

Emulating real soccer

Ciprian Candea, Marius Oancea, Daniel Volovici
“Lucian Blaga” University of Sibiu, Dept. of Computer Science
4 Emil Cioran St, 2400 Sibiu, ROMANIA
Phone: +40-69-217928 Fax: 212716
email: {ccandea, mariuso, daniel}@cs.sibiu.ro
web: <http://hercules.sibiu.ro/~mariuso/ai/roboCup/>

Abstract. We use simulated soccer to study multi-agent learning. Each team member tries to learn from the corresponding human player in a real game. Following a unified approach, strategic and tactical behavior is learned synergistically by training a feed-forward neural network (ANN) with a modified back-propagation algorithm. It aims at decreasing the learning time and avoiding the local maximums. We tried to minimize the computation effort, as required in classic **back-propagation** (BKP) methods.

1. Introduction

Multi-agent (intelligent) systems (MAS) are software systems composed of several autonomous software agents running in a distributed environment. Besides the individual goal of each agent, global objectives, are established committing all or some agent groups to their completion. Agent coordination is a crucial aspect of multi-agent systems.

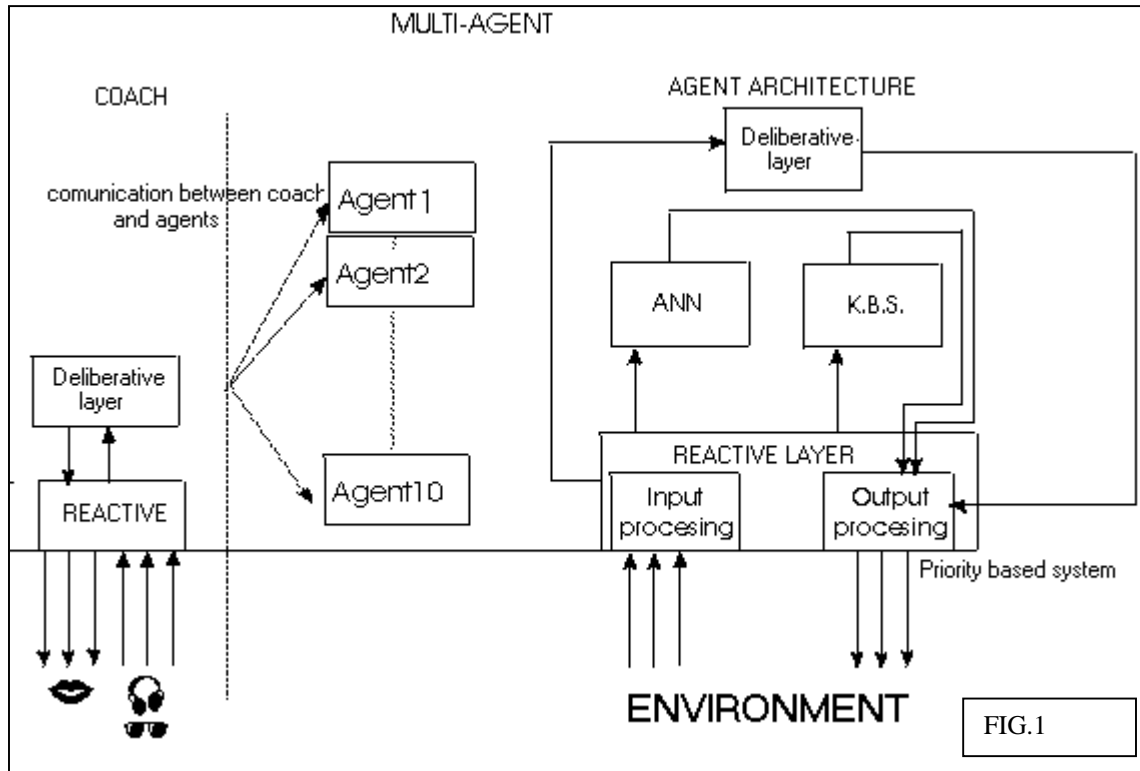
We use a multi-layer agent architecture. Such approaches have been proved by (Fischer, 1998) to be an efficient modeling method for behavioral abstraction levels in multi-agent architectures.

The architecture of an agent is structured in three layers: a) reactive (the agent acts accordingly to the environment changes perceived), b) deliberative (mixing two active components: a neural network - trained according to real game environment for strategic assignment - and a knowledge based system used for ball handling), c) cooperative planning (a layer distributed among all team members).

The remaining of this paper is organized as follows: Section 2 describes the layered architecture. Section 3 presents the neural network including the learning algorithm, based on a new variant of the BKP. At the end, Section 4, draws some preliminary conclusions.

2. Discussion

The overall architecture is presented in Fig.1. The multi-agent nature of the system is expressed through a coach coordinating ten (different) agents. The coach communicates with the agents through his voice and the agents' aural perception.



Each agent has a reactive layer (the goalkeeper has only this layer, therefore the system has only ten agents included in the MAS). This layer is responsible for the agent behavior when he controls the ball. Its inputs comprise all the agent sees and hears, updating its world perception. The outputs are actions upon the environment (based on an analytical approach).

The next layer has two components: an ANN and a KBS. The ANN determines the tactical behavior when the player is without the ball (i.e., movement, depending on the game phases). The KBS has the same role, but when the player controls the ball.

The last level is the most abstract one, being influenced by the coach, at "team level". Practically, that means that the individual behavior pattern of each agent is supplemented by a collective one, established by the global evaluation of the game through the coach.

