International Journal of u- and e- Service, Science and Technology
Vol.8, No. 6 (2015), pp.331-340
http://dx.doi.org/10.14257/ijunesst.2015.8.6.33

A Novel Searching Algorithm based on Reinforcement Learning

[1]Anil Kumar Yadav and [2]A. K. Sachan

[1]Department of CSE
IFTM University India
[2]Department of CSE
RITS RGPV University Bhopal India
[1] aky125@gmail.com, [2]sachanak_12@yahoo.com

Abstract

We introduce an application-oriented reinforcement learning searching algorithm designed for problem with fast learning and capturing goal in less amount of time especially in robotics and games. The importance of game playing in machine learning is an exhaustive application of autonomous agent in real-world problem domain. In our previous published article represent that how autonomous agent learned through self-training and successful trained agent ready for execution [11].In this paper, we design and proposed a new application-oriented searching algorithm especially for game playing in grid world problem. In which first of all agents train all state and able to capture goal successfully. Reinforcement learning is a type of decision making system that takes decision on the basis of reward or penalty signal and learned from environment. Many games, there are no such things that follow fast learning as well as searching and genuine movement for each step. For every state action agent stored previous values in terms of q values in a look-up table. It helps for agent decision making capability during goal hitting or pray captured in the real-world game. In order to access and simulate new searching algorithms in mat lab and evaluated by comparison with different RL techniques [2, 11-12].

Keywords: *machine learning, environment, reinforcement learning, game playing, searching, agent*

1. Introduction

It is very difficult for researcher to simulate multiple agents based on self learning for game playing. In this view, it is an important and exciting field for designing real application-oriented reinforcement learning searching algorithm in the field of pray capturing, complex decision, robotics and optimization. In general most of the games being played based on supervised learning, while we are introducing reinforcement learning here. RL is acts as observer, it may a observe situation of next steps. Reinforcement learning also used in virtual neural network for resource allocation where reinforcement learning agent used for auto monitoring resource allocation action through neural network training [13]. Online adaptive optimal control based on self decision making as RL agent to solve horizon effect and non linear problem and neural network acts as a classifier for optimal control [14]. Reinforcement learning has ability to predict anything without prior knowledge of the system. It used as feedback predictor and decision maker for non liner discrete time system [15]. The best possible action towards gain the objective is in the form of left, right, up and down. Once the agent makes any stroke, the environment will give a response indicator, which is called reward. It is a decision making system based on self mechanism what to do and how to relate situations to performed action and maximize a numerical reward value. In this machine learning system, the agent [1, 8] is a decision making that takes deeds in an environment and takes

ISSN: 2005-4246 IJUNESST

reward or punishment for its actions in demanding to work out a difficulty. After review number of experiment and mistake, it would be trained the best policy. Sequence of deeds that exploit the entire reward and an environment is presented by a set of states in which an agent can recognize and receive deeds to modify. Knowledge provides capability to the agent operate in at the start unfamiliar environments and to happen to more proficient than its initial awareness unaccompanied might permit. Reinforcement learning describes in terms of a group of machine learning algorithms that try to find estimated solutions to stochastic chronological decision tasks with scalar manipulation criticism. The system can learns how to achieve that goal by number of trial and fault interactions with its environment *i.e.* RL is defined as a decision making system which take decision without prior knowledge of the system. Figure 1. Basic description of self learning in terms of reinforcement learning and how agent interact with given environment expressed in given figure.1.0 agent measure efficiency of reinforcement learning system through reward and basic goal of agent to increase commutative reward or future reward. Reinforcement learner system [5, 9] defined as, it contains five tuple implies [S, A, π, γ, VF], in which S is number of state of environment, A is possible deeds or actions performed by agent, π is applied policy and γ is discount rate at which agent learned during back tracking value function respectively. Recital of self learning based on RL for regular movement counted through rate of q [16-17].

