Transfer and Representation Learning with Memory Evaluation for Deep Reinforcement Learning

Unique Divine, Erik Skalnes
Columbia University in the City of New York
{u.divine, eas2301}@columbia.edu

Abstract

In this semester project, we experiment with transfer learning and deep reinforcement learning with raw image inputs in partially observable environments. To accomplish this task and allow full flexibility with the environment, we develop a new image-based, sparse-reward environment to iterate on new ideas. We make use of REINFORCE (Williams, 1992), a Monte-Carlo policy gradient algorithm, for preliminary training. In addition, we implement modules for the masked contrastive unsupervised representation learning algorithm, M-CURL (Zhu et al., 2020), and experiment with enhancing our convolutional image encoder using an unsupervised auxiliary task. Additionally, we introduce some novel approaches for utilizing the context learned by Transformer modules to optimize the replay buffer and potentially further improve sample efficiency (Note, these experiments are still a work-in-progress). All code for our project, including the custom environment and all neural network systems, is available at https://github.com/eskalnes/RL_memory/tree/main/rl_memory.

1 Introduction

We are interested in developing RL agents that generalize well. Deep reinforcement learning (RL) as whole suffers from a sort of bottleneck, where even the most impressive breakthough systems of our time such as AlphaZero (Silver et al., 2017), Deep Q-Network (Mnih et al., 2015), and MuZero (Schrittwieser et al., 2020) take upwards of a few hundred thousand or sometimes millions of episodes to converge to peak performance. The goal with this project is to use colored image inputs like the DQN paper for Atari except with greater sample efficiency and using lighter network modules. One of the main ways we do this is by learning predictive low-dimensional representations to speed up training.

Transfer Learning Environment:

For the purposes of our project, it was important to start out with a humble baseline for an environment that would still be relevant for the experiments we want to perform in both transfer learning and novel contributions to the processing of memories for RL optimization. Atari was too complex and many AI Gym environments in OpenAI are state-based and/or written in such a way that they are not easy to customize. So, we decided early on to implement a customizable environment that could allow both state (as in "board game state") and image-based training, multiple goals, multiple failure criteria, randomized starts, and even multiple agents. We came up with a simple, maze-like game (Fig. 2), where an agent tries to reach a goal for a positive reward without falling in a hole.

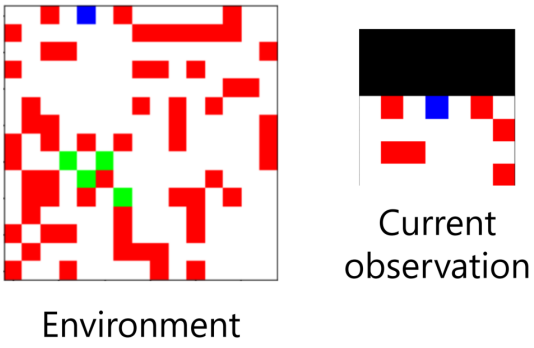

Figure 1: An example transfer learning environment. Here, the agent is the blue square and moves around in a 2-D space in any direction. If the agent steps in a hole (red), it dies, ending the episode, and producing a negative reward. If the agent reaches a goal (goal), it instead receives a positive reward.

This environment has undergone robust unit and integration testing so that it could be maximally useful as an open-sourced tool. It's available in our repo under rl_memory/custom_env and functions with a similar API to that of the environments in Open AI Gym.

Contrastive Learning: Contrastive learning is a self-supervised algorithm for representation learning developed in the computer vision community. It's an instance-level pre-training technique to get an encoder (Chen et al., 2020), which here refers to a map from high-dimensional inputs (video, images, sound, etc.) to low-dimensional representations. Contrastive learning is particularly useful in the spare-reward setting because rewards often act as labels in deep reinforcement learning. Thus, self-supervised techniques that don't require labels can aid in improving sample efficiency(Laskin et al., 2020). Self-supervised pre-training, just like word vectors, gives us a great starting point to vectorize an image into a semantically relevant geometric space. It's been a game changer in the computer vision world since around 2018 (Oord et al., 2018).

