Journal of Computer Science 1 (1): 28-30, 2005
ISSN 1549-3636

Seek of an Optimal Way by Q-Learning

[1]Y. Dahmani and [2]A. Benyettou
[1]Signals, Images and Speech Laboratory, Ibn Khaldoun University
Tiaret, B.P. 78 C.P. 14000, Algeria
[2]Data Processing Department, University of Sciences and Technology of Oran
B.P. 1505 M'Naouer 31000, Algeria

Abstract: In this article, we presented the Q-Learning training method which is a derivative of the reinforcement learning called sometimes training by penalty-reward. We illustrate this by an application to the mobility of a mobile in an enclosure closed on the basis of a starting point towards an unspecified arrival point. The objective is to find an optimal way optimal without leaving the enclosure.

Key words: Reinforcement Learning, Q-Learning, Exploration Phase, Exploitation Phase

INTRODUCTION

The inherent difficulty in the construction of a training database in the learning process represents an operational limit of certain type of training such as the supervised training [1, 2, 3].
The reinforcement learning is a possible alternative to define the training database by an operator. One of its advantages resides in the form of its training examples. They are triplets (input, output, quality), where the last component represents the utility to produce such " output " for such " input ". The examples of learning are generated here automatically during a phase known as of "exploration ". It is generally about a random exploration of the research space [4].

Reinforcement Learning: The reinforcement learning called sometimes learning with critic is a slightly supervised learning [5, 6].
Initially, the study consists in observing the training, not directly, but by the means of the behavior; in the second time, the apprentice does not make any more that to answer to an environmental antecedent, and operates on the environment (Fig. 1). In this model, with each interaction with his environment, the apprentice represented by the robot and characterized by his behavior B perceives the state S in which it is by the means of function I (S).
While basing itself on its perception of the state, the apprentice then chosen an action U among the whole available actions in this state according to a probability P. When this action is applied to the environment, the system changes state, the apprentice receives a reinforcement signal r by the reinforcement function R [7].
The objective of the reinforcement learning is thus to find the output of greater utility. The success of the application will depend on the quality of the function specifying the utility of a pair (input, output). This function of utility, usually called reinforcement function [2].

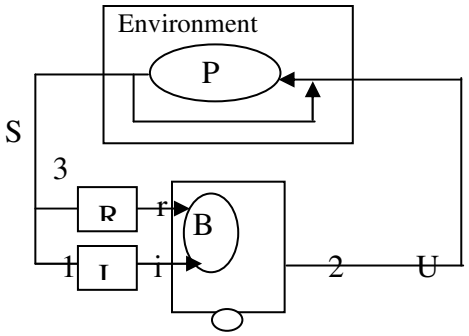

Fig. 1: Standard Model of the Reinforcement Learning

The Q-Learning: The Q-Learning was proposed for Markovien's problems decision, with discrete states and actions spaces. At each step of time an agent observes the vector of state x_t chooses and applies an action u_t. The system passes in state x_{t+1} and the agent receives a reinforcement $r(x_t,u_t)$.
The goal of the training is to find the policy of order which maximizes the sum of the future reinforcements.
For a given policy π, we note $u_t = \pi(x_t)$ the selected action.
The evaluation function π, noted V^{π}, is given by:

$$V^{\pi}(x) = E\left\{\sum_{t=0}^{\infty} \gamma^t r(x_t, \pi(x_t) / x_0 = x\right\}$$

The parameter γ, $0 \le \gamma < 1$, ensure the convergence of the sum. The optimal evaluation function V^{*}, corresponds to an optimal policy, checks the equation of optimality of Bellman :

$$V^* = \max_{u \in U_x} \left[r(x,u) + \gamma \sum_y P_{xy}(u) V^*(y) \right] = \max_{u \in U_x} Q^*(x,u)$$

where:

U_x = set of possible actions in the state x

P_{xy} = probability of passing from the state x to y by the action u

Q(x, y) represents the total reinforcement if the action u is selected in state x and if an optimal policy is selected then. It is called the quality function for a couple (state, action).

