Advice Replay Approach for Richer Knowledge Transfer in Teacher Student Framework

Extended Abstract

Vaibhav Gupta*
International Institute of Information Technology
Hyderabad, India

Daksh Anand*
International Institute of Information Technology
Hyderabad, India

Praveen Paruchuri
International Institute of Information Technology
Hyderabad, India

Balaraman Ravindran
Indian Institute of Technology
Madras, India

ABSTRACT

One of the major drawbacks of RL is the low sample efficiency of the learning algorithms. In many cases domain expertise can help to mitigate this effect. Teacher-Student framework is one such paradigm, where a more experienced agent (teacher) upon being queried helps to accelerate the student's learning by providing advice on the action to take in a given state. Real world teachers not only provide the action to take in a given state but also provide a more informative signal using the synthesis of knowledge they may have gained with experience. With this motivation, we propose a richer advising framework where the teacher augments the student's knowledge by also providing the expected long term reward of following that action. The student can then use this value to steadily guide its Q-Network in the correct direction which can lead to a quicker convergence. To help student relive the advices received throughout its learning, we introduce an additional memory called the Advice Replay Memory (ARM). Results show that a student following our approach (a) is able to exploit the environment better, and (b) has a steeper learning curve.

KEYWORDS

Reinforcement Learning; Teacher-Student Framework; Transfer Learning; Deep Learning;

ACM Reference Format:
Vaibhav Gupta, Daksh Anand[1], Praveen Paruchuri, and Balaraman Ravindran. 2019. Advice Replay Approach for Richer Knowledge Transfer in Teacher Student Framework. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Low sample efficiency of RL algorithms becomes critical in many real world domains like [5] [4] [1]. Transfer Learning [9] aims to alleviate this problem by exploiting domain expertise to accelerate the learning speed. Teacher-Student framework [10] is one such paradigm where a more experienced agent (teacher) helps to accelerate the student's learning by providing advice on the action to take in a given state. The student queries the teacher about a particular state and the teacher provides advice about the action to take in that state. An RL agent typically tries sub optimal actions repeatedly before converging to the optimal policy. Such transfer of advice can help mitigate this behavior and potentially lead to a better learning curve.

Although this guided exploration helps the student to explore the promising parts of the state space more effectively, real world teachers typically don't provide only the action to take in a given state. Instead, they provide a more informative signal using the synthesis of the knowledge they may have gained over continuous experience in the environment. Such richer advising can help the student to relate better to the action, enabling it to make better use of this knowledge in the future. Using this insight, we propose to extend the current advising framework to utilize the teacher's knowledge better. Our approach is a first step towards studying such informative advice exchange methods in RL. In particular, the proposal is to make the information provided by the teacher richer by advising the student not just on the best action to take in a state but also about how promising the action is. Given that teacher in our paper is modeled as an RL agent with a Q-function [12], it has a natural way of calculating this measure. We therefore propose to include the Q-value associated with the state-action pair as the additional piece of advice.

In the real world, agents do not have access to each other's learning architectures. The environment models, internal network structures and learning algorithms might vary across the different agents. One agent might be using Sarsa(λ) [8] as the learning algorithm and a much deeper / complex neural network like the one proposed in [7] as the learning architecture whereas the other agent might be using Double Q-learning [11] along with a smaller / simpler network as proposed in [6]. Hence no direct mapping exists between the two agents and using each other's internal parameters becomes infeasible.

For experimentation purposes, we demonstrate the results of our approach on the 5 Atari 2600 games from the Arcade Learning Environment [3]. We used the 2 most popular deep Q-learning models proposed in [6] and [7], as the underlying neural networks for the student and the teacher. We believe that the use case where the teacher is trained on a complex network while the student is trained on a much simpler network is particularly useful. This is because in real world situations the student can have small memory

*Equal contribution.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada. © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

and computation limits (like hand-held devices). Such a student can significantly benefit by querying the teacher who may have access to more memory and computation power (a large server) and hence can train using a more complex architecture. Our results show that the student following our approach

- is able to exploit the environment better
- has a steeper learning curve

2 APPROACH

We propose to extend the current Teacher-Student framework by making the teacher provide more information to the student by advising it not just on the best action to take in a state but also about how promising that action is. Given that we model the teacher as an RL agent with a Q-function, it has a natural way of calculating this measure given by $Q(s, a)$ which is the long term reward associated with an action. An RL agent without a teacher would have to learn this state-action pair value on its own. Once the expected value v_T becomes available through the teacher's advice, the agent can move its network in the direction of these values.

