Transfer Learning between Different Levels in the Same Game

Yirong Tang
Department of Computer Science
Tufts University
Medford, MA 02155
yirong.tang@tufts.edu

Shucheng Tian
Department of Computer Science
Tufts University
Medford, MA 02155
shucheng.tian@tufts.edu

Abstract

Improving generalizability has been a hot topic in reinforcement learning. In this report, we experimented transfer learning tasks between two gaming scenario that share the same gaming mechanism and strategy but have different raw pixel image representation. We found that reinitializing first convolutional layer to learn new raw pixel image representation and retraining on all transferred layers to associate image representation to previously trained gaming mechanism and strategy can boost the performance the most.

1 Introduction

1.1 Background

Reinforcement learning (RL) Sutton and Barto [1998] is an area of machine learning concerned with how an agent seeks to maximize long-term rewards through experience in its environment. While progress has been achieve in improving learning for single, the emphasis on generalization of learning results has just been placed. In RL realm, the approach to achieve generalization is often referred as transfer learning, an idea that generalization can occur not only within tasks, but also across tasks.

1.2 Motivation

Deep reinforcement learning is the result of applying deep neural networks to approximate the state value from numerous features, and it is especially useful for approximating the state value from image. Although Deep Reinforcement Learning has managed to achieve state-of-the-art results in learning control policies directly from raw pixels, it fails to generalize. Using the different scenarios in VizDoom with Deep Q-Learning, we demonstrate the difficulty of a trained agent in adjusting to simple modifications in the raw image, ones that a human could adapt to trivially. Therefore, we want to explore how to transfer the knowledge learned in Deep Q-learning, which tries to learn an optimal policy from its history of interaction with the environment.

1.3 Hypothesis

We hypothesized that with the proper way of transferring the knowledge in DQN, for two scenarios that had the same fundamental mechanism but different visual representation, training on one scenario could boost the performance when training on the other scenario in terms of higher average return in each episode as well as faster convergence.

Preprint. Work in progress.

1.4 Major Ideas and Results

After the DQN first trained on *basic* scenario, we tried to transfer to *rocket basic* scenario. First, we experimented with directly used the model we trained for *basic* scenario to train on *rocket basic* scenario. Then, we experimented with re-initializing the first convolutional layer of the previously trained model to train on *rocket basic* scenario. The results showed that the approach of just re-initializing the first convolutional layer boosted the performance in terms of higher average return in each episode as well as faster convergence.

2 Related Works

In DeepMind's paper "Playing Atari with Deep Reinforcement learning" Mnih et al. [2013], researchers developed a single Deep Q-Network (DQN) that is able to play multiple Atari games, in many cases surpassing human expert players. The model consists of convolutional layers followed by fully connected layers. The model takes in the raw image pixels of a game and outputs Q-values estimating future rewards. The model is trained with a variant of Q-learning with a target network and experience replay. The target network helps reduce oscillations or divergence of the policy, and experience replay removes correlation in the observations and prevents the model from getting stuck in bad local minima.

For transfer learning, using the Atari game Breakout, Gamrian and Goldberg [2018] demonstrate the difficulty of a trained agent in adjusting to simple modifications in the raw image, ones that a human could adapt to trivially. It is shown in the paper that using various forms of fine-tuning, a common method for transfer learning, is not effective for adapting to such small visual changes. The paper uses Unaligned Generative Adversarial Networks (GANs) to create a mapping function to translate images in the target task to corresponding images in the source task that allow us to transform between various variations of the Breakout game, as well as different levels of a Nintendo game, Road Fighter.

3 Technical Approach

3.1 Deep Q-Learning

We used Deep Q-learning convolutional neural network models with greedy exploration. Greedy exploration is quite important in the context of Deep Q-learning as it has so many states, the learning algorithm is very easily to be stuck in exploring the same seemingly optimal path. At each step, the agent chooses a random action with probability p or picks the action that has the highest value of state-action pair following the value approximation with probability $(1 \; p)$. The p decays over time, so initially we want the agent to explore more and to understand the environment, but after some time we assume that we have enough knowledge about the environment so we want to follow what we have learned with only little exploring. We also use experience replay, providing the model with signals throughout its training history, and target networks to avoid oscillations and divergences in policy.

3.2 Transfer Learning

Transfer learning is a technique that improves learning in a new task through the transfer of knowledge from a related task that has already been learned. In this case, we use convolutional neural networks that have performed very well at value approximation for one scenario and try to adapt them, with a little more training, to estimate Q-value for another scenario.

We want to mainly explore transfer learning in VizDoom from *Basic* scenario to *Rocket Basic* scenario. Our hypothesis is that because the two scenarios have same objectives of killing target, transferring learned model will help speed up convergence and perhaps attain better performing models than before.

3.2.1 Base Line Method

Transfer all layers from the *Basic* scenario and retrain on all layers on *Rocket Basic* scenario.

3.2.2 Focus Method

Transfer all layers from the *Basic* scenario, reinitialize first convolutional layer, and retrain on all layers on *Rocket Basic* scenario.

4 Experiments and Results

4.1 Game Mechanism

ViZDoom is a Doom-based AI research platform for reinforcement learning from raw visual information and it allows developing AI bots that play Doom using only the screen buffer. Kempka et al. [2016]

4.1.1 Basic Scenario

Basic scenario is the default scenario of VizDoom. It involves a single player who controls the actions of VizDoom Marine to eliminate the target using a pistol. States are represented as raw pixel images. Three actions are available: move left, move right, and shoot. For each time step it survives, the VizDoom Marine receives an -1 reward; for each missing shoot, the VizDoom Marine receives an -6 reward; for each success in killing the target, the VizDoom Marine receives an +100 reward. Each action takes one time step to complete, and for each episode, there is a timeout at 300 time steps.