If the transitions probabilities $P_{xy}(u)$ and the law of the reinforcement r(x,u) are known, it is possible to find an optimal policy by using a dynamic programming algorithm.

Instead of using the evaluation function, Watkins proposed to estimate the function Q* by the function :

(x, u) → Q(x, u) which is updated with each transition by [8, 9] :

$$Q(x_t,u_t) \leftarrow Q(x_t,u_t) + \beta \left[r(x_t,u_t) + \gamma \max_{v \in U x_{t+1}} Q(x_{t+1},v) - Q(x_t,u_t) \right]$$

β is a training parameter which must tend towards 0 when t tends towards the infinite one.

Exploration/Exploitation: The generation of the learning base is done in parallel with the phase of exploration, the learning is incremental. This is why, when a representative base of learning was finally built, the learning is finished.

The optimal policy is obtained by choosing the action which in each state maximizes the function of quality called Greedy policy:

$$u = \arg \max_{v \in U_x} Q^*(x,v)$$

At the beginning of the learning the values Q(x,u) are not significant and the Greedy policy is not applicable.

To obtain a useful estimate of Q it is necessary to sweep and evaluate the whole of the possible actions for all the states; what one calls the phase of exploration; then an exploitation phase is started once the finished learning.

The Policy of Exploration/Exploitation PEE(x) can be selected according to Glorennec [7].

Pseudo-stochastic Method: The action with better value of Q has a probability P of being selected if not an action is selected randomly among all the possible actions in a given state.

Distribution of Boltzmann: The action u is selected with the probability :

$$P(u/x) = \frac{\exp(\frac{1}{T} Q(x,u))}{\sum_{v \in U} \exp(\frac{1}{T} Q(x,v))}$$

T is comparable with the temperature, gives the importance of the random factors. This parameter decrease in time.

Algorithm: After the choice of a policy of exploration/exploitation, the algorithm is held in the following way (Fig. 2):

* Observe the situation u
* Select u=PEE(x)
* Apply the action u and observe the new state
* Get the reinforcement r(x,u)
* Update Q(x,u)

Fig. 2 : The Q-Learning Algorithm

Example of Application: During this study, we applied this training to the problem of the displacement of a mobile in a matrix with 3 lines and 5 columns. The mobile has the choice between 4 possible actions (go down, go up, go right, go left) (Fig. 3).

S				
				A

Fig. 3 : Displacement Matrix of Mobile

The user chooses an unspecified starting point S and a point of arrival A of his choice. The mobile passes by two phases:

* A phase of exploration, where the mobile tries to move according to 4 possible choices' and with each action a reward or a punishment is generated according to whether one approaches the goal without leaving the matrix (Fig. 4).
* In the second phase which is the exploitation phase (Fig. 5), the mobile moves towards its point of arrival by using the best actions learned during the first phase with maximum qualities.

One needed 39 iterations for the phase of exploration and 6 only for the exploitation phase to go from the starting point (1st line, 1st column) towards the arrival point(3rd line, 5th column) for this simulation

As for the matrix of qualities, we can give an explanation. If the mobile robot is with the first line first column, the quality to go down is the best (q=1.0), as for the action to go up or go on the left (q=-0.9) is punished, they are thus actions to prevent whereas the first action (q=0), was not tested yet.

