Ising Networks for Deep Hierarchical Reinforcement Learning

1 Introduction

2 Related Work

2.1 Hierarchical Reinforcement Learning

2.1.1 Temporal Abstraction

Various methods have devised hierarchical techniques to abstract the temporal correlation in MDPs [14, 15, 24, 27]. The Hierarchical Deep Q-Network (hDQN) [14] proposes an integration of temporal abstraction and intrinsic motivation in order to solve long-horizon problems. hDQN demonstrates improved long-horizon optimality at the cost of manual feature engineering of the states and reward function. This is addressed by making use of deep successor networks [15] which extract relevant goal embeddings for the agent and direct its behavior towards the goal. A more sophisticated temporal scheme consists of learning abstraction of hierarchies based on expert demonstrations and regularized latent space constraints [29]. However, sub-policies rely on memory-based controllers for learning fine-grained abstraction and often cripple the lower levels of the hierarchy once the task has been accomplished. This leads to limited transfer of skills between policies and hence, results in sub-optimal convergence [6]. Transfer of skills between different policies operating in sub-MDPs is extensively studied using exploration [13]. [20] presents MAVEN, a hierarchical architecture for variational exploration in the case of multi-agent settings. Policies carry out exploration in the latent space using variational inference. MAVEN demonstrates improved coordination between agents in the long-horizon. However, policy behavior distributed across agents which forces sub-policies to rely on other agent policies for optimal convergence. [4] proposes the trajectory autoencoder which enables the agent to exhibit self-consistent behavior and tackle sparse rewards in the long-horizon. Long-horizon suitability can further be improved by jointly training all policies and utilize the skills learned individually [16]. Such kind of sub-policy adaptation stabilizes the behavior of agent and provides efficient training of hierarchies at all levels.

Abstraction may be carried out among states [2] or using a different medium such as language [12] which allows the agent to solve temporally-extended tasks in a diverse manner. [5] presents the MAXQ algorithm which proposes an efficient method for training state-based abstractions in the form of hierarchies. Q-values corresponding to each lower-level task are aggregated at the higher-level policy nodes to yield a modular framework for abstraction. While the MAXQ demonstrates suitability for low-dimensional tasks, its performance cannot be assessed in high-dimensional state and action spaces due to the larger number of abstractions produced by the framework. [1] tackles the problem of high-dimensional and long-horizon learning by presenting a theoretical perspective of abstracting states using transitive behavior which helps in optimal computations. Such techniques have motivated active research in the form of modified abstraction algorithms for performance improvement and schemes leading to richer abstract representations [18].

2.1.2 Option-based Hierarchies

Hierarchies can be constructed on the basis of skills or possible options of policies which the agent may choose from during its interactions in the environment [27, 3]. [3] presents the Options-Critic Architecture which is a collection of various policy options. Agent can select from the collection of sub-policies at various timesteps to yield a policy which is robust and optimal in the long-horizon. The provision of options enables the agent to acquire a wide variety of skills during exploration and obtain a policy which makes best use of these skills. Furthermore, the architecture incorporates learning of termination conditions of a particular option which is essential for switching option policies [10]. The architecture is extended to the Double Actor-Critic architecture which comprises of two parallel MDPs [31]. A dual architecture enables the learning of intra-option policies with termination conditions of each option. While, option-based learning provides diversity in skills and control of temporal extension to the agent, frequent fluctuations between options results in instability in updates and hence, sample-inefficient learning [17]. [17] addresses the instability in training by combining the options framework with off-policy maximum entropy reinforcement learning [9, 8]. Maximization of entropy allows the agent to effectively explore all options and consistently select the optimal sub-policies based on selected options. However, usage of different skills and compactness remains an open problem in the case of real-world tasks consisting of large action spaces. [22] addresses the problem of compact options by improvising communication between policies using binary vectors. Modular policies using binary vectors enable richer communication and mixing of skills. Techniques related to the relative optimization of policy can be extended using advantage-weighted importance sampling [21] in options based on option-value functions.

2.1.3 Hierarchical Control

Hierarchical Reinforcement Learning borrows from multiple approaches in control [23] and adaptive policy optimization [16]. Multiplicative Compositional Policies (MCPs) [23] present the composable nature of hierarchies which arises from coordination of multiple control-based skills. MCPs improve the hierarchical coordination between sub-policies which results in adaptive motor skills in the case of robotic control such as locomotion and manipulation. While MPC demonstrates the practical usage of hierarchical reinforcement learning motivated by adaptive control, [11] extends the hierarchical framework to Quadruped Locomotion. The lower-level policy uses latent commands for robot actuator control and the higher-level policy operates at different time intervals. Abstraction of control between lower and higher-level policies allows the agent to acquire a diverse set of task-related skills. Moreover, policies can be efficiently adapted on a different task from the same domain. Additionally, diversity can be achieved in a scalable manner by training policies levelwise [25]. However, levelwise training of sub-policies becomes intractable in cases of a large number of hierarchies [7]. [7] tackles the problem of a large number of hierarchies and levelwise training by making use of latent space policies. Each latent space policy is trained using the maximum entropy learning framework in order to directly solve the task. Hierarchies trained in latent space demonstrate suitable initializations on complex control tasks which enable the agent to maintain consistent behavior. Consistency can also be achieved using goal embeddings which motivate goal-directed behavior in the case of navigation tasks [26]. [28] additionally improves consistency in hierarchical agents by making use of information theoretic regularizations which pose a reward penalty on the policy distribution. Hierarchical control in reinforcement learning has been successful in tackling knowledge transfer as well as data-efficiency [30]. Various levels of the hierarchy make use of inductive-biases for sharing off-policy data. Sharing knowledge across policies in a compositional manner results in positive transfer of policies. Such techniques are additionally used to solve numerically challenging control problems consisting of unstructured environments [19].

