Reinforcement Learning Based Generative Art Agent

Literature Review

Nathan Grabaskas
Zhizhen Wang

1. General Problem

Drawing and creative art are a critical part of human civilisation and culture. To learn how to draw would take years of learning and practising for humans. Hence we want to explore the idea of training an artificial intelligence agent that can paint strokes on a canvas in sequence to generate a new painting.

The experiment contains two parts. First part is training the agent. We will start by defining a 2D environment (canvas) for agents to simulate the painting, and each training canvas will be initialised with a set of negative rewarding grid, positive rewarding grid and hazard zones. The initial parameters for the rewarding would be based on abstract art paintings. We will be pre-processing those images to extract only the edges and shapes of the painting. Then the canvas grid point will be defined accordingly. The next step is to allow the agent to explore the environment while avoiding danger zones, maximize coverage given a set amount of energy, and maximize surface area covered if the agent had a "sprayer" attached.

In order to accomplish this we will be looking at implementing and possibly expanding existing sparse reward exploration based reinforcement algorithms. With methods such as asynchronous advantage actor-critic for learning efficiency and Rewarding Impact-Driven Exploration (RIDE) for intrinsic reward motivation.

2. Concise Summary

2.1 (A3C) Asynchronous Methods for Deep Reinforcement Learning [Mnih et al. 2016] - proposes a conceptually simple and lightweight framework for deep reinforcement learning that uses asynchronous gradient descent for optimization of deep neural network controllers. They show that parallel actor-learners have a stabilizing effect on training and succeed on Atari games, a variety of continuous motor control problems, and navigating random mazes. They take four known algorithms, one-step Q-Learning, one-step Sarsa, n-step Q-Learning, and Advantage Actor-Critic and create asynchronous update methods.

There are two major findings of the paper. One, shows that all async methods achieve substantial speedups from using multiple worker threads, with 16 threads leading to at least an order of magnitude speedup. They believe this is due to the positive effect of multiple threads to reduce the bias in one-step methods. Two, is there is usually a wide range of learning rates that leads to good scores, indicating that all methods are quite robust to the choice of learning rate and random initialization.

Asynchronous optimization works with both value-based and policy-based methods, off-policy as well as on policy methods, and in discrete or continuous domains. They are able to achieve state-of-the-art results with less training time. Extensions of this could include incorporating experience replay into the asynchronous reinforcement learning framework, improvements to the existing methods, and improvements to the neural network architecture.

2.2 (PPO) Proximal Policy Optimization Algorithms [Schulman et al. 2017] - They propose new policy gradient methods that enable multiple epochs of minibatch updates. The objective has clipped probability ratios, which forms a lower bound (pessimistic) estimate of the performance of the policy. To optimize policies, they alternate between sampling data from the policy and performing several epochs of optimization on the sampled data.

Their method is compared against tuned implementations of cross-entropy method (CEM) [Szita and Lorincz, 2006], A2C [Mnih et al. 2016], and A2C with trust region [Wang et al, 2016]. The 3D Humanoid and Atari game environments are used for the experiments. The proposed Proximal Policy Optimization(PPO) methods were shown to have the stability and reliability of the compared methods but are much simpler to implement.

2.3 RIDE: Rewarding Impact-Driven Exploration for Procedurally-Generated Environments [Raileanu and Rocktäschel, 2020] - Exploration in sparse reward environments remains one of the key challenges of model-free reinforcement learning. They propose a novel type of intrinsic reward which encourages the agent to take actions that lead to significant changes in its learned state representation. They evaluate their method on multiple challenging procedurally-generated tasks in MiniGrid. This approach is more sample efficient than existing exploration methods and the intrinsic reward does not diminish during the course of training and it rewards the agent substantially more for interacting with objects that it can control.

In procedurally-generated environments, the agent needs to solve the same task, but in every episode the environment is constructed differently, making it unlikely for an agent to visit the same state twice. Thus, agents in such environments have to learn policies that generalize well. Formally, Rewarding Impact-Driven Exploration is computed as the L2-norm of the difference in the learned state representation between consecutive states. However, to ensure that the agent does not go back and forth they discount RIDE by the number of times that state has been visited. The parameters used to learn the intrinsic reward signal are used only to determine the exploration bonus and never part of the agent’s policy. Otherwise, the agent can cheat by constructing state representations with large distances among themselves.