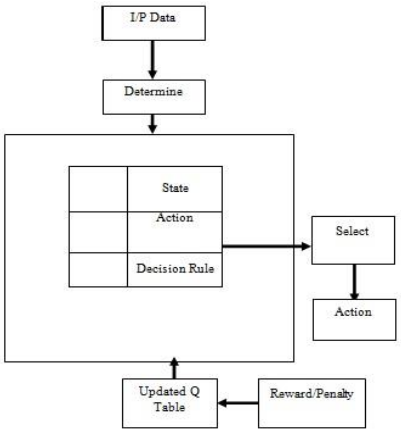


Figure 1. RL Decision Making System

Machine learning algorithms were developed to carry a variety of decision making responsibilities. A policy defines to the given value function of the expected reward signals can receives by agent. The state-action value function (s,a) under a policy π is defined as $V^{\pi}(s) = E_{\pi}\{R_t Is_t=s\}=E_{\pi}\{\sum_{k=0}^{\infty} r\gamma t\}$. This is the expected responses over the policy π opening from state S and Mathematical expectation is [3, 2].A insatiable process is always exploits and Exploration means it will takes action than the greedy action. The reason of exploration is to determine other actions which might be improved than the greedy action [10]. The e- insatiable technique is that performed both exploitation and exploration. It will take decision for the the greedy action and e selects a randomly action. In any decision system machine that contains four elements which is rewords, policy where environment is represented by a set of states and learning agent is a decision maker that is based on perception, it perceives and chooses an action within four movements for the system. In reinforcement learning used for acquire difficulty such as maze problem using Temporal Difference network (TDN). In the TDN methods agent is predicting time delayed rewards and it will be evaluate future rewards. Here cast worth utility is new task for calculate approximately of upcoming concert in its place of mixed commutative plunder [4][6].Respite of the study is prepared as follows: here examine associated effort

in segment 2. Our given approach is presented in part 3 and evaluate in fragment 4.lastly; part 5 concludes the research paper, generous and finale for outlook effort.

2. Related Work

There is credible work regarding self learning decision making for an agent through tails-and-error interaction. Ethan *et al.* [1] helps us for comprehensive survey on reinforcement learning and also motivated toward, how agent or RL system learn from environment and gives decision according input state action pair through reward and penalty without prior information about system. Most of the existing research work in reinforcement learning focus on improving the reward value and episodes in query based learning [2].But in our knowledge agent should be fast searching in less amount time with respect to improving discount rate and number of episodes. Existing work on [11] agent self learning is based on three main approaches: agent on training, agent on work and neural network classifier used for removing redundant information in look-up-tale. This helps agent for exact learning. For example [3] is a types of reinforcement learning techniques, [4] is based on greedy attribute selection, while [10] uses prediction based on effective integration of imitation learning and reinforcement learning by generating internal reward. The difference between our proposal and these previous works is not equal in the context of solutions tool like ANN [7], but more applicable in application-oriented approach for learning as well as searching. [5] is based on learning rules and their exceptions, while [6] is rule-based part of speech tagging. A combination of RL and ANNs has been applied to several problem such as [11-13].in these proposals, neural networks are used as decision helper or classifier. This paper contributes towards design and simulating application-oriented RL searching algorithm for fast learning and capturing efficient

3. Proposed RL-Searching Algorithm (RLSA)

This algorithm is the modified form of an earlier algorithm [2, 11] for learning and searching optimistic path without repetition in unknown environment. The earlier algorithm has a problem of learning many redundant paths therefore, that wasting huge amount of time and effort. This problem is rectified in the algorithm presented in this paper. Therefore, if there is a searching and capturing time constraint on testing then this algorithm will perform better by efficient learning more independent path in small amount of time. The purpose of an algorithm is to traverse all the states firstly with the help of look-up-table which contain approximate value of the number of further paths possible along a particular path.