2.2 Ising Models

3 The Ising Model

4 Intuition for Spin Values

5 Implementation Details

References

[1] D. Abel, D. Arumugam, L. Lehnert, and M. Littman. State abstractions for lifelong reinforcement learning. In *International Conference on Machine*

Learning, 2018.

[2] D. Andre and S. J. Russell. State abstraction for programmable reinforcement learning agents. In *AAAI/IAAI*, 2002.

[3] P.-L. Bacon, J. Harb, and D. Precup. The option-critic architecture, 2016.

[4] J. D. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach, P. Abbeel, and S. Levine. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *arXiv preprint arXiv:1806.02813*, 2018.

[5] T. G. Dietterich. State abstraction in maxq hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, 2000.

[6] C. Florensa, Y. Duan, and P. Abbeel. Stochastic neural networks for hierarchical reinforcement learning, 2017.

[7] T. Haarnoja, K. Hartikainen, P. Abbeel, and S. Levine. Latent space policies for hierarchical reinforcement learning, 2018.

[8] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies, 2017.

[9] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[10] A. Harutyunyan, W. Dabney, D. Borsa, N. Heess, R. Munos, and D. Precup. The termination critic, 2019.

[11] D. Jain, A. Iscen, and K. Caluwaerts. Hierarchical reinforcement learning for quadruped locomotion. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[12] Y. Jiang, S. S. Gu, K. P. Murphy, and C. Finn. Language as an abstraction for hierarchical deep reinforcement learning. In *Advances in Neural Information Processing Systems 32*. 2019.

[13] N. K. Jong, T. Hester, and P. Stone. The utility of temporal abstraction in reinforcement learning. In *AAMAS (1)*, 2008.

[14] T. D. Kulkarni, K. R. Narasimhan, A. Saeedi, and J. B. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation, 2016.

[15] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman. Deep successor reinforcement learning, 2016.

[16] A. Li, C. Florensa, I. Clavera, and P. Abbeel. Sub-policy adaptation for hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2020.

[17] C. Li, X. Ma, C. Zhang, J. Yang, L. Xia, and Q. Zhao. Soac: The soft option actor-critic architecture, 2020.

[18] S. Lin and R. Wright. Evolutionary tile coding: An automated state abstraction algorithm for reinforcement learning. In *Proceedings of the 8th AAAI Conference on Abstraction, Reformulation, and Approximation*, 2010.

[19] Q. Ma, S. Ge, D. He, D. Thaker, and I. Drori. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning, 2019.

[20] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems 32*. 2019.

[21] T. Osa, V. Tangkaratt, and M. Sugiyama. Hierarchical reinforcement learning via advantage-weighted information maximization, 2019.

[22] A. Pashevich, D. Hafner, J. Davidson, R. Sukthankar, and C. Schmid. Modulated policy hierarchies, 2018.

[23] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies, 2019.

[24] D. Precup. Temporal abstraction in reinforcement learning, 2000.

[25] Y. Song, J. Wang, T. Lukasiewicz, Z. Xu, and M. Xu. Diversity-driven extensible hierarchical reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 2019.

[26] S. Sukhbaatar, E. Denton, A. Szlam, and R. Fergus. Learning goal embeddings via self-play for hierarchical reinforcement learning, 2018.

[27] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. 2018.

[28] D. Tirumala, H. Noh, A. Galashov, L. Hasenclever, A. Ahuja, G. Wayne, R. Pascanu, Y. W. Teh, and N. Heess. Exploiting hierarchy for learning and transfer in kl-regularized rl, 2019.

[29] W. Wang, Y. Hu, and S. Scherer. Learning temporal abstraction with information-theoretic constraints for hierarchical reinforcement learning, 2020.

[30] M. Wulfmeier, A. Abdolmaleki, R. Hafner, J. Tobias Springenberg, M. Neunert, N. Siegel, T. Hertweck, T. Lampe, N. Heess, and M. Riedmiller. Compositional transfer in hierarchical reinforcement learning. *Robotics: Science and Systems XVI*, 2020.

[31] S. Zhang and S. Whiteson. Dac: The double actor-critic architecture for learning options, 2019.