A REINFORCEMENT LEARNING APPROACH BASED ON THE FUZZY MIN-MAX NEURAL NETWORK

Aristidis Likas

Department of Computer Science

University of Ioannina

P.O. Box. 1186 - GR 45110 Ioannina, Greece

tel/fax: +30-651-48131

e-mail: arly@cs.uoi.gr

Kvstas Blekas

Computer Science Division

Department of Electrical and Computer Engineering

National Technical University of Athens

157 73 Zographou, Athens, Greece

e-mail: kblekas@softlab.ece.ntua.gr

Correspondence: A. Likas

Abstract

The fuzzy min-max neural network constitutes a neural architecture that is based on hyperbox fuzzy sets and can be incrementally trained by appropriately adjusting the number of hyperboxes and their corresponding volumes. Two versions have been proposed each one suitable for supervised and unsupervised learning respectively. In this paper a modified approach is presented that is appropriate for reinforcement learning problems with discrete action space and is applied to the difficult task of autonomous vehicle navigation when no a priori knowledge of the enivronment is available. Experimental results indicate that the proposed reinforcement learning network exhibits superior learning behavior compared to conventional reinforcement schemes.

Keywords: Fuzzy min-max neural network, reinforcement learning, autonomous vehicle navigation.

A REINFORCEMENT LEARNING APPROACH BASED ON THE FUZZY MIN-MAX NEURAL NETWORK

Abstract

The fuzzy min-max neural network constitutes a neural architecture that is based on hyperbox fuzzy sets and can be incrementally trained by appropriately adjusting the number of hyperboxes and their corresponding volumes. Two versions have been proposed each one suitable for supervised and unsupervised learning respectively. In this paper a modified approach is presented that is appropriate for reinforcement learning problems with discrete action space and is applied to the difficult task of autonomous vehicle navigation when no a priori knowledge of the enivronment is available. Experimental results indicate that the proposed reinforcement learning network exhibits superior learning behavior compared to conventional reinforcement schemes.

Keywords: Fuzzy min-max neural network, reinforcement learning, autonomous vehicle navigation.

1 Introduction

According to the general framework of reinforcement learning, a system accepts inputs from the environment, selects and executes actions and receives a reinforcement signal r that is usually a scalar value rewarding or penalizing the selected actions. The basic approach to deal with such problems is based on the use of two networks [5]: the *action network* which provides the action to be executed at each step and the *evaluation network* which provides as output a prediction r_{pred} of the evaluation of the current state. The evaluation network is usually a feedforward network trained using on-line backpropagation with the training error specified through the method of temporal differences. The action network is again a feedforward network that for each input state provides a vector of action probabilities p_i $(i = 1, \ldots, K)$ (K distinct actions are assumed) with which the final action is selected. Consider that, for a given state, action j has been selected, r_{pred} is the output of the evaluation network and r is the corresponding reinforcement. Training is performed using on-line backpropagation with error based on $r - r_{pred}$. If $r - r_{pred} > 0$ then weights are modified to increase the probability p_j, otherwise they are modified to decrease the probability p_j.

In this letter, we present an approach to reinforcement learning problems with discrete action space where the fuzzy min-max neural network [1, 2] is employed as a model for the action network. The fuzzy min-max network is suitably adapted in order to be able to cope with the specific requirements imposed by the reinforcement learning framework. The proposed method constitutes an attempt to use a local learning technique (based on quantization of the input space into closed regions) for the action selection network and it is interesting to compare its performance against the approaches based on feedforward networks with Bernoulli output units. Of course, other on-line supevised local learning techniques could also have been employed and adapted to fit to the reinforcement framework.

2 The Proposed Action Selection Network

Fuzzy min-max neural networks [1, 2] consitute one of the many models of computational intelligence that have been developed in recent years from research efforts aiming at synthesizing neural networks and fuzzy logic.

The fuzzy min-max *classification* neural network [1, 3] is an on-line supervised learning classifier that is based on *hyperbox* fuzzy sets. A hyperbox constitutes a region in the pattern space that can be completely defined once the minimum and the maximum points along each dimension are given. Each hyperbox is associated with exactly one from the pattern classes and all patterns that are contained within a given hyperbox are considered to have full class membership. In the case where a pattern is not completely contained in any of the hyperboxes, a properly computed fuzzy membership function (taking values in $[0,1]$) indicates the degree to which the pattern falls outside of each of the hyperboxes. During operation, the hyperbox with the maximum membership value is selected and the class associated with the winning hyperbox is considered as the desicion of the network. Learning in the fuzzy min-max classification network is an on-line incremental *expansion-contraction* process that consists of partitioning the input space by creating and adjusting hyperboxes (the minimum and maximum points along each dimension) and also associating a class label to each of them. Details concerning the learning process can be found in [1]. An important issue is that there is only one parameter θ (maximum hyperbox size) that must be specified at the beginning of the learning process.