4.1.2 Rocket Basic Scenario

The *Rocket Basic* scenario has the same fundamental mechanism as *Basic* scenario in state representation, actions available, and rewards. But instead of pistol, VizDoom Marine use a rocket launcher in this scenario, and has a shorter distance from the target. This means that we can apply a technique called *flick*, which means that if we launch the rocket during the process of moving at one direction, the rocket will move towards the direction of our movement even after it has been launched.

Figure 1: Basic Scenario

Figure 2: Rocket Scenario

4.2 Basic Scenario vs. Rocket Basic Scenario

We can see that at *Basic* scenario, the reward is lower than *Rocket Basic* scenario at the very beginning, but reward at *Basic* scenario grows fairly fast and surpass reward at *Rocket Basic* scenario around xx episodes. We wanted to see if transfer learning from *Basic* scenario to *Rocket Basic* scenario would speed up convergence and improve the average reward.

4.3 Transfer All Layers vs. Transfer Partial Layers

At first, we tried retraining all layers without reinitializing any layers. We find that although the improvement on total reward gained throughout each episode is not significant, transfer learning already achieve a slight faster convergence.

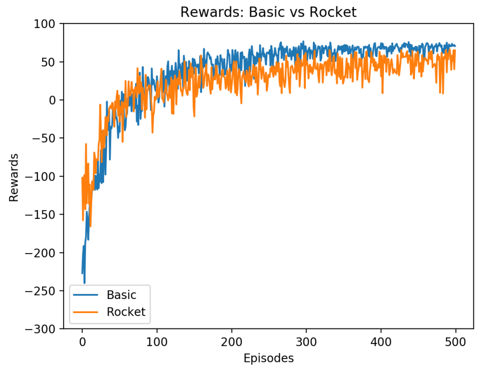

Figure 3: Rewards: Basic vs. Rocket Basic

Then, we tried transferring partial layers, that is, reinitializing only the first layer and retraining all layers. We found that this method perform significantly well - rewards seem to take less episodes to converge, and average reward in the transfer learning context is consistently higher than training from scratch. Compared with transferring all layers, the performance is overall better in faster convergence and higher average rewards.

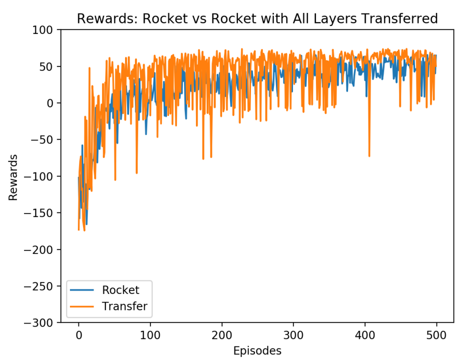

Figure 4: Rocket Basic vs. Rocket with All Layers Transferred

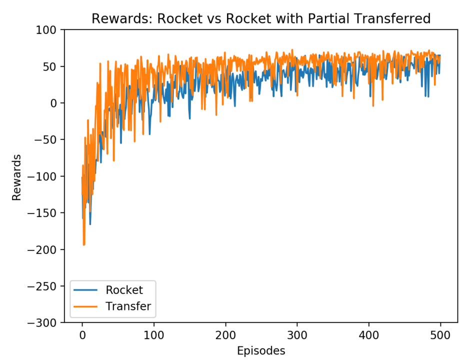

Figure 5: Rewards: Rocket vs Rocket with Partial Layers Transferred

4.4 Discussion

The experiment result confirmed our early hypothesis - for two scenarios that have the same fundamental mechanism but different visual representation, training on one scenario can boost the performance. Also from the comparison between *Transfer All Layers and Transfer Partial Layers*, we find that reinitializing first convolutional layer performs much better than transferring all weights from previously trained model. We believed that it was due to the characteristics of Convolutional Neural Network. Layers that are close to the fully-connected layer and the final action layer tend to have specific features that are applicable to the game trained on, while earlier layers are convolutional layers which extract visual features. So when we have two scenarios that have the same mechanism in playing game but slight different representation in raw pixel image, it would be useful to keep later layers that store information about how to play the game and reinitialize earlier layers so that they can learn specific visual representation and connect the newly learned visual representation with the previously learned gaming strategy.

5 Conclusion and Future Work

From our experiment result, we found out that for two scenarios that have the same fundamental mechanism but different visual representation, training on one scenario can boost the performance, and reinitializing first convolutional layer performs much better than transferring all weights from previously trained model.

In the future, we are interested in seeing if we can successfully use the same techniques to other games that have same mechanism but different raw pixel image representation in different levels, such as Super Mario to improve performance – if our future work succeed, it would indicate that transfer learning can be more applicable in the context of games for deep reinforcement learning.

References

Shani Gamrian and Yoav Goldberg. Transfer learning for related reinforcement learning tasks via image-to-image translation. *arXiv preprint arXiv:1806.07377*, 2018.

Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. ViZDoom: A Doom-based AI research platform for visual reinforcement learning. In *IEEE Conference on Computational Intelligence and Games*, pages 341–348, Santorini, Greece, Sep 2016. IEEE. URL `http://arxiv.org/abs/1605.02097`. The best paper award.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.