Inter-Task Similarity for Lifelong Reinforcement Learning in Heterogeneous Tasks

Sergio A. Serrano
Instituto Nacional de Astrofísica, Óptica y Electronica
sserrano@inaoep.mx

Abstract

Reinforcement learning (RL) is a learning paradigm in which an agent interacts with the environment it inhabits to learn in a trial-and-error way. By letting the agent acquire knowledge from its own experience, RL has been successfully applied to complex domains such as robotics. However, for non-trivial problems, training an RL agent can take very long periods of time. Lifelong machine learning (LML) is a learning setting in which the agent learns to solve tasks sequentially, by leveraging knowledge accumulated from previously solved tasks to learn better/faster in a new one. Most LML works heavily rely on the assumption that tasks are similar to each other. However, this may not be true for some domains with a high degree of task-diversity that could benefit from adopting a lifelong learning approach, *e.g.*, service robotics. Therefore, in this research we will address the problem of learning to solve a sequence of RL heterogeneous tasks (*i.e.*, tasks that differ in their state-action space).

1 Research Problem

Let $T = (T_1, ..., T_i, ..., T_N)$ be a finite sequence of RL tasks (whose length is unknown by the system) that differ in their state-action space, D_L a lifelong reinforcement learning (LRL) agent (see Fig. 1), D_S a standard RL agent, $C(d, T_i)$ the amount of data required by agent d to learn to solve task T_i, $K(d, \{..., T_i, ...\})$ the memory space required to store the knowledge of agent d after it learned to solve the set of tasks $\{..., T_i, ...\}$, and $P_i^L(T_j)$ the performance of agent D_L in task T_j after it has learned to solve task T_i, where $T_i, T_j \in T$ and $j \leq i$.

Thus, this research is concerned with the analysis and design of an LRL agent D_L that satisfies the following conditions, $\forall i \in [1, N]$:

1. $C(D_L, T_i) \leq C(D_S, T_i)$,
2. $K(D_L, \{T_1, ..., T_i\}) \leq \sum_{k=1}^{i} K(D_S, \{T_k\})$, and
3. $\forall j \in [1, i-1]$ then $P_{i-1}^L(T_j) \leq P_i^L(T_j)$

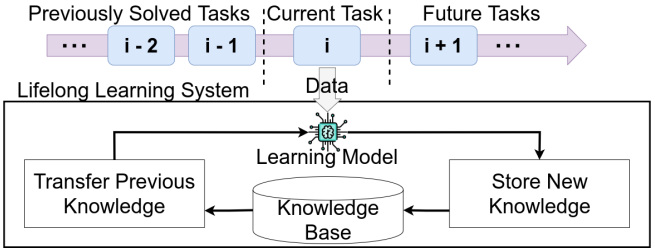

Figure 1: A lifelong learning system learns to solve tasks sequentially by transferring knowledge from previous tasks and storing new knowledge to reuse it in future tasks.

In other words, the LRL agent 1) should learn tasks at least as fast as the RL agent, 2) it should not occupy more memory space than the one required by multiple RL agents, and 3) it should not forget how to solve previously learned tasks.

2 Related Work

In [Kirkpatrick *et al.*, 2017] a method for the consolidation of knowledge in neural networks (NN) is introduced. A single NN is trained on a sequence of tasks. Based on the Fisher information matrix, the learning process is slowed down in those weights that are more *important* for previous tasks. Alternatively, [Rusu *et al.*, 2016] propose to instantiate a new NN for each task the system encounters. Knowledge is transferred via lateral connections, from the hidden units of older NNs to the NN that is currently being trained.

[Mendez and Eaton, 2020] propose to represent a set of task-specific policies, as a factorization $L \cdot s_i$ with a matrix of latent components L and a set of task-specific vectors s_i. Common knowledge is transferred across tasks through L, while s_i encodes the particularities of the i-th task. On the other hand, [Lecarpentier *et al.*, 2020] propose a pseudo-metric based on the Lipschitz continuity between the optimal value function of different tasks. The pseudo-metric determines which of the previous value functions is the closest to the optimal value function of the current task, that is, the best candidate for transferring knowledge to the current task.

In terms of scalability, with respect to the amount of tasks, [Rusu *et al.*, 2016] offer a quadratic growth, [Mendez and Eaton, 2020; Lecarpentier *et al.*, 2020] offer a linear growth, while in [Kirkpatrick *et al.*, 2017] the model remains

with a constant size throughout the sequence of tasks. Regarding the consolidation of knowledge, [Rusu *et al.*, 2016; Lecarpentier *et al.*, 2020] guarantee to remember older tasks, since task-specific knowledge is stored separately. Whereas in [Rusu *et al.*, 2016; Lecarpentier *et al.*, 2020], the capability to retain knowledge depends on how similar tasks are. That is, since task-specific knowledge is partially or completely stored in a set of shared parameters, if tasks are significantly dissimilar then conflicts between their parameters may arise, as a consequence of striving to satisfy different goals.