Masked Training: In the Bidirectional Encoder Representations for Transformers (Devlin et al., 2018) paper, popularly known as "BERT", the authors found much success from the use of a masked language model. The basic idea was that some percentage of the input sequence is masked, and a transformer leverages its context in order to eventually predict the masked elements.

Masked Contrastive Learning: Masked contrastive learning blends the idea of the masked language model with that of contrastive learning for training image encoders. With the M-CURL algorithm, (Zhu et al., 2020) surpassed the current state of the art (CURL algorithm (Laskin et al., 2020) on many reinforcement learning tasks that use image inputs: 14 out of 16 environments from DMControl suite and 21 out of 26 environments from Atari 2600 Games.

2 What We Did: Methods & Results

Image based RL tasks are often framed as partially observable Markov decision processes (POMDPs), which can be described by the tuple $(\mathcal{O}, \mathcal{A}, p, r, \gamma)$, where

- $\mathcal{O}$ represents observations, a collection of images rendered from the environment, and
- $\mathcal{A}$ is the action space,

Hence, $o_t \in \mathcal{O}$ represents the rendered image at time t and $a_t \in \mathcal{A}$ denotes the action.

We can use observations $\{o_t\}$ as states for model-free RL on the simple environment (Fig. 2), i.e. $s_t = o_t$, however it is often useful to work with several video frames or stacks of images so that temporal information can be encapsulated in the state representation. Following the example of preprocessing in the DQN paper (Mnih et al., 2015), we also sometimes utilize a stack of K (> 1) consecutive observations as the input to capture more information. This sequence of observations makes a state,

$$s_t = (o_{t-K+1}, o_{t-K+2}, \cdots, o_t).$$

The collection of all states $\{s_t\}$ is denoted by $\mathcal{S}$. A convolutional netwok (ConvNet) encoder f_θ with weights θ is used to map states $s_t \in \mathcal{S}$ into lower-dimensional representations.

2.1 Policy Gradient

Our first experiment employs a paradigm similar to that of the first successful application of deep policy gradient algorithms for deep reinforcement learning in POMDPs, which to our knowledge was in the Minecraft control paper (Oh et al., 2016).

We implement vanilla REINFORCE on a simple environment to show that it works. Then we pre-train a ConvNet encoded policy network with a simple environment in order to solve a much more difficult environment through transfer learning. We tried multiple configurations to no avail but eventually trained a performant agent with three changes:

1. Small reward-shaping that punished an agent for staying still helped urge exploration and caused the agent to find more goals.
2. Holding the environment fixed each episode completely prevented the policy from understanding features that dealt with relative position, so the agent would inevitably converge to picking the same action each time. To fix this, we forced transfer learning by starting the agent on different environments each episode.
3. Over-saturating the reward signal didn't work with either extreme. If there were too many goals or too many holes, the agent would simply perform quasi-random actions. We found that a small 3 by 3 environment with one goal and one hole gave the fastest performance increase during training.

The combination of these changes made an encoder and policy network that could succeed in