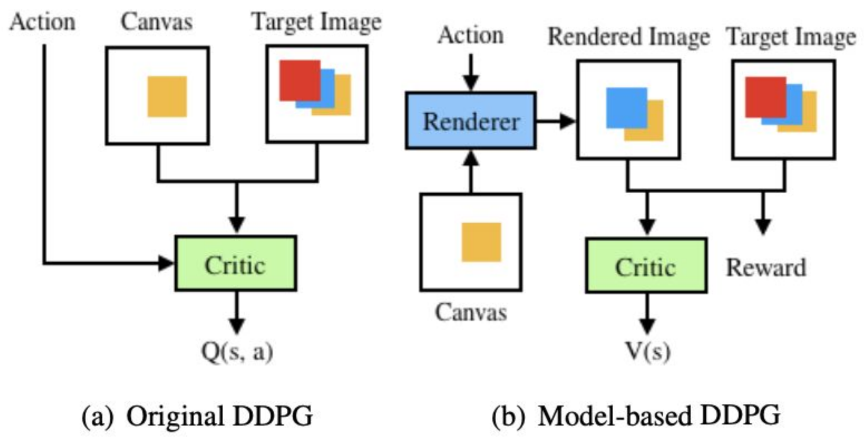
This method is trained and evaluated using Gym-MiniGrid for random procedurally generated environments, Mario, and VisDoom. Experiments show the method attracts agents to states where they can affect the environment and it provides a signal even after training for a long time. Proposed methods to improve the work include considering longer-term effects of the agent's actions, promoting diversity among the kinds of changes the agent makes to the environment, or distinguishing between desirable and undesirable types of impact the agent can have in the environment.

2.4 Gym-MiniGrid [Chevalier et al. 2018] - a minimalistic gridworld environment which seeks to minimize software dependencies, be easy to extend, and deliver good performance for faster training. The environment comes with the existing object types wall, floor, lava, door, key, ball, box and goal. This gives the agent diverse, generated environments with tasks such as getting the key to unlock the door, hazards like lava to be avoided, and the purpose of getting to the goal.

2.5 Learning to Paint With Model-based Deep Reinforcement Learning [Zhewei et al. 2018] - In this paper, the team trained an AI agent that can paint on a canvas to generate a painting that is similar to the target image. Compared to other drawing experiments, the research team made contributions in three areas. First, the drawing agents are able to decompose the target painting into hundreds of strokes in sequence that can be learned over time. Second, the team built a neural network renderers model that can differentiate paintings and strokes, and give corresponding rewards to the agent in the end to end training process. Lastly, the model-based DRL (deep reinforcement learning) agent is general enough to handle multiple types of images (including digits, handwritten, streetview, human portains and etc.).

In the training process, the agent model has three different states representing all the possible information that can be observed in the environment. Agent's action constraints a set of parameters that control the position, shape, color and transparency of the stroke. With different policy functions π, agents will be rewarded at each step by the metric to measure the difference between the current canvas and the target image.

The learning algorithm is expanded from the original Deep Deterministic Policy Gradient. Because drawing a real world painting requires a complex environment that is hard for the agent to model, the team designed a neural renderer so that the agent can observe a modeled environment.



The neural renderer is a neural network consisting of several fully connected layers and convolution layers. It will generate strokes of images differentiable and enables end-to-end training which boosts the performance of the agent.

2.6 On the Complexity of Exploration in Goal-Driven Navigation - [Al-Shedivat et al. 2018] - This paper analyzes the hierarchical design of the agent’s policy and its impact on the exploration capability.

To reduce the uncertainty of a complex and dynamic environment, an agent would use an exploration mechanism. The team tries to understand and measure the complexity of exploration in environments with multiple dependent subgoals, and the effects of hierarchical design of the agent’s policy in such environments. EscapeRoom is the grid-world environment the team designed and generated by a collection of procedurals. Then the agent’s goal space was represented with discrete dependency graphs. When running multiple random experiments, the team can measure complexity of exploration for the agent to walk in the grid space and reach the final goal.

In the experiment, the team tained a few hierarchical and non-hierarchical policies using methods based on proximal policy optimization (PPO). Metrics used for measuring the total payoff includes success rate, the time and steps the agent takes to achieve the final goal. They summarise the result and indicating goals is crucial to enable learning in the environment. In the highly complex environment, the agent generally performs worse and policy hierarchy plays an important role in the agent's learning process. The performance improvement of HiPPO over other flat PPO baselines, which indicate hierarchical policies is beneficial when operating the agent as different temporal scales.

2.7 Creating safer reward functions for reinforcement learning agents in the gridworld - [De Biase et al. 2019] - proposed the Goal-Oriented Action Planning (GOAP) to create a safer artificial intelligence. The team defines a safer reward function as the one that allows an agent to reach its final goal without unsafe function that could potentially damage itself. In order to improve the agent’s behavior without changing the IR algorithm, the team implemented the GOAP into the reward function, and used it to measure the success of the AI decision making.