In order for the fuzzy min-max classification network to be employed as an action selection network in a reinforcement learning scheme (with discrete action space), a correspondence must be established between the notion of action and the notion of class, i.e. each action is treated with the same way as a class in the supervised case. There are also two main issues that have to be treated: The first one is related with the modifications to be performed in the case where the selected output is penalized, since we don't know which is actually the correct output (in contrast to the supervised case). The second is how to introduce randomness in the output selection process so that, in the case where an action has been penalized, alternative actions can be explored that may lead to rewarding states.

In the proposed scheme, both issues are resolved with the introduction of the notion of *random hyperbox*. If (for a specific input) a random hyperbox is selected (has maximum membership) the final action will be obtained with *uniform random selection* through the set of possible actions. For distinguishing pupроses, a non-random hyperbox will be called *deterministic*. Using the notion of the random hyperbox the learning process (expansion, overlap test, contraction) [1] of the classical fuzzy min-max network takes now the following form:

- In the case of reward ($r - r_{pred} > 0$),
  - if the action has been derived from a deterministic hyperbox then we proceed as in the classical fuzzy min-max case.
  - if the action has been derived from a random hyperbox then this hyperbox is marked deterministic and is associated with the corresponding rewarded action. Moreover, hyperbox overlap test followed by hyperbox contraction (if necessary) are performed.
- In the case of penalty ($r - r_{pred} < 0$),

– if the action has been derived from a deterministic hyperbox, then a new random hyperbox is created centered at the input point and consequently the conventional learning process takes place to adjust the parameters of the neighboring hyperboxes. It must be noted that the action associated with the initially selected hyperbox (which was penalized) does not change, only its volume is contracted due to the creation of the new random hyperbox.

– if the action has been derived from a random hyperbox, no learning takes place, since it is necessary to maintain stochasticity until a rewarding action has been discovered for the selected hyperbox.

Therefore, learning in the reinforcement case can be considered as a process of adding random hyperboxes that later become deterministic as learning proceeds. After an adequate number of steps it is expected that no random hyperboxes will exist any more. Random hyperboxes give the learning system the ability to explore the discrete output space to discover the best action. When such an action is found (according to the evaluation of the critic) it is assigned to the random hyperbox which now becomes deterministic.

3 Application to Autonomous Vehicle Navigation

We have tested our method in the problem of collision-free autonomous navigation of a vehicle in various uknown grounds. The objective is to train the fuzzy min-max action network to provide the proper driving commands as a response to the current state of the vehicle, so that it moves in a course without collisions [4].

The autonomous vehicle perceives its environment through the use of eight sensors. Four of them are located at the front and two at each side. Each sensor can detect the presence of an obstacle situated within a conic space in front of it and provides a measure of the distance of the obstacle. Specifically, the distance in the area tracked by each sensor is measured by a value in the range $[0, \ldots, 27]$. These eight integer values constitute the input state of the

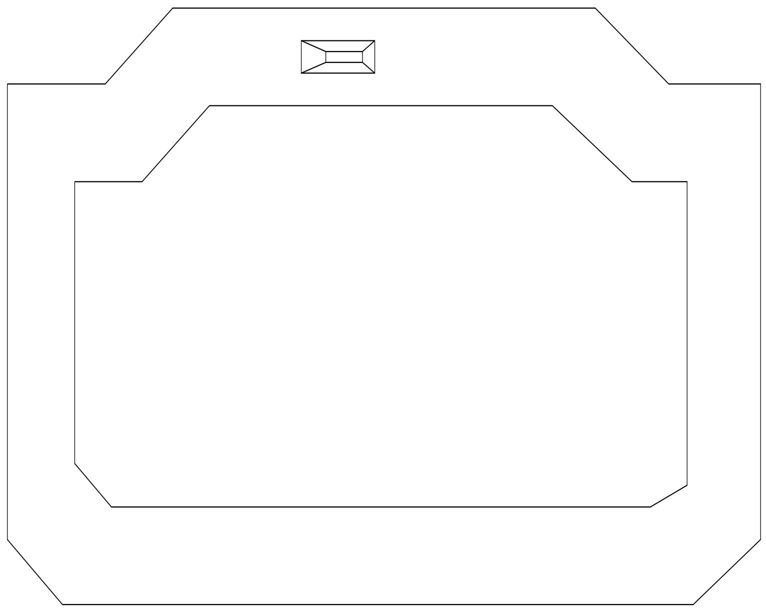

Figure 1: A typical ground

system. A sensor value close to 0 indicates that the sensor has detected an obstacle very near, while a value equal 27 declares the absence of obstacles. During navigation, the vehicle at each step performs one of the following five actions in response to the current state: *Ahead*, *30 degrees right*, *60 degrees right*, *30 degrees left*, *60 degrees left*. These actions correspond to the classes of the fuzzy min-max action network.