Due to the high level of generalization over the state space in DQN, the input (s,a) cannot be directly mapped to the output v_T in one shot. Doing large updates in direction of v_T might lead to overfitting and unstability. Also, it would be wasteful to use these values only once and then forget about them. Student must also give proper importance to the experiences it gathers by following its own policy. Hence, the student needs to smoothen the training over both kind of experiences: experiences due to its own policy and experiences due to teacher's advice. To balance learning across both these experiences the student along with its normal updates, also needs to steadily move its network in the direction of the advised values throughout its learning. To handle this, we introduce an additional replay memory called the **A**dvice **R**eplay **M**emory, (**ARM**). ARM stores the advice tuples ($state$, $action$, $value$) given by the teacher. The student can therefore repeatedly relive the experiences due to teacher's advice over time. Thus the student's network has to minimize a loss L' due to the tuples in ARM given by -

$$L'(\theta) =_{s,a,v_T} [(v_T - Q(s, a; \theta))^2] \quad (1)$$

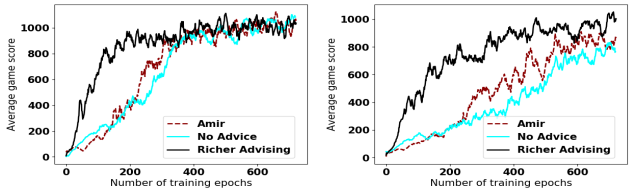
Due to the nature of updates in Q-Learning, the target of DQN $((r+max_a Q(s', a)))$ fluctuates. On the other hand, the loss due to the tuples in ARM has a fixed target value v_T, since these values were given by the teacher who is assumed to be following a fixed policy. The student relives experiences from these two replay memories. To smoothen the training over both of these experiences, at every training step, we randomly sample a mini batch from each of these memories and compute the losses for them.

3 EXPERIMENTAL ANALYSIS

This section showcases the practical performance of our approach. To demonstrate the generality of our framework we conducted 2 sets of experiments. First, we demonstrate flow of knowledge from the teacher to the student, with both having the same network architecture. Then we demonstrate knowledge transfer across heterogeneous network architectures using the models proposed by [7] and [6] as the underlying architectures for the teacher and the student respectively. We compare the performance of our algorithm against current state of the art Teacher-Student advising framework [2]. In their approach, the teacher provides advice only about the optimal action to take in the given state. We ran our experiments on a set of five Atari 2600 domain games from the Arcade Learning Environment [3] namely: Boxing, Space Invaders, Alien, Qbert and Breakout. Due to space constraints, we present the graphs only for Qbert.

Figure 1 shows that due to guided exploration, the student to whom the teacher was providing advice about the optimal action to take, is able to outperform (in terms of learning rate) the agent who was learning without any teacher. We can also observe that the performance of the student is boosted further when the teacher appends the knowledge of the estimated Q-value of the action in the advice.

Performance improvement in the case of heterogeneous network architecture: One thing that stands out in the case of heterogeneous networks is that, advising is leading to a much higher performance improvement as compared to the case when the teacher and the student had homogeneous networks. [7] showed that their model is able to outperform the model proposed by [6] in terms of the maximum achievable scores on convergence. It implies that by learning without teacher, the underlying network architecture of the student will not be able to perform as good as the teacher's network. Combining this with the average game score values from figure 1 (b), we conclude that advising from a comparatively superior teacher therefore becomes a lot more valuable for the student and results in a much higher performance.



(a) Homogeneous networks **(b) Heterogeneous networks**

Figure 1: Qbert training comparison: Agents were trained for 30M frames with each training epoch size of 40k frames.

4 CONCLUSION AND FUTURE WORK

This paper investigates the role of richer knowledge transfer for the case of Teacher-Student framework. We show that if the teacher provides advice to the student about the optimal action to take along with its expected long term reward, the student is able to outperform the state-of-the-art advising framework. In spite of many papers that looked at the teacher student architecture, usage of Qvalue (v_t) as advice to student has not been explored in literature. We not only propose the idea but also present a complete framework to make effective use of the richer advising model. In particular, we introduce a novel architecture named ARM to effectively use the advice provided by the teacher. We also propose a way of giving proper attention to both kinds of experiences the student receives (experiences due to advice from teacher and experiences due to normal learning).

REFERENCES

[1] Pieter Abbeel, Adam Coates, and Andrew Y Ng. 2010. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research* 29, 13 (2010), 1608–1639.

[2] Ofra Amir, Ece Kamar, Andrey Kolobov, and Barbara Grosz. 2016. Interactive teaching strategies for agent training. (2016).

[3] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47 (2013), 253–279.

[4] Maja J Matarić. 1997. Reinforcement learning in the multi-robot domain. In *Robot colonies*. Springer, 73–83.

[5] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng. 2005. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 593–600.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[8] Richard S Sutton, Andrew G Barto, et al. 1998. *Reinforcement learning: An introduction*. MIT press.

[9] Matthew E Taylor and Peter Stone. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10, Jul (2009), 1633–1685.

[10] Lisa Torrey and Matthew Taylor. 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*. International Foundation for Autonomous Agents and Multiagent Systems, 1053–1060.

[11] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-Learning.. In *AAAI*, Vol. 2. Phoenix, AZ, 5.

[12] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8, 3-4 (1992), 279–292.