3 Contributions and Expectations

With the presented research we expect to make the following contributions:

1. An inter-task similarity measure for tasks that differ in their state-action space.
2. A transfer learning algorithm for tasks that differ in their state-action space.
3. A knowledge consolidating algorithm for tasks that differ in their state-action space.
4. A lifelong reinforcement learning algorithm for tasks that differ in their state-action space.

We believe that by designing an LRL system based on the concept of inter-task similarity, our approach will provide a robust learning agent that is capable of exploiting common knowledge between tasks to learn faster, as well as avoiding harming its performance as a consequence of the dissimilarities heterogeneous tasks are likely to present. Hence, our LRL agent will remain *skeptical* about the similarity tasks hold, only until they prove to be similar enough for knowledge consolidation and transfer purposes.

4 Inter-Task Similarity Measure

We propose an inter-task similarity measure as the basis for an LRL agent in heterogeneous tasks. By separating the processes of consolidation and transfer of knowledge, we adopt what [Silver *et al.*, 2013] call the *system* approach. We propose to use the inter-task similarity measure as a heuristic to select from which tasks knowledge should be transferred (similar to [Lecarpentier *et al.*, 2020]), as well to decide how knowledge should be organized and stored. That is, based on the similarity the latest learned task has with previous tasks, the system will decide if they will share parameters or if the new knowledge requires its own set of parameters, in order to avoid harming older pieces of knowledge.

Thus, besides aiming to reduce the data required to learn in each task by transferring knowledge, with the inter-task similarity measure our system will strive to keep the size of the model as small as possible without forgetting how to solve previous tasks. Contrary to the revised literature, our system will instantiate new parameters in an informed manner, *i.e.*, only for significantly dissimilar tasks, while similar tasks can safely share parameters.

Our inter-task similarity measure compares Q-tables to assess the similarity between tasks. We evaluated the similarity measure in three discrete RL tasks: Taxi domain (TX), Frozen Lake (FL) and Frozen Lake 8×8 (F8)[1]. Table 1 shows how the similarity measure correctly assigns a greater score to each task with itself, as well as an improvement in performance after transferring Q-values (with a procedure based on our similarity measure) between FL and F8. This is reasonable, considering that F8 is an extension of FL, where they share the action space but F8 has a larger state space.

	Similarity score			Transer ratio		
	TX	FL	F8	TX	FL	F8
TX	**0.519**	0.481	0.492	-	0.358	0.059
FL	-	**0.496**	0.463	0.994	-	**1.276**
F8	-	-	**0.512**	0.992	**1.167**	-

Table 1: Inter-task similarity scores (higher is more similar) and transfer ratios obtained from transferring knowledge from a source task (rows) to a target task (columns). Transfer ratios above 1 represent an improvement of performance in comparison to learning from scratch.

5 Directions for Remaining Work and Evaluation of Success

Currently, we are focused on the development of the similarity measure for heterogeneous tasks. Specifically, we are working to extend it for tasks with continuous state-action spaces. Based on the findings we obtain, we will incorporate them into the design of the transfer, consolidation and LRL algorithms. Our approach is to analyze what features are the most relevant for transferring and consolidating knowledge, despite a mismatch in the state-action spaces. To determine the success of our LRL system, it must suffice the three conditions presented in Section 1. Additionally, we will evaluate experimentally the LRL agent in a wide variety of control heterogeneous tasks, and compare it to other LRL works.

References

[Kirkpatrick *et al.*, 2017] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[Lecarpentier *et al.*, 2020] E. Lecarpentier, D. Abel, K. Asadi, Y. Jinnai, E. Rachelson, and M. L. Littman. Lipschitz lifelong reinforcement learning. *arXiv preprint arXiv:2001.05411*, 2020.

[Mendez and Eaton, 2020] J. A. Mendez and E. Eaton. Lifelong learning of factored policies via policy gradients. In *4th Lifelong Learning Workshop at International Conference on Machine Learning (ICML)*, 2020.

[Rusu *et al.*, 2016] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[Silver *et al.*, 2013] D. L. Silver, Q. Yang, and L. Li. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*. Citeseer, 2013.

[1] See https://gym.openai.com/envs/#toy_text for more details.