Sample-Efficiency in Complex Environments using Reinforcement Learning

Nimish S 312217104098
***** 312217104***
***** 312217104***

BE CSE, Semester 7

*****

Supervisor

Project Review: 0 (24 October 2020)

Department of Computer Science and Engineering
SSN College of Engineering

---

1 Abstract

Current Reinforcement Learning algorithms excel at performing in non-hierarchial environments. However, when it comes to complex, hierarchial & sparse environments they are impractical to use as sample-inefficiency becomes more prominent due to the curse of dimensionality. This makes reinforcement learning impractical to use in many real-world situations as many of these environments are complex, hierarchial & sparse in nature. Superior exploration techniques like Curiosity-driven Exploration have proven to improve sample efficiency in simple environments. We propose to use such techniques in combination with RL algorithms like DDPG and DQN to succesfully navigate a complex environment.

2 Introduction

Reinforcement learning algorithms aim at learning policies for achieving target tasks by maximizing rewards provided by the environment. Reinforcement learning combined with neural networks has recently led to a wide range of successes in learning policies for sequential decision-making problems. From a practical standpoint, RL is currently used in simple environments where PID *(Proportional Integral Derivative)* controllers cannot be easily applied (e.g. simple robotic control, HVACs & autonomous vehicles). Currently, it is crucial for the environments to be designed in such a way as to encourage learning.

However, in most real-world applications, there are three important environmental characteristics most reinforcement learning algorithms struggle with. The rewards supplied to the agent are extremely sparse or sometimes missing altogether. The actions to be taken are hierarchial in nature (i.e. each goal state does not result in termination, rather opens the possibility to new goal states). The state space is extremely high dimensional leading to a lot of time spent on extracting useful features from higher-level representations. A combination of all these problems results in extremely sample-inefficient performance of existing algorithms.

In 2018, the team at OpenAI tried to attain human-level control in an extremely complex environment called Dota2. It is a strategic real-time video game with 170,000 possible actions in each time-step that took humans approximately 600 hours to learn. The rewards too were very sparse with the agent attaining significant reward once every 5-10 minutes. A *Proximal Policy Optimization* agent was trained over a period of 10 months. It cost 7.5 million dollars to train and was trained using 128,000 CPU cores and 256 GPUs. It read in 1,048,576 observations every second which is an unimaginably large number of samples. This entire setup is simply impractical for use on a large scale.

Sample-inefficiency is just the result of the explore-exploit dilemma in practice. In order to attain the maximum possible reward, sufficient exploration must be done to identify the best possible sequence of decisions in a given situation and then the agent must exploit its knowledge of the environment. There is a fine line between exploration and exploitation as more exploration would take a lot of time to experience states further down the hierarchy and pure exploitation will result in the agent getting stuck in a local maxima.

Currently for complex enironments, algorithms explore using a naive method based on adding random noise to the action probabilities of the agent. We propose to use intelligent exploration techniques introduced in much simpler environments in combination with powerful RL algorithms to significantly improve sample efficiency in complex environments. In order to test this theory, we plan to use a complex environment called MineRL which is based on a 3D sandbox survival video game called Minecraft. The game focuses on the player collecting resources and crafting materials in order to survive. We plan to test this theory first using the SuperMarioBros environment which is a 2D environment but presents its own set of complications due to its large non-repetitive state space and complex action space.

3 Literature survey

Minh, et al. [1] introduced the off-policy DQN *(Deep Q-Network)* agent which used neural networks to aproximate the values of continuous state spaces. Till then, the applicability of RL has been limited to domains with fully observed, low-dimensional state spaces with discrete actions. The DQN agent could tackle significantly more complex environments than possible before this point. The agent was tested on the Atari 2600 platform which consisted of 49 games. The observation was given in the form of pixel representations of the game at every time step. The performance of the agent was scaled between 100% (professional human player) and 0% (random agent). The experiments showed that with a single set of hyperparameters, the agent was able to achieve superhuman ($>$ 200%) performance in 14 games, professional level ($>$ 100%) performance in 9 games and above average ($>$ 75%) performance in 6 games. This proved the robustness of the algorithm to different environments and also proved that the agents could learn meaningful state representations from high-level features like pixels. DQfD *(Deep Q-learning from Demonstrations)* [2], leverages small sets of demonstration data to massively accelerate the learning process. It is able to automatically assess the necessary ratio of demonstration data while learning thanks to the Prioritized Experienced Replay [3] mechanism.

