Proceedings of the 33rd International Conference on Information Technologies (InfoTech-2019)
19-20 September 2019, Bulgaria

KNOWLEDGE TRANSFER IN REINFORCEMENT LEARNING AGENT

***Digest of paper*[1]**

Vanya Dimitrova Markova and Ventseslav Kirilov Shopov

Institute of Robotics , Bulgarian Academy of Sciences
139 Rouski blvd., 4000 Plovdiv, Bulgaria
markovavanya@yahoo.com
Bulgaria

Abstract: This manuscript is focused on transfer learning methods for reinforcement learning agents. An preview of contemporary papers in area of transfer Leaning and Knowledge transfer. We provided the background and overview of knowledge transfer methods with an emphasis on the topics of reinforcement learning.
Key words: knowledge transfer, transfer learning, reinforcement learning

1. INTRODUCTION

Multy-Agent Systems (MAS) are becoming more widely used in robotics, distributed management, computer games, hierarchical systems and other areas. The problem of presenting and transferring knowledge between agents is becoming a reality, in order to make more efficient use of resources or fast and secure decision making. The Machine Learning framework provides an answer to these issues, but also puts a number of research challenges. Machine Learning [2, 3] is a computer science research area considering methods for identifying and using patterns in a given task based on collected data related to the task and without any explicit programming. Methodologically, machine learning is strictly related to computational statistics with which they share a series of problems and methods [1, 4].

A many of this problems can be formulated and solved as machine learning problems. The sequential decision problems can also be solved using machine learning methods. In this case we want to learn a desirable behaviour over time for an agent acting in a largely unknown environment [5]. Based on the type of this information, machine learning methods can be broadly categorized to the following families of

[1] The full paper is proposed for including in the IEEE Xplore Digital Library

methods [6, 7]: Supervised Learning; Unsupervised Learning and Reinforcement Learning. Reinforcement learning is a machine learning paradigm addressing the problem of how an agent can learn a behaviour through trial-and-error interactions with a dynamic environment [8].

Similarly to reinforcement learning, transfer learning (TL) [1, 9, 10] shows first that humans learn a task better and faster when they have first experienced a similar task [11]. For humans, the cognitive ability of seemingly sharing knowledge across similar tasks is innate and present at all ages. The primary objective of this article is to study and analyse transfer learning approach for reinforcement learning agents.

This paper is organized as follows: in the second part, we describe the theoretical background and our implementation. In the third, we describe our experimenters and present the results. In the last part, we make a conclusion.

2. KNOWLEDGE TRANSFER IN REINFORCEMENT LEARNING

The idea of reusing or transferring knowledge can often be considered as a secondary or additional process. In the context of RL, transfer training [9, 10] refers to the process of knowledge use, which are acquired in one or more pre-learned tasks.

The TL setting describes a specific transfer problem. For example, it could be a scenario in which we attempt to transfer knowledge across two tasks with different state-action spaces or it could be a multi-task scenario in which we attempt to speed up learning for a set of tasks belonging in a specific domain. The transferred knowledge dimension categorizes TL methods by the type of knowledge they are designed to transfer [13, 14].

The knowledge to be transferred between agents and tasks is different according to the different TL methods. They include value functions, whole policies, task models. In this section we present three categories of knowledge transfer that are strongly related to those in [9, 15]. Value functions are transferred in the Value-Addition method presented in and policies in the form of action advices in the Q-Teaching method [16].

Meta-Knowledge transfer includes TL methods that transfer characteristics of a task or characteristics of a successful source task policy. An example of such a method is [11] where the best set of features for approximating a source task is transferred successfully to a target task. In [16] a TL algorithm is proposed which generates and transfers heuristics. It first learns a source task using RL and storing the knowledge obtained in a case base. Then, it does an unsupervised mapping of the source task actions to the target task actions as heuristics and speed up learning in the target task [17].

3. EXPERIMENTS AND RESULTS

We present the transferred knowledge as an internal representation of a learned policy either in the form of value functions or policies. This presentation is policy-specific since the transfer knowledge is meaningful for a very specific way across two agents in similar domains. The objective of learning speed improvement is about reducing the training time needed in order to solve a task and this usually means reduced time complexity.

In our study, the agent does not know the exact values of these two functions. And it approximated them during their training. R may be a combination of medium and agent models, but we will only consider the case where R is only dependent on the environment. There is a nuance here that a part of P could be the same for all tasks, and another part could be specific to each task.

We may ask the following question: does it matter how close the positive targets are and what is the measure of proximity? We define a measure of distance as Manhattan distance to positive goals in two different charges. Therefore, we can define min_dist and max_dist. We define min_dist when the two targets coincide. On the other hand, max-dist is when the two targets are maximally spaced. Thus, we define a metric distance between goals (tasks).

Our basic hypothesis is that when applying TL, the student agent will be trained faster in similar domains. We will measure the effectiveness of training with the difference in the overall prize after the first N episodes.

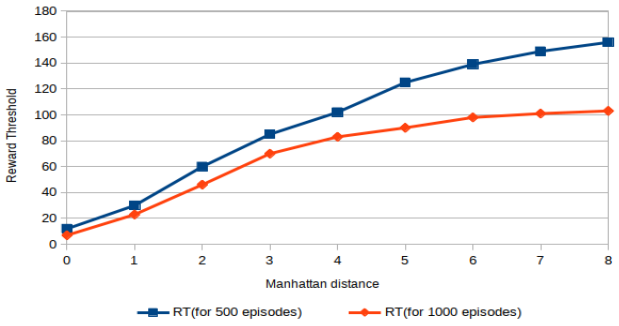

Fig. 1. Transfer Learning trough Q function.