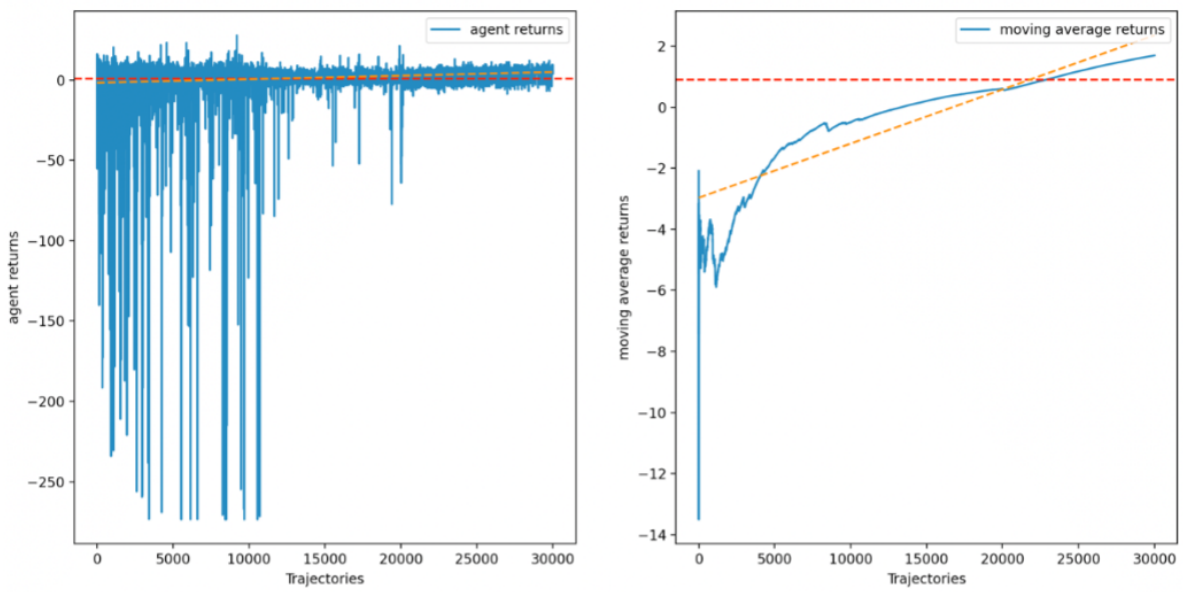

Figure 2: Our agent succeeds to transfer learn onto new environment.

more challenging, extreme environments. We show that the representation learned in the simple environment allows the agent to solve an environment (Fig. 2) that is effectively unsolvable with policy gradient methods after approximately 30,000 episodes of training on the 3 by 3 environment . Policy gradients fail in sparse reward environments, where the likelihood that an agent experiences a positive reward is so small that it converges to a policy that learns to avoid negative rewards instead, as described in (Kakade and Langford, 2002). We conclude that pre-training a representation is essential for improving sample efficiency in difficult environments.

The point here is that a feature-rich representation of the agent state is essential to discovering good policies. To that end, we show that a policy gradient algorithm that uses a baseline that depends on having a good representation of state can solve problems even in the complicated domain of combinatorial optimization. We submit that this is a proof-of-concept for this type of memory-based algorithm in visual learning domains, where reinforcement learning algorithms will have to learn the representation, as one is not always readily available.

2.2 Masked Contrastive Learning

For the purposes of further enhancing an image encoder without additional RL episodes, we implemented a similar style of framework to that used in Masked Contrastive Representation Learning for Reinforcement Learning (Zhu et al., 2020). **Note**, a concise yet informative description of the method is included in the attached poster file. At the time of us writing this report, we're trying to get the image encoder f_θ to learn from unsupervised pre-training on simulated episodes that are made (1) from random action sequences and (2) from previous trajectories saved from the policy gradient experiments. As it currently, stands, we have not benchmarked the effectiveness of masked contrastive learning, but this will continue in current/future work.

3 Ongoing & Future Work

We state some of our plans for the rest of the project because we intend to work to a publishable result. Since our training systems have managed to master this simple environment, we'll now test out other model-free algorithms in combination with the masked contrastive learning paradigm to see if we can increase sample efficiency even more.

Additionally, we'll now benchmark on more challenging environments. It seems that the DM-Control suite, robotic control tasks, and Atari games are the standard for testing image based RL systems, so these are options we're considering. There are two, key experiments we intend to conduct that have largely gone unexplored.

1. Using the Transformer's sequence context to prioritize sequences on the buffer.

2. Use determinantal point processes to diver-

sify the replay buffer and see if that improves training (credits to Professor Krzysztof Choromanski for this idea).

The poster that was submitted with this report has more details on the future experiments as well.

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sham Kakade and John Langford. 2002. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Junhyuk Oh, Valliappa Chockalingam, Honglak Lee, et al. 2016. Control of memory, active perception, and action in minecraft. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Jinhua Zhu, Yingce Xia, Lijun Wu, Jiajun Deng, Wengang Zhou, Tao Qin, and Houqiang Li. 2020. Masked contrastive representation learning for reinforcement learning. *arXiv preprint arXiv:2010.07470*.