Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Policy Reuse in Deep Reinforcement Learning*

Ruben Glatt, Anna Helena Reali Costa
Escola Politécnica da Universidade de São Paulo, Brazil
Av. Prof. Luciano Gualberto, n.158, Engenharia Elétrica, São Paulo, CEP: 05508-970.
{ruben.glatt,anna.reali}@usp.br
http://www.cowhi.org, Phone: *+55 11 97019-3738*

Abstract

Driven by recent developments in Artificial Intelligence research, a promising new technology for building intelligent agents has evolved. The approach is termed Deep Reinforcement Learning and combines the classic field of Reinforcement Learning (RL) with the representational power of modern Deep Learning approaches. It is very well suited for single task learning but needs a long time to learn any new task. To speed up this process, we propose to extend the concept to multi-task learning by adapting *Policy Reuse*, a Transfer Learning approach from classic RL, to use with Deep Q-Networks.

Introduction and Context

The area of *Reinforcement Learning (RL)* has been gaining a lot of renewed interest lately. It is concerned with solving sequential decision-making problems that can be formulated as *Markov Decision Processes*, where an agent explores a given environment by performing actions and observes the feedback he receives to deduce a behavior policy to solve a given task (Sutton and Barto 1998).

When dealing with large state spaces it is often beneficial to work with abstractions, which can be realized with *Neural Networks (NN)*. Advances in algorithms for *Deep NNs (DNN)* have brought up a new wave of successful applications of these networks in *RL* and established the field of *Deep Reinforcement Learning (DRL)* (Mnih et al. 2015).

In *DRL*, a trained *DNN* can be seen as a kind of end-to-end RL approach, where the agent learns a state abstraction and a policy approximation within a single network directly from input data. The policy follows the optimal action-value function $Q^*(s, a)$ to select the best action in every state towards solving the task. $Q^*(s, a)$ is approximated by the *DNN*, which is therefore termed *Deep Q-Network (DQN)*.

However, *RL* algorithms in general need a long time to converge to good results and the resulting policy can commonly only be used for a single task.

*We are grateful for the support from CAPES and CNPq (grant 311608/2014-0). The HPC resources for the computation are provided by the Superintendency of Information at the University of São Paulo. We also thank for the support from Google (Research Award) and the Nvidia corporation (GPU donation).

Objective

We have seen impressive results in many areas utilizing *DRL* over the last years but those remarkable successes are mostly located in the domain of single task learning, while knowledge transfer and multi-task learning are underrepresented subjects to date.

Consequently, in our research efforts we are interested in investigating the adaptation of *DRL* to the multi-task case, where *Transfer Learning* approaches can be used to benefit from gathered knowledge from previous experiences (Taylor and Stone 2009). The goal of our research is to overcome the restriction to single task learning for *DQN*s by exploiting past experiences, describing appropriate methods to determine similarities between tasks, and selecting the *right* knowledge to transfer.

In this context, we are initially proposing to adapt effective solutions for knowledge transfer from classic to deep *RL*. Here, we specifically propose to modify *Policy Reuse* for use with *DQN*s (Fernández and Veloso 2006).

Related work

In *Policy Reuse*, a library of policies is built over time. During the training of a new task, policies from the library can be used to speed up training and improve results. The agent determines the suitability of already existing policies at the beginning of every training episode and chooses the currently best policy to accelerate the training process of the new policy. After the training is complete, the agent decides if the new policy should enter the library or not by comparing the result minus a threshold value with the best result from existing policies. Over time the library is expected to present a set of core policies for the domain.

More recently, a general *Single Policy Network*, coined *Actor-Mimic-Network (AMN)*, was introduced to solve a variety of distinct tasks using the guidance of several expert networks (Parisotto, Ba, and Salakhutdinov 2015). In this approach, a bigger network is trained with the help of previously trained expert *DQN*s to initialize the parameters of a new *DQN* for unknown tasks using a network compression technique. The training of the multi-task policy in the *AMN* not only takes the expert policy into account but also integrates the feature representation of each expert. Although the results show that this approach improves and speeds up

the learning of multiple tasks especially at the beginning of training, there are two major downsides. First, once trained, the *AMN* stays fixed and does not evolve with a growing number of tasks. Second, after the initialization it has no more influence on the training process.