In the experiment, the research team designed a 2D grid environment, and ran the Advantage-Actor-Critic models to collect the execution logs from training RL agents. To avoid bias, the agent was awarded the same with or without GOAP. The team confirmed the result with promising evidence to show that GOAP would bring potential benefit to the agent in both safety and non-safety scenarios. However, GOAP requires concrete and meaningful goals which might be hard to identify taking only safety in consideration.

The shortcoming of the above summarisation is that it relies exclusively on what the agent is able to perceive at the time the plan is created, meaning the agent is not flexible enough to create more elaborate plans dynamically and adjust the goal in the runtime based on the new observation about the environment. This means, for highly complex scenarios or for an agent with smaller perception, the benefit of implementing GOAP is negligible.

3. Compare & Contrast

3.1 Generating Environments - Using only a singleton environment to train the agent leads to poor generalization sometimes crippling the agent with even small changes to the environment as seen in Raileanu and Rocktäschel. Generating different environments for each training episode requires the agent to learn a general exploration policy. Existing methods to generate these environments in Minigrid involve randomly placing the objects. The location of walls, keys, lava, goals etc can be varied along with the size and number of each room. We will experiment using artwork to define room, wall, and hazard locations. We will extract edges from the artwork and use this to define locations. We believe this is a novel way to train robust agents to any environment.

3.2 Exploration Algorithms & Rewards - PPO and A3C both aim to increase the efficiency of training reinforcement learning agents. PPO attempts this by sampling data and running multiple mini-batch updates to converge the policy faster. A3C does this by training multiple asynchronous agents to reduce the bias in one-step methods. In Schulman's paper both methods are compared with one another and both are shown to do well on humanoid and Atari game environments. The techniques presented in RIDE could be added to existing methods, including PPO or A3C. Their technique adds an intrinsic reward to the agent for state changes and encourages exploration in sparse reward environments. Our work could combine these two approaches and show the effective on our artwork derived environments.

3.3 Success Metrics - Minh and Schulman use similar metrics of the Atari game score, data efficiency, and training time to evaluate their models. Raileanu and Rocktäschel with RIDE use mean intrinsic reward over 100 frames, average frames to solve the level, and average reward per episode. De Biase uses average steps until trained, average steps per episode, and percentage of unsafe actions. Al-Shedivat uses the average success rate of completing the objective and average frames per episode. The trends across all these papers are to have a measure of success (e.g. minimize unsafe actions, reaching the goal, escaping the room, etc.) and a measure of efficiency (e.g. frames per episode, data efficiency, average training steps). Our measure of success will have two measures, one for reaching the goal and two for maximizing the intrinsic reward for exploration. Our measure of efficiency will be the average number of frames per episode.