Lillicrap, et al. [4] presented a model-free, off-policy actor-critic algorithm called DDPG *(Deep Deterministic Policy Gradient)* that can operate over continuous action spaces unlike DQN. DDPG can learn competitive policies using low-dimensional observations (e.g. cartesian coordinates or joint angles) and even directly from pixels using the same hyperparameters and network structure. Robotic control environments, where the agent could control all the joints of a robotic arm in 7 degrees of freedom and gait-simulation environments, where the agent could control the joints of a bipedal and a quadripedal entitiy were used to evaluate the performance of the algorithm. In both cases, the algorithm was able to converge close to an optimal policy within 2.5 million steps. Since DDPG can only work in continuous action spaces and DQN can only work with discrete actions, there is no baseline for comparison between the two. However, it has been generally proven via experiments that in most cases DDPG could converge to a better policy whereas DQN was more sample efficient. DDPG tended to overestimate values of certain states leading it to converge to suboptimal policies. Fujimoto, et al. [5] introduced a modified version of the DDPG algorithm called the TD3 *(Twin Delayed Deep Deterministic Policy Gradient)* which addressed the overestimation bias of the DDPG, allowing it to converge faster to better policies.

Pathak et al. [6] introduced the ICM *(Intrinsic Curiosity Module)* architecture as an alternative to random exploration techniques in environments with sparse or no extrinsic rewards. RL algorithms were predominantly trained in environments that provided a continuous supply of extrinsic rewards. ICM architecture enabled training in environments that resembled real-world scenarios due to their lack of extrinsic rewards. Curiosity also aided agents in exploring environments for new knowledge and learning skills that might help in future tasks. ICM architecture was tested in two environments for 3 different settings - sparse extrinsic rewards, no extrinsic rewards and novel scenarios. The sparse extrinsic reward experiment was performed on VizDoom by varying the degree of reward sparsity. The curious agent learnt much faster in the 'dense' and sparse reward case. The baseline agent failed to solve the task in the sparse reward case while both failed to solve the 'very sparse' reward case. The no reward setting saw agents successfully explore and navigate environments with no extrinsic rewards in Super Mario Bros and VizDoom environments. The agent also discovered winning behaviours. The novel-scenario setting evaluated generalization of agent behaviours in two ways: 'as is' (no further learning) and fine-tuning the policies. Good generalization was observed in 'as-is' for scenarios that weren't significantly different visually. Fine-tuning with curiosity helped tackle the mentioned drawbacks.

Andrychowicz et al. [7] introduced a novel technique called HER *(Hindsight Experience Replay)* that enabled sample-efficient learning by combining it with off-policy RL algorithms to problems with sparse and binary rewards. A standard RL algorithm would learn little to nothing from a sequence of actions that lead to an unsuccessful state while HER encourages a replay of the same actions for a different goal giving rise to faster learning. Transition tuples encountered during training are stored in a replay buffer for harvesting information using different goals. HER was tested in combination with off-policy algorithms like DDPG for 3 different robotic control tasks - pushing, sliding and pick-and-place. For HER, each transition was stored in the replay buffer twice: for the goal used to generate the episode and the goal corresponding to the final state. DDPG without HER was unable to solve any of the tasks while DDPG with HER solved them all, almost perfectly. DDPG with HER improved performance even when the goal state was identical in all episodes. The agent learned faster when training episodes had multiple goals. When reward functions were shaped (and not the previous binary), both DDPG and the HER supported agents were unable to solve any of the tasks. An analysis of alternative replay goals tested future, episode and random goal strategies. It showed that the most valuable goals for replay were ones that were going to be achieved in the near future. HER was also tested on a physical fetch robot without fine-tuning and performed well on the pick-and-place task.

4 Environments

References

[1] Mnih, V., Kavukcuoglu, K., Silver, D. et al. *Human-level control through deep reinforcement learning.* Nature 518, 529–533 (2015).

[2] Hester, T., Vecerik, M., Pietquin, 0., et al. *Deep Q-learning from Demonstrations.* ArXiv (2017).

[3] Schaul, T., Quan, J., Antonoglou, I., and Silver, D. *Prioritized Experience Replay.* ArXiv (2015).

[4] Lillicrap, T., Hunt, J., Pritzel, A. et al. *Continuous control with deep reinforcement learning.* ArXiv (2015).

[5] Fujimoto, S., Hoof, H., Meger, D. *Addressing Function Approximation Error in Actor-Critic Methods.* ArXiv (2018).

[6] Pathak, D., Agrawal, P., Efros, A., Darrell, T. *Curiosity-Driven Exploration by Self-Supervised Prediction.* IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017).

[7] Andrychowicz, M., Wolski, F., Ray, A., *Hindsight Experience Replay.* Advances in Neural Information Processing Systems 30 (NIPS) (2017).