Another interesting approach is *A Deep Architecture for Adaptive Policy Transfer (ADAAPT)*, which blends the policies of a set of previously trained experts during training (Rajendran et al. 2015). The key component of this proposal is an additional network which learns weights for each source policy, termed attention network. While the agent is learning the new policy, current action values for each state are calculated by multiplying the weights of the attention network with the results from the expert policies and the current policy. This approach is especially well suited for transfer from multiple sources if the tasks share the same state and action space. It is also robust to negative transfer because bad policies are only considered with a low weight, while good policies have higher weights. The downside of this approach is that the network might miss out on a better actuation with higher rewards because of the blended policy.

Proposal

Adapting the original *DQN* algorithm to work in a multi-task scenario using *Policy Reuse* enforces some changes to both approaches (see simplified sketch in Figure 1).

In *Policy Reuse*, the policy used during training is selected for every training cycle, or episode, from the current policy or the policies in the library. Since the episodes in our experiment domain are initially rather short, we will investigate the case where the reused policy is selected only after a greater number of episodes.

The major changes have to be in the *DQN* algorithm itself to face the problems arising from comparing different policies from different tasks with different valid action sets. First, it appears to be favorable to adapt the structure of the *DQN* to provide a fixed length output layer to represent all possible actions for all tasks. Then, the network is extended with a softmax layer on top to produce a probability for each action instead of an expected reward value to deal with the different reward structures of the individual tasks. During training only task relevant actions are considered for the softmax, while the other actions are masked out. This also has implications on the loss function, which has to consider the cross-entropy between the target and current network.

The alternative policy to reuse during training is selected from the policy library and stays fixed for a number of episodes. The selection of the action to perform follows the *Policy Reuse* exploration strategy, where we choose an action following the policy from the library with probability ψ or apply the usual exploration/exploitation strategy otherwise as indicated in Figure 1.

The greatest challenge is to determine the suitability of a policy for a given task by providing a similarity metric. We intend to investigate at least two possible approaches: (1) Run a short number of determination episodes to see, which existing policy achieves the highest score on average and select it for the following episodes. (2) Introduce a second network, which is trained in parallel and gives us an immediate indication of the suitability of a certain policy, similar as the attention network of the *ADAAPT* approach.

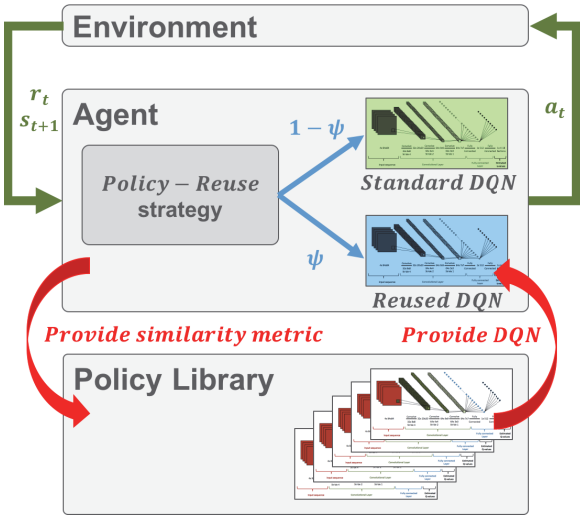

Figure 1: Simplified sketch of integrating *Policy Reuse* into the *DQN* algorithm.

We intend to show benefits over the single task approach and compare our results with *AMN* and *ADAAPT*.

Conclusion

We think that *TL* offers great potential to accelerate *DQN* single task learning, but there are still many aspects to be understood before we can formulate a comprehensive framework for knowledge transfer across *DQN*s.

In the long-term we also intend to investigate the use of skills to allow for a more targeted transfer with common subtasks, and to explore what role advice from a general expert can play for knowledge transfer.

References

Fernández, F., and Veloso, M. 2006. Probabilistic policy reuse in a reinforcement learning agent. In *AAMAS*, 720–727.

Mnih, V.; Silver, D.; Rusu, A. A.; Riedmiller, M.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Parisotto, E.; Ba, L. J.; and Salakhutdinov, R. 2015. Actor-mimic: Deep multitask and transfer reinforcement learning. *CoRR* abs/1511.06342.

Rajendran, J.; Prasanna, P.; Ravindran, B.; and Khapra, M. M. 2015. Adaapt: A deep architecture for adaptive policy transfer from multiple sources. *arXiv preprint arXiv:1510.02879*.

Sutton, R. S., and Barto, A. G. 1998. *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press.

Taylor, M. E., and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research (JMLR)* 10:1633–1685.