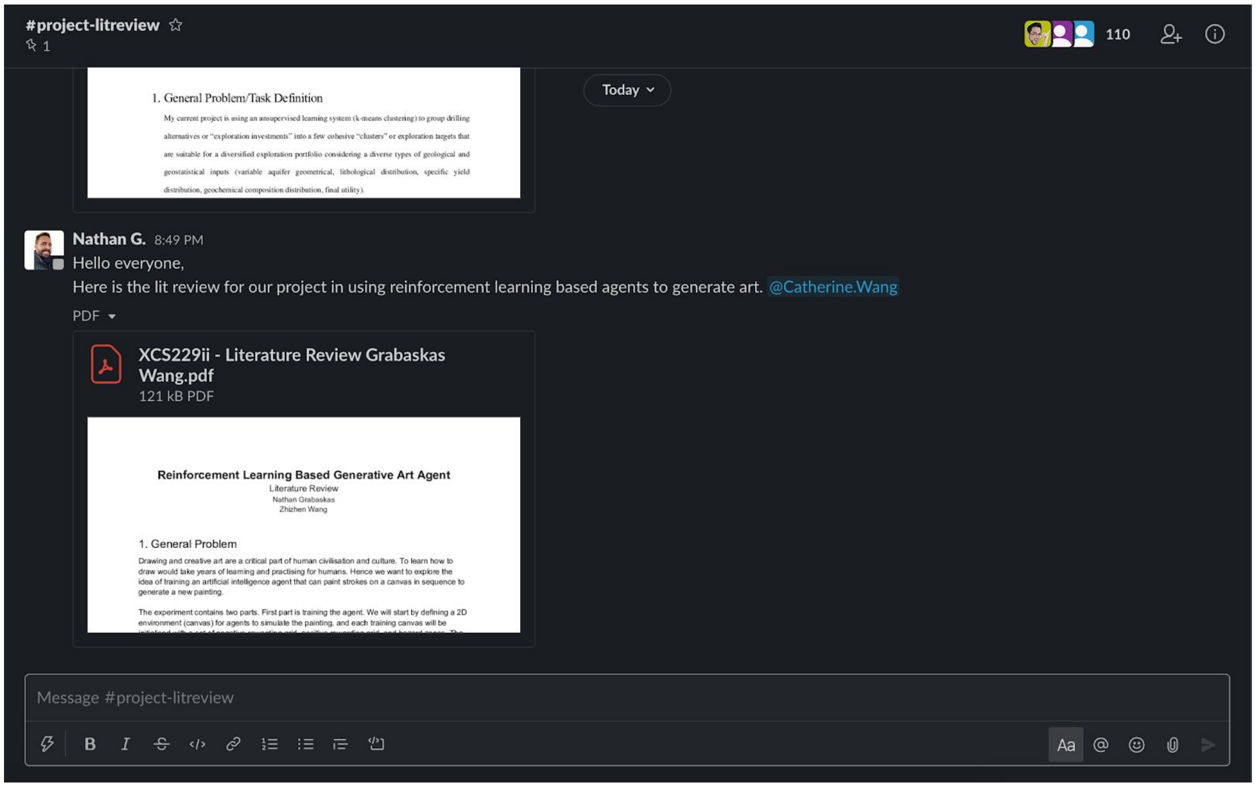
4. Future Work

- Use existing artwork as a method to generate new grid environments to explore.
- Expand A3C/PPO with intrinsic rewards for state changes to encourage exploration of the generated environments.
- Compare existing performance with intrinsic rewards expansion on current generated MiniGrid environments and artwork derived environments. This gives four variants to evaluate.
    - Baseline - existing method with existing generated environments.
    - Variant 1 - existing method with artwork derived environments.

    - Variant 2 - hybrid method with existing generated environments.
    - Variant 3 - hybrid method with artwork derived environments.
- (Optional) Visual how the agent feel intrinsic rewards
- (Optional) Visual how agent behavior evolves over time.

5. References

- Maruan Al-Shedivat, Lisa Lee, Ruslan Salakhutdinov, Eric Xing. On the Complexity of Exploration in Goal-Driven Navigation. arXiv preprint arXiv:1811.06889. 2018.
- Andres De Biase and Mantas Namgaudis. Creating safer reward functions for reinforcement learning agents in the gridworld. University of Gothenburg Chalmers University of Technology. Sweden 2018.
- Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.
- Zhewei Huang, Wen Heng, Shuchang Zhou. Learning to Paint With Model-based Deep Reinforcement Learning. arXiv preprint arXiv:1903.04411 . 2019.
- Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In International conference on machine learning, pp. 1928–1937, 2016.
- John Schulman, Filip Wolski, Praffula Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. 2017.
- Istvan Szita and Andras Lorincz. Learning Tetris using the noisy cross-entropy method. Neural computation 18.12, pp. 2936–2941. 2006.
- Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. International Conference on Learning Representations, 2020.
- Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample Efficient Actor-Critic with Experience Replay. arXiv preprint arXiv:1611.01224. 2016.



#project-litreview

1

110

1. General Problem/Task Definition

My current project is using an unsupervised learning system (k-means clustering) to group drilling alternatives or "exploration investments" into a few cohesive "clusters" or exploration targets that are suitable for a diversified exploration portfolio considering a diverse types of geological and geostatistical inputs (variable aquifer geometrical, lithological distribution, specific yield distribution, geochemical composition distribution, final utility).

Today

Nathan G. 8:49 PM

Hello everyone,

Here is the lit review for our project in using reinforcement learning based agents to generate art. @Catherine.Wang

PDF

XCS229ii - Literature Review Grabaskas Wang.pdf

121 kB PDF

Reinforcement Learning Based Generative Art Agent

Literature Review

Nathan Grabaskas

Zhizhen Wang

1. General Problem

Drawing and creative art are a critical part of human civilisation and culture. To learn how to draw would take years of learning and practising for humans. Hence we want to explore the idea of training an artificial intelligence agent that can paint strokes on a canvas in sequence to generate a new painting.

The experiment contains two parts. First part is training the agent. We will start by defining a 2D environment (canvas) for agents to simulate the painting, and each training canvas will be

Message #project-litreview