The algorithm accepts some pseudo code as the input for training process of agent. Complexity of time and space is deliberate through the algorithm. In which RL agent moves in all possible direction and arrived through independent paths from the start state to the goal state. The output the algorithm is pray captured in less amount of time. The explanation of how the agents are trained is explained by considering the following program.

A. Algorithm: agent training for searching

I/P=Initial query, define possible state(s, a) and move (L R U D) O/P= Goal achieved/searching completed

```
  # Start of algorithm
       Begin
for (every state s_i: S)       // Generates action for transition\\
 Begin
 Step1: fetch the start state s_i
```

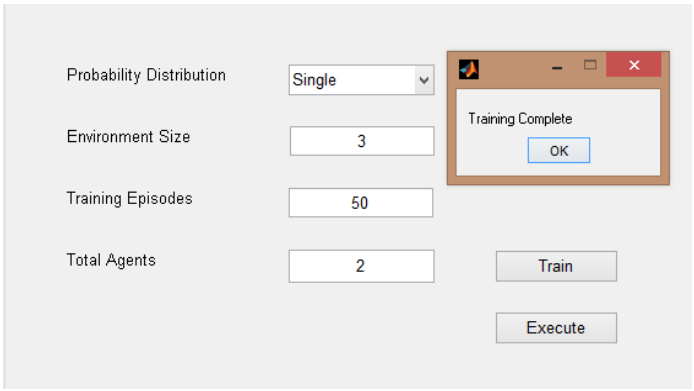
Step2: find possible move or search (L R U D) and execute
Step3: check for goal status
Step4: if (yes: goal=start state)
Update discount rate $d_1=d\gamma^{N_1 - i}$ and i=i+1
Step5: else
{
Goal ≠stat states
Store (s,a) value in look-up-table
}
Step6: Go to step2

End
End. //searching completed\\

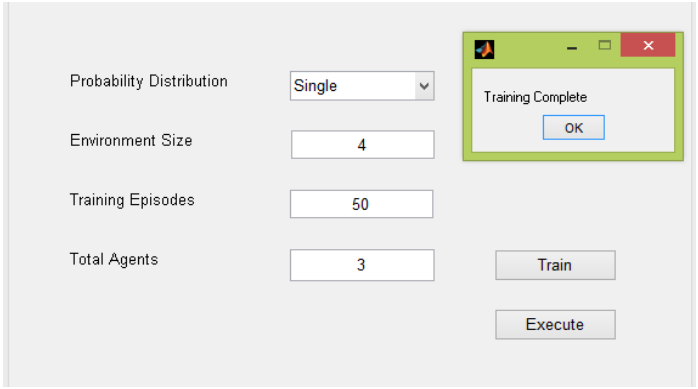
Where discount rate (d_1) is γ their range (0<γ<1), which reduces effect of future expected rewards. N1 is counted step and i is total step in the problem. Action or movement performed by agent is represented in the form of L: Left, R: Right, U: Up, D: Down.

B. Training Model

In this section, training is performed with all the agents in the given environment or grid world problem. The screen shot of the training model Figure 2 is shown as a_1 a_2 a_3 and Figure 3, $b_1b_2b_3$ respectively. In this firstly we select probability density then we have to decide environment size *N*. Now suitable training episode and number of agents are set.



(a_1)



(b_1)

Figure 2. Training Process of Agent for Game Playing

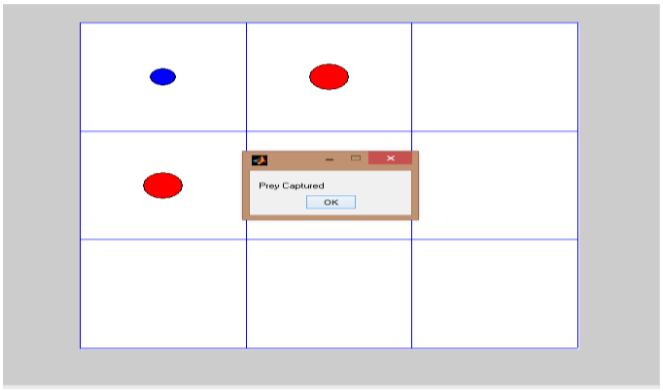
C. Training Parameter

(a_2) For 3×3 States
New States of Agents = 8 6
Pray state=9
Possible pray states new = 0 0 0 0
Capturing time = 25.0493