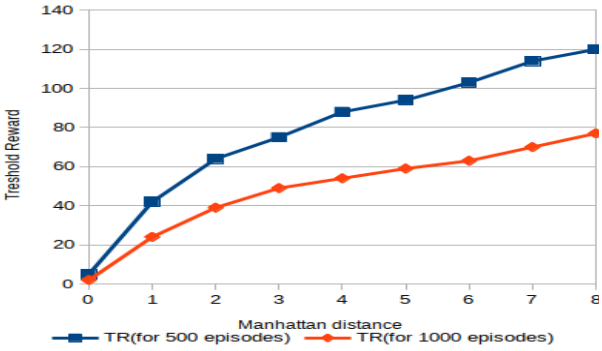

Fig. 2. Transfer Learning trough policy.

We have 2D environment map. On this map are placed targets and traps. The agent must reach the target with a minimum number of steps. Ina addition agent should avoid traps. there are also obstacles placed In the map. If the agent' next state lies in such obstacle then we name this a collision. After collision agent returns to its previous state. Each map of the environment is characterized by the number of traps and the number of obstacles.

First, we create a basic task. Then we generate tasks with the same traps and obstacles but with different targets. At the next stage, we sort the tasks according to Manhattan distance between the target states. After that we create the following pairs: <source_task, target_task_01>. For each pair, we perform the RL and TL process, and as we result we get the Time-to-Threshold criteria. We conduct two experiments: the first we pass the Q function. And in the second experiment, we pass the final policy.

From Figure 1 and Figure 2 one can clearly see that the difference in benefit to the TL approach is significant. We can say that when we have a small difference between the target positions, the use of TL leads to a significant improvement in summary rewards. As the number of episodes increases, this difference decreases. We may conclude that TL is particularly useful at an early stage of training. Which shows that TL can speed up the training of autonomous agents. From both experiments, it can be concluded that the transmission of the Q function and the transmission of the policy lead to a significant acceleration of the training in the earlier stages.

4. CONCLUSION

In classical RL, agents start with a zero or random initialized policy and quality (value) function even when they solve similar tasks. Consequently, knowledge already accumulated does not help to speed up training in subsequent similar tasks.

From both experiments, it can be concluded that the transmission of the Q function and the transmission of the policy lead to a significant acceleration of the training in the earlier stages.

REFERENCES

[1] Bengio, Y., Lodi, A., & Prouvost, A. **(2018).** Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon. *arXiv preprint arXiv:1811.06128*.

[2] Bishop, C. M. **(2006)**. *Pattern recognition and machine learning*. springer.

[3] Mitchell, T. M. **(1997)**. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*.

[4] Fachantidis, A., Partalas, I., Taylor, M. E., & Vlahavas, I. **(2014)**. An Autonomous Transfer Learning Algorithm for TD-Learners. In *Hellenic Conference on Artificial Intelligence* (pp. 57-70). Springer, Cham.

[5] Ng, A. Y., Harada, D., & Russell, S. **(1999)**. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML* (Vol. 99, pp. 278-287).

[6] Fachantidis, A., Partalas, I., Tsoumakas, G., & Vlahavas, I. **(2013)**. Transferring task models in reinforcement learning agents. *Neurocomputing*, *107*, 23-32.

[7] Sutton, R. S. **(1978)**. Single channel theory: A neuronal theory of learning. *Brain Theory Newsletter*, *4*, 72-75.

[8] Sutton, R. S., & Barto, A. G. **(1998)**. Introduction to reinforcement learning. Vol. 135.

[9] Lazaric, A. **(2012)**. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning* (pp. 143-173). Springer, Berlin, Heidelberg.

[10] Stone, P., Kaminka, G. A., Kraus, S., & Rosenschein, J. S. **(2010)**. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

[11] Lazaric, A., Restelli, M., & Bonarini, A. **(2008)**. Transfer of samples in batch reinforcement learning. In Proceedings of the 25th international conference on Machine learning (pp. 544-551). ACM.

[12] Puterman, M. L. **(2014)**. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

[13] Fachantidis, A., Di Nuovo, A., Cangelosi, A., & Vlahavas, I. **(2013)**. Model-based reinforcement learning for humanoids: A study on forming rewards with the iCub platform. In 2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB) (pp. 87-93). IEEE.

[14] Fachantidis, A., Partalas, I., Taylor, M. E., & Vlahavas, I. **(2013)**. Autonomous Selection of Inter-Task Mappings in Transfer Learning. In *2013 AAAI Spring Symposium Series*.

[15] Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., & Fujimura, K. **(2002)**. The intelligent ASIMO: System overview and integration. In *IEEE/RSJ international conference on intelligent robots and systems* (Vol. 3, pp. 2478-2483). IEEE.

[16] Torrey, L., & Taylor, M. **(2013)**. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (pp. 1053-1060).

[17] M. Wiering and M. Van Otterlo, 'Reinforcement learning', *Adapt. Learn. Optim.*, vol. 12, p. 3, 2012.

[17] M. Wiering and M. Van Otterlo, 'Reinforcement learning', *Adapt. Learn. Optim.*, vol. 12, p. 3, 2012. Heidelberg.