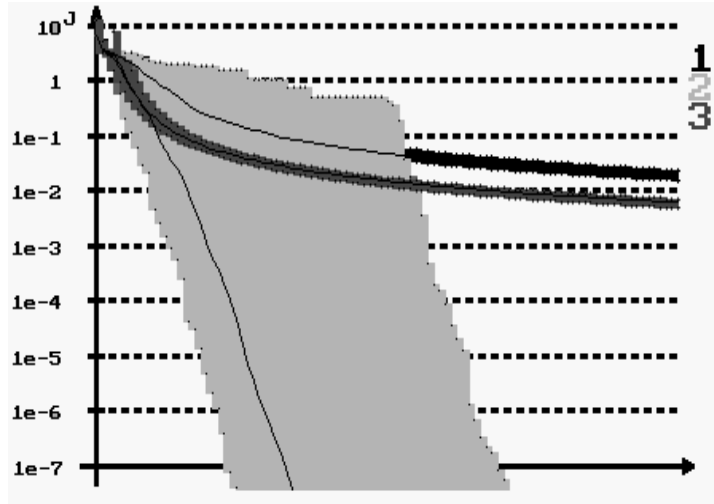
The actual output is determined by a selective mixing of the three levels through different priorities, the highest being given to the reactive level.

Thus, we expect to meet the requirements contained in the following quotation: "The individual player has to perform several behaviors one of which is selected depending on the current situation. It is difficult to find a simple method for learning these behaviors, definition of social behaviors [...] Alternative such as 'coordination by imitation', should be considered".

3. Neural network

The proposed method has a very good performance regarding the convergence speed and the (remarkable) advantage that it does not rely on any “empirically chosen” parameter (as step value), which is always a problem for BKP methods.

3.1 Method



The proposed way of modifying the learning rate ρ is:

- Compute an initial ρ and set $i = 1$. J_0 available from previous iteration.
 - Compute the weights' variation Δw using the conjugate gradients with restart formula
compute the weights $w = w_k + \rho \cdot \Delta w$
using w determine J_1
 - If $J_1 < J_0$ Repeat
 $i = i + 1$
compute the weights $w = w_k + (\rho \cdot u_i) \cdot \Delta w$
using w determine J_i
Until $J_i > J_{i-1}$
 $w_{k+1} := w_k + (\rho \cdot u_{i-1}) \cdot \Delta w$
 - Else Repeat
 $i = i + 1$
compute the weights $w = w_k + (\rho / u_i) \cdot \Delta w$
using w determine J_i
Until $J_i < J_0$
- $w_{k+1} = w_k + (\rho / u_i) \cdot w$

- proceed to the next iteration.

The u_i array may be the 2's powers array ($u_i = 2^i$), but we prefer the Fibonacci array (1,1,2,3,5,8, .. i.e. $u_i = u_{i-1} + u_{i-2}$) for its good results proved on the optimization theory.

The initial choosing of ρ at each iteration is very important: which ρ should be large enough to exit local minimums, but not too large to avoid repeating, at each iteration, the same step decreasing procedure. On simulations, we used the formula given in the

$$\rho(k+1) = \left(\frac{1}{2} + \xi\right) \cdot \rho_{mean}(k) \cdot (n_{iterat} + 1)$$

equation. ρ is a uniform random number within (0,1), n_{iterat} counts for how many iterations does the criterion J decrease with less than 1% and ρ_{mean} is the arithmetic mean of ρ 's values for the last 10 iterations (too large or too small values being eliminated). At the beginning of the learning process, ρ_{mean} is 0.1.

Results

We test the performance of the classical BKP method, BKP with Term Proportion, BKP with Term Proportion and Restart and Conjugate Gradient BKP against each other and the new BKP proposed method. For tests, we used 6 well-known problems: ♦number's parity, ♦bit counting, ♦multiplexor problem, ♦pattern recognition, ♦associative memory, ♦function emulation. Experimental results prove that in all cases the new BKP method gives better results. The range of the results obtained (for 10 different random initial values of the weights) for the multiplexor problem is presented in the figure (1 - classic BKP, 2 – new BKP method, 3 – Conjugate Gradient BKP).

In such a context we implemented a feed-forward neural network made up by three layers. The ANN has 20 inputs (two for each of the 19 players and the ball – because the goalkeepers have no role to play in this simplified strategy). The player inputs are composed of two bits enabling to represent four different vectors describing the location and speed of the players, related to each agent (that means, four distinct “tactical” components of the overall picture)

4. Conclusions

The proposed architecture is a trade-off between flexibility and reduced complexity. Its main features, representing our intention to experiment new approaches are: a) dividing the middle layer into a symbolic processing part and a subsymbolic one; b) training the neural network for each team they have to meet; c) training it also for each player; d) the high proportion of the coach instructions contribution in the agent behavior.

The implementation is based on a new method, better than the classic BKP for its shorter searching time and for its ability to escape from local minima.

References

1. Fischer, K. (1998). An Agent-Based Approach to Holonic Manufacturing Systems. In Camarinha-Matos, L.M., Afsarmanesh, A., and Marik, V. (Eds.), *Intelligent Systems for Manufacturing. Multi-Agent Systems and Virtual Organizations*. Kluwer Academic Publishers, Boston. p. 3-12
2. Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa. Robocup : The Robot World Cup Initiative
3. Peter Stone. Layered Learning in Multi-Agent System. PhD Thesis, December 1998
4. DARPA Neural Network Study (October, 1987-February, 1989). MIT Lincoln Lab
5. Learning internal representations by error propagation by Rumelhart, Hinton and Williams (1986).