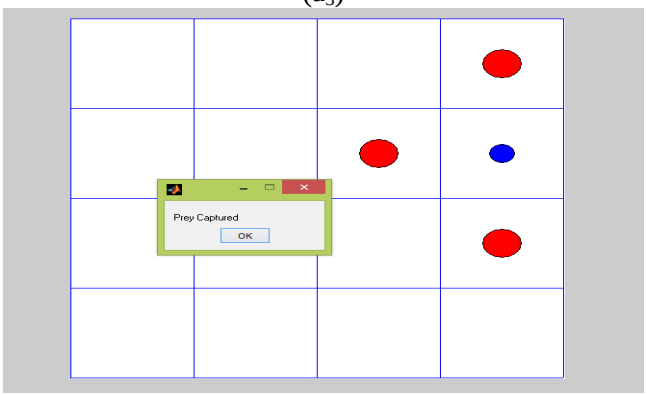
(b_2) For 4×4 States
New States of Agents = 8 3 6
Pray state=4
Possible pray states new = 0 0 0 0
Capturing time = 36.5614

C. Testing Model

After successful training completion in mat lab agent ready to execute for pray capturing as shown in Figure 3 given below.



(a_3)



(b_3)

Figure 3. Trained Agent for Pray Capturing

4. Performance Evaluation

A. Simulation environment

To evaluate our proposed RLSA, with other RL techniques QRL, TDN and DNN-BT in the context of number of episodes and discount rate were generated using [2, 11-12] respectively with shown in Table 2. RLSA help agent to fast training and execution both

shown in Figure 2. During testing phase, found that trained agent has been taken less amount of time for pray capturing shown in Figure 3.

B. Simulation Parameters

Both training and testing agent for game playing was generated on 3x3 and 4x4 grids. For decision making capabilities there discount rate and state action pairs as a look-up-table varies between minimum and maximum values shown in Table 1.

Table 1. Simulation Parameters

S. No.	Parameter	Value
1	Learning rate (α)	0.99
2	Discount rate(γ)	0<γ>1
3	Probability (p)	1
4	Environment size (E)	3 & 4
5	Training episode(E)	50-9000
6	Number of agent	2 & 3
7	Path or states	9 or 16

C. Accuracy Assessments

Performance of RLSA measures are used to assess searching and learning accuracy. Rd is measure of goal capturing efficiency and overall percentage of correctly evaluated training and execution rate.

Efficiency or discount rate $R_d = 1 - \frac{\text{count step (si)} - \text{total step (sn)}}{\text{no of program output}} \text{x100}$

D. Compared Algorithms

We compare the performance of our proposed algorithms RLSA with 3 respective algorithms QRL, TDN [2] and DNN-BT [12]. After completion of total simulation works I found that RLSA is better than others better in the context of goal capturing efficiency or discount rate as well as learning rates also. Here we did the performance comparison between different Algorithms, which have shown in figure 4 and compared results, are tabulated in Table 2.

The worst case complexity of the algorithm is O (n^2), which are in exponential in nature. Redundant paths and goal are recognized and iterative path are removed as can be seen from step2-6 of the algorithm.