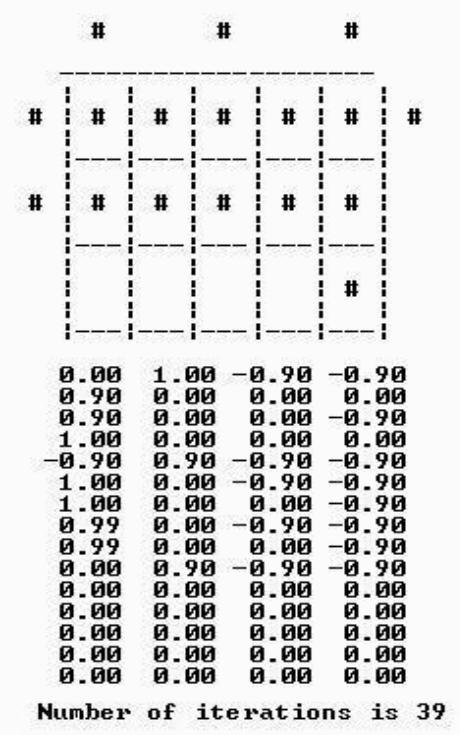

Fig. 4 : Exploration Phase

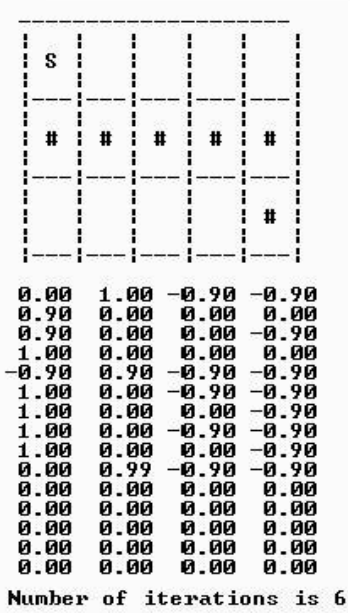

Fig. 5 : Exploitation Phase

The equation of the function of reinforcement is given by:

$$r=\begin{cases}+1 & \text{if } \Delta d<0 \text{ or we don't leave the matrix}\\ -1 & \text{otherwise}\end{cases}$$

where Δd=the distance variation towards the goal (actual_ditance – preceding_distance).

After having received the reinforcement (r=-1 or r=1), the variation of qualities is given by:

$$Q(m(t),u(t))\leftarrow Q(m(t),u(t))+\alpha\left[r-Q(m(t),u(t))\right]$$

where $\alpha = 0.9$

CONCLUSION

This training makes it possible progressively to obtain a base of training with the phase of exploration, which makes it possible to the apprentice to learn and perfect its behavior during the exploitation phase. Further work is needed and should be retained:

* It should be noticed that the reinforcement function plays a significant role in the process of training and that its formulation remains a point to be described with rigour.
* More especially as the use of an adaptive reinforcement can be used in order to mitigate all mow case of reinforcement or possible punishment.

REFERENCES

1. Jouffe, L, 1997. Training of Fuzzy Inference Systems by Reinforcement Methods. Application to the regulation of ambiance in a building of pork raising. Ph. D Thesis University of Rennes I, France.
2. Harmon, M.E, 1998. Reinforcement Learning. A Tutorial. W/LAACF 2241 Avionics Circle Wright Laboratory.
3. Beom, H.R. and H.S. Cho, 1995. A Sensor-Based Navigation for a Mobile Robot Using Fuzzy Reasoning and Genetic Algorithm. IEEE Transactions on Systems Man and Cybernetics, Vol. 25 80.
4. Touzet, C., 1999. The Reinforcement Learning in Connexionnisme and Applications C.J Masson, (editor). CESAR-ORNL, USA.
5. Glorennec, P.Y., L. Foulloy and A. Titli, 2003. The Reinforcement Learning, Application for Fuzzy Inference Systems. Fuzzy Order 2, Treated IC2, Ed Lavoisier.
6. Wang, L. and J.M. Mendel, 1991. Generating Fuzzy Rules By Learning From Examples. CH3019-7/91/0000-0263$01.00 IEEE.
7. Glorennec. P.Y., 1999. Algorithms of Optimization for Fuzzy Inference Systems : Application for Identification and Order. National Institute of Applied Sciences Rennes. Eds Hermès, France.
8. Garcia P., Zsigri, A., Guitton A., 2003. A Multicast Reinforcement Learning Algorithm for WDM Optical Networks. 7th Int. Conf. on Telecommunications-ConTEL. June 11-13, Zagreb, Croatia.
9. Mark, D.P., 2000. Reinforcement Learning in Situated Agents. Theoretical Problems and Practical Solutions. Lecture Notes in Artificial Intelligence 1812, pp. 84-102. Berlin, Germany.