The evaluation of the vehicle's state is estimated in terms of the positions of obstacles as they are detected by the sensors. The scalar reinforcement signal r provided by the environment is real-valued ranging over $[0, 1]$, which indicates a graded transition from failure ($r = 0$) to success ($r = 1$). More specifically, r is computed as an average of the partial reinforcements r_i, corresponding to the four front sensors ($r_i = \xi_i/4$ with $\xi_i \in \{0, \ldots, 27\}$).

The performance of the proposed system has been investigated through computer simulation experiments. Each experiment consisted of a number of runs that differed only in the seed values for the random number generator. Each run consisted of a sequence of cycles, with each cycle beginning with the vehicle at the same initial state and ending with a failure signal. At the start of each run only one initial random hyperbox was considered with

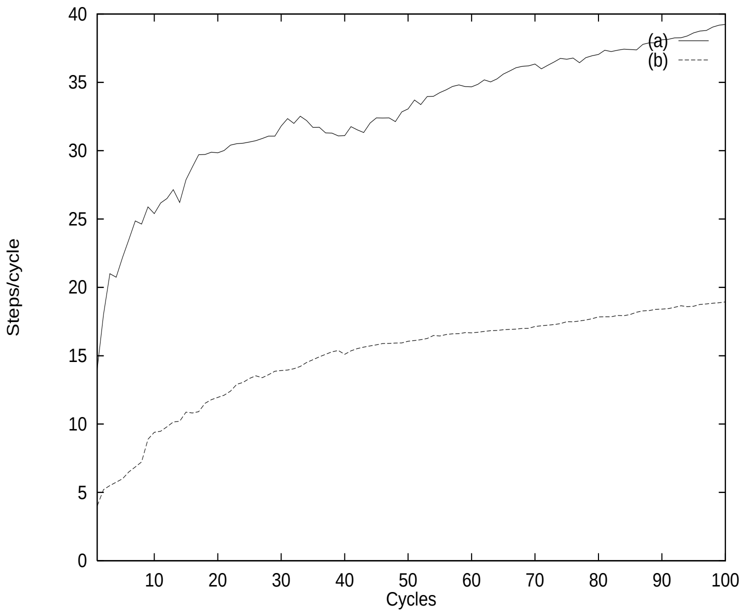

Figure 2: Training Performance

parameters specified by the sensor values (normalized to $[0, 1]$) corresponding to the initial vehicle position.

Statistical results of the effectiveness of learning during each run were obtained as follows. For smoothing purposes, at the end of each cycle an average value of the number of steps per cycle was computed by averaging over all cycles from the begining of the run up to that point. Finally, curves were plotted at the end of each experiment by averaging over all runs. This representation aims at giving an overall view of the progress of learning without being affected by random fluctuations. Several training grounds of varying difficulty have been explored. A typical one is presented in the Figure 1. In all the experiments the parameter θ was set equal to 0.115. The evaluation feedforward network contained one hidden layer with 6 units and one output unit. The value of the learning rate was 0.18 while the momentum rate was 0.009.

At early stages of the learning process the fuzzy min-max network created many random hyperboxes so as to search efficiently the input space and select the appropriate driving commands. As learning proceeds, new random hyperboxes are added at a very slow rate indicating that the network has discovered the appropriate class boundaries in the action space. In total, 350 fuzzy hyperboxes were created in the experiments concerning the depicted ground.

In order to evaluate the overall performance of our model in the given control task, we have compared it with the approach described in [4] where a feedforward action network was employed (with no hidden units) and Bernoulli output units. Figure 2 illustrates the training performance of the two approaches, where the learning curves (a) and (b) were obtained from experiments on the typical ground depicted in Figure 1 and correspond to the proposed and the conventional method respectively. This graphical representation illustrates the superiority of the proposed method in terms of learning speed and suggest that it is worth considering alternative networks and training algorithms in reinforcement learning schemes. Moreover, we expect that the performance of our approach can be further refined by assigning a vector of action probabilities to each hyperbox, which can be suitably adjusted during the learning process. This idea constitutes the subject of our current research.

References

[1] P. K. Simpson, "Fuzzy Min-Max Neural Networks-Part 1: Classification", IEEE Trans. on Neural Networks, vol. 3, no. 5, pp. 776-786, 1992.

[2] P. K. Simpson, "Fuzzy Min-Max Neural Networks-Part 2: Clustering", IEEE Trans. on Fuzzy Systems, vol. 1, no. 1, pp. 32-45, 1993.

[3] A. Likas, K. Blekas and A. Stafylopatis, "Application of the Fuzzy Min-Max Neural Network Classifier to Problems with Continuous and Discrete Attributes", in Proc. IEEE Workshop on Neural Networks for Signal Processing, pp. 163-170, Ermioni, Greece, 1994.

[4] D. Kontoravdis and A. Stafylopatis, "Reinforcement Learning Techniques for Autonomous Vehicle Control", Neural Network World, vol. 3-4, pp. 329-346, 1992.

[5] L. Lin, "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching", Machine Learning, vol. 8, pp. 293-321, 1992.