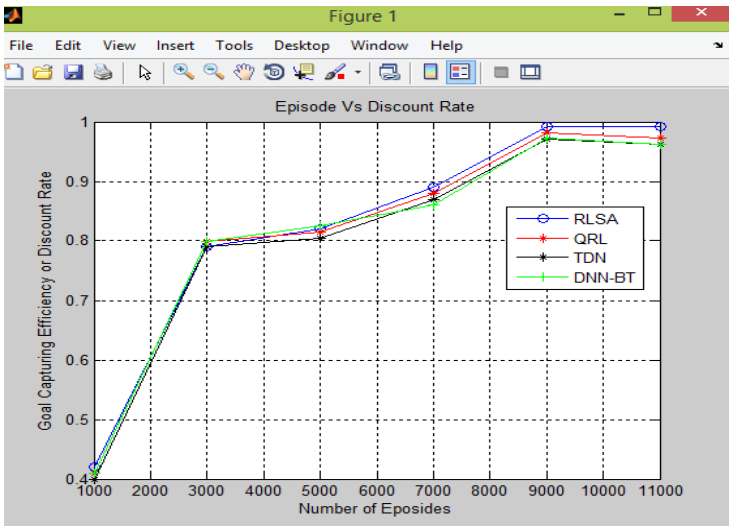


Figure 4. Performance Comparison between Different Algorithms

E. Discussion of Results

The simulation results are shown in figs.2-4.as can be seen from fig.4 reinforcement learning searching algorithm (RLSA) perform better than other three algorithms in terms of capturing efficiency or discount rate and number of given episodes. The reason for better searching of RLSA is to rectify similar paths or data and remove it. During recall session agent takes near to exact solutions from stored look-up-table based on Marko decision process.

5. Conclusion and Future Work

Through this research paper, we inspect that our proposed algorithms as RLSA to boost learning precision and capture time. The main emphasis of the work to modify the reinforcement learning techniques [2, 11] and also confine time varying properties in terms of discount rate. Basic characteristics of RLSA additionally improve rate of learning and avoid overlapping related problem.

In nowadays machine learning is emerging research field for generating and simulating human kind ideas and their application over machine that is known as human computer interaction. In this work, specially focus on game playing searching in the context of discount rate and goal hitting efficiency for pray capturing. This is advanced real application of reinforcement learning in the field of game theory. In future, our research will emphasize in enhancing the ability to generalize environment. Besides this we are also targeting towards the study of complexity issues of the proposed approaches with existing models.

6. Application of Searching Algorithms

There are various applications given below:

(1) Data collection and model learning for flight control
(2) Controller design
(3) Game playing
(4) Resource planning
(6) Dynamic neural network

References

[1] E. Alpayadin, “Introduction to machine learning”, MIT press, textbook, 2005.
[2] Anil kumar yadav and Shailendra kumar shrivastav, “Evaluation of Reinforcement Learning Techniques”, ACM, vol. 132, pp. 88–92. 2010.
[3] Hitoshi Ima and Yaouk Karo, “Swarm Reinforcement Learning Algorithms Based on Sara Method”, IEEE, pp.2045-2049, 2008
[4] R. Caruana and D. Freytag, “Greedy attribute selection”, Proceedings of the Eleventh International Conference on Machine Learning, USA, pp.28-36. 1994.
[5] H. Dejean, “Learning rules and their exceptions”, Journal of Machine Learning Research, 2002.
[6] E. Brill, “Some advances in rule-based part of speech tagging”, Proceedings of the 12th National Conference on Artificial Intelligence, Washington, 1994.
[7] Andrew and Julian, “Cartesian Genetic Programming encoded Artificial Neural Networks, ACM, pp.1005-1012, 2013.
[8] E.F. Tjong Kim Sang, “Memory-based shallow parsing”, Journal of Machine Learning Research, 2002.
[9] T. Zhang, F. Damereau, and D. Johnson, “Text chunking base on a generalization of winnow”, Journal of Machine Learning Research, 2002.
[10] Keita Halmahera, Tadahiro, “Effective integration of imitation learning and reinforcement learning by generating internal reward”, proceedings of Intelligent Systems Design and Applications, IEEE, pp.121-126,2008.
[11] Anil Kumar Yadav, “Ajay Kumar Sachan, Research and Application of Dynamic Neural Network Based on Reinforcement Learning”,springer, AISC vol.132, pp. 931–942. 2012.
[12] Anil Kumar Yadav, Ajay Kumar Sachan, “DNN Tree Search for Bayesian Reinforcement Learning to Machine Intelligence”, IJSCE, vol.4, issue 4, pp. 8–11. 2014.

