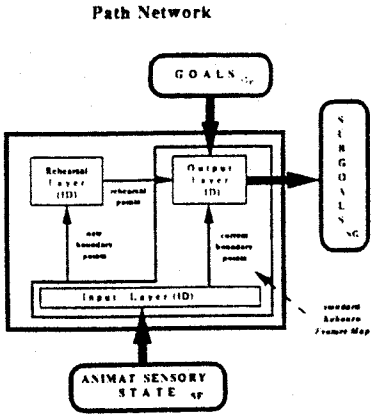

Figure 4. Path network architecture

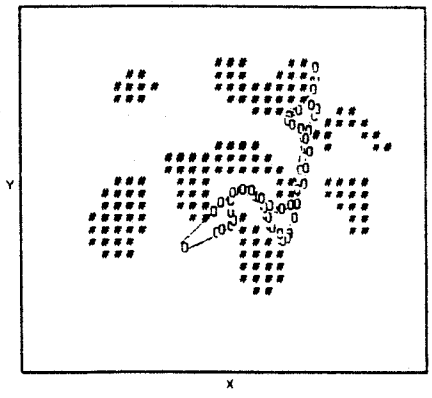

Figure 5. Path representation in maze

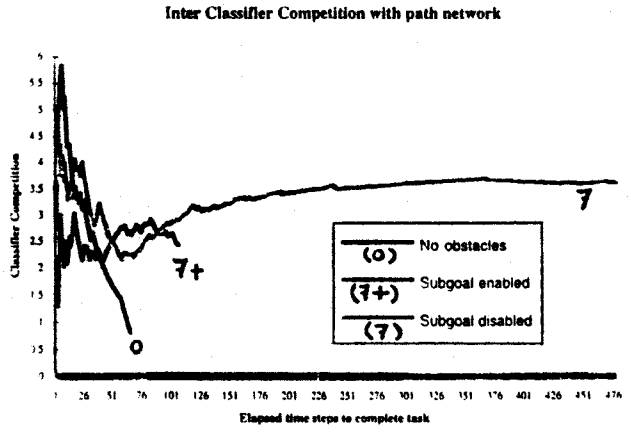

Figure 6. Inter-classifier competition with subgoaling

obstacle boundaries and use these as primary training data for the output layer. This data is also latched onto the rehearsal layer and replayed to provide continued reinforcement when the animat moves into free space. The addition of a frequency counter on each node provides a ranking measure based on frequency of visitation. The network can expand from a size of 2 to a size P_{max}.

3.2 Mechanism

Training

(1) extract current boundary points from the animat's sensory state **S** and place onto input layer as vector set **B**.

(2) find best matching node on the output layer **O** for each boundary point **b** in **B**
$\min_i \{\| \mathbf{o}_i - \mathbf{b}\|\} \ \forall \mathbf{o}_i$ in **O**

(3) increment frequency of visitation f counter of node i - $f_i := f_i + 1$

(4) perform adaptation of node i and physical neighbours -
$\mathbf{o}_i := \mathbf{o}_i + A^{123} * (\mathbf{o}_i - \mathbf{b})$
where A^{123} equals fixed adaptation rate associated with Kohonen Mexican Hat profile [6]. *Note* - there is no global attenuation of this profile.

(5) find best matching node on rehearsal layer **R** for each boundary point **b** in **B**
$dr = \min_i \{\| \mathbf{r}_i - \mathbf{b}\|\} \ \forall \mathbf{r}_i$ in **R**

(6) if dr ~ 0.0 (i.e new boundary point) then
(i) if network size = P_{max} then delete least recently visited node
(ii) insert new boundary point on layer **R** between points i and j given by -
$\min_i \{\| \mathbf{r}_i - \mathbf{b}\|\}$, $\min_{j \sim i} \{\| \mathbf{r}_j - \mathbf{b}\|\}$

Rehearsal

(1) for all $\mathbf{o}_i$ in **O** -
$\mathbf{o}_i := \mathbf{o}_i + A^1 * (\mathbf{o}_i - \mathbf{r}_i)$
where A^1 equals fixed adaptation rate associated with the first zone of the Kohonen Mexican Hat profile. *Note* - normally nearest neighbours.

Searching and subgoaling

The network can be applied to the subgoaling problem triggered whenever the animat moves adjacent to an obstacle. The basis algorithm employed is to attempt to subgoal onto output layer nodes that best match the global goal **G** and current state **S** attenuated by the frequency of visitation -

(1) insert goal **G** onto output layer **O** as node k between positions i and j given by
$\min_i \{\| \mathbf{o}_i - \mathbf{g}\|\}$, $\min_{j \sim i} \{\| \mathbf{o}_j - \mathbf{g}\|\}$
with $f_k = 0.5 * (f_i + f_j)$

(2) next subgoal $_{sg}$ is given by -
$\min_i \{ (\| \mathbf{o}_i - \mathbf{g}\| + \| \mathbf{o}_i - \mathbf{s}\|) * (1.00 + f_i) \} \ \forall \mathbf{o}_i$ in **O**
Note - the term f biases subgoaling away from frequently visited nodes.

(3) submit $_{sg}$ to Classifier feature Maps for hill climbing.

(4) remove node k from path network.

3.3 Behaviour

The type of structure generated by the path network is shown in figure 5 with path nodes denoted by 'O's. The effect on inter-classifier competition with the search mechanism enabled is shown in figure 6. The animat is able to achieve the global goal in approximately 126 steps compared compared with 476 steps with searching disabled. The profile of competition is also improved with a downward trend discernible after the initial calibration phase. Over repeated trials an animat applying subgoaling was able to outperform an animat without by a factor of 2.5.

4. Conclusions

Two limitations are apparent at this stage of the research - the need for sensory fields and the Euclidean representation of boundary nodes. The method requires that the animat has a sensory field of at least one unit range to enable the boundary of an obstacle to be identified. In contrast the Classifier Feature Maps are capable of driving the animat towards the goal with only animat state data. Obstacle boundaries are represented by their coordinates in physical space rather than by their relationship to the sensory field. This implies that large changes to the domain require considerable re-learning of the path network before subgoaling can be successfully applied. The main strengths of the new approach are that it is robust in capturing new boundary points and has scaled up to handling a moderate number of obstacles. In addition it generates stable fitness values in the Classifier Feature Maps which allows the GA-based Adaptive Strategy to function more effectively .

References

[1] R. Arkin, *Behaviour-Based Robot Navigation for Extended Domains.* Adaptive Behaviour, Vol. 1,No 2, pp 201-225, MIT press, 1992.
[2] N. Ball, *Cognitive Maps in Learning Classifier Systems.* Phd dissertation, University of Reading, 1991.
[3] N. Ball, *Reinforcement learning in Kohonen Feature Maps.* Proc. ICANN94, pp 663- 666, Springer Verlag, 1994.
[4] N. Ball, *Organizing an Animat's behavioural repertoires using Kohonen Feature Maps.* In J. Meyer, S. Wilson, D. Cliff (eds)., *From animals to animats 3.* MIT Press, 1994.
[5] J. Holland, K. Holyoak. R. Nisbett, P.Thagard, *Induction : Processes of inference, learning and discovery.* MIT Press, 1986.
[6] T. Kohonen, *Self Organization and Associative Memory.* Springer Verlag, 1984.
[7] S. Wilson, *Knowledge growth in an artificial animal.* Proc. 1st Int. Conference on Genetic Algorithms and Their Applications, pp 16-23. Lawrence Erlbaum, 1985.