[13] Rashid Mijumbi, Maxim Claeys,"Neural Network-based Autonomous Allocation of Resources in Virtual Networks", IEEE, pp.1-6, 2014.
[14] XiongYang, Derong Liu, "Neural-network-based online optimal control for uncertain non-linear continuous-time systems with control constraints", IET, pp. 2037-2047, 2013.
[15] Bin Xu, Chenguang Yang, "Reinforcement Learning Output Feedback NN Control Using Deterministic Learning Technique", and IEEE Transaction on neural network and learning system, VOL. 25, NO. 3, pp.635-641, 2014.
[16] Wei Sun, Xuedong, "Reinforcement Learning Method for Continuous State Space Based on Dynamic Neural Network", World Congress on Intelligent Control and Automation, Chongqing, pp.750-754, 2008.
[17] Junfei Qiao, Zhanjun Hou, "Application of Reinforcement Learning Based on Neural Network to Dynamic Obstacle Avoidance", Proceedings of the IEEE International Conference on Information and Automation pp.784-788, 2008.

Authors

Ajay Kumar Sachan, he had completed his PhD (Computer Science & Engineering) from Rajeev Gandhi Technical University, Bhopal in 2007. He has published more than 25 research papers in National & International Journals. He has more than 21 years teaching & research experience. He is working presently as Director in R.I.T.S Bhopal.

Anil Kumar Yadav, His Pursuing PhD (Computer Science & Engineering) from I.F.T.M university, India and had completed his M.Tech (Information Technology) from S.A.T.I., Vidisha, India in 2009 and B.Tech (Computer Science & Engineering) from U.P.T.U. Luck now in 2005.Current research include Artificial intelligence, Machine Learning and Data mining.

Table 2. Trail Efficiency Measures for Various RL Methods

Trail Efficiency/ discount rate (d)	Proposed Algorithm (RLSA) For Episode (E)					Query based RL Algorithm For Episode (E)					TDN Algorithm For Episode (E)					DNN-Bayesian Tree Algorithm(DNN-BT) For Episode (E)				
	1000	3000	5000	7000	9000	1000	3000	5000	7000	9000	1000	30000	5000	7000	9000	1000	3000	5000	7000	9000
d_1	72	98.9	83.6	99.1	85	70	98.8	83	99.8	83	97.9	71	76.6	98.9	99.2	77	70	83.3	99.7	83
d_2	71.50	74	62	82.1	97.1	70.9	70	63	83	98.8	71	70	62	79.9	82.5	70.4	77.6	63.5	84	98.7

d_3	83.60	70.7	96.1	99.6	83.1	71.1	68.9	97.6	97.6	82.9	69.9	72	93.5	76.6	85.5	70	68.5	97.6	97.7	83.5
d_4	85	85.3	98	84.6	99.7	83	74.5	98.7	83.90	99.8	75	82.9	95	82	84.9	81	81.7	98.8	83.9	99.6
d_5	98	98.4	57	99.1	97	99.9	99.8	51.1	99.05	98.9	99.7	98.8	53	95	96.4	97.2	98.7	53.7	98.9	99.7
Average (%)	**81.70**	**85.4**	**79.3**	**92.9**	**92.9**	**78.9**	**82.4**	**78.6**	**92.6**	**92.8**	**82.7**	**78.9**	**76**	**86.4**	**89.7**	**79.1**	**79.3**	**79.3**	**92.8**	**92.8**