Accelerating Deep Reinforcement Learning via Imitation Learning

Xiao Lei Zhang
York University
Toronto, Canada
zhang205@cse.yorku.ca

Anish Agarwal
University of Waterloo
Fremont, California
a22agarw@outlook.com

Under the umbrella of unsupervised learning there are two categories of training: imitation learning and reinforcement learning. In imitation learning the machines learns by mimicking the behavior of an expert system whereas in reinforcement learning the machine is given directed via direct environment feedback. Traditional deep reinforcement learning takes a long time before the machine starts to converge to an optimal policy. This paper proposes a method by which deep reinforcement learning convergence can be accelerated by using imitation learning as the initial training process in deep reinforcement learning.

An implementation of this project is located at: https://github.com/veda-s4dhak/Cartpole-Imitiation-Learning

Keywords—imitation learning, deep reinforcement learning, deep q-learning, behavioral cloning

I. Introduction

Imitation learning in deep learning system is focused on modeling the behavior of an expert system or focused on modeling the reward function which best approximates the expert system's behavior.

The performance of an imitation learning system alone is limited by the performance of the expert player. In the ideal sense we want systems which can increase the upper bound of performance by going beyond that of an expert system. To achieve this, imitation learning alone is not sufficient. We are bounded by the limits of supervision.

Current fully unsupervised deep learning systems which have performance beyond known expert systems have been designed using reinforcement learning, where machines are learning from direct environment interaction. Deep reinforcement learning however requires quite a bit of training period till the model reaches expert level performance. This is exacerbated in increasingly complex environments.

It seems that there is a mutual advantage to augment reinforcement learning with imitation learning. A reinforcement learning model can accelerate its initial training time by imitating an expert system. An imitation learning model can increase its upper bound and go beyond the expert system by switching to direct environment interaction.

In this paper we consider this augmentation. We use traditional imitation learning approaches as a precursor to deep reinforcement learning. A deep neural network first imitates an expert system and then is allowed to reinforce directly through the environment. We first setup the framework of the experiment, followed by the implementation details and concluding with the experimental results.

The variation of imitation learning we are applying is similar to Behavioral Cloning via Forward Training[1]. We modified the forward training method to use a stationary policy rather than a non-stationary policy. We use batching and random sampling for generating training data for imitation. Lastly our training

Iterating from 1 to T?
Stationary vs Non-stationary policy?

How is the train classifier exactly defined (in the sense of forward training in the paper)?

For both training portions, imitation learning and reinforcement learning, a standard Deep Q-Learning model (DQN) is used. The DQN first computes imitation error during the imitation learning portion and subsequently computes the reinforcement error during the reinforcement learning portion.

II. Framework

The set of initiation learning approaches can be classified into two categories: behavioral cloning and inverse reinforcement learning. Behavioral cloning as the name says focuses on making an deep learning system directly model an expert system. Inverse reinforcement learning on the other hand focuses on making a deep learning system model the reward function which the expert system is trying to optimize.

For the purposes of this experiment, behavioral cloning is preferred. This is because the training environment under consideration in this paper has a well defined reward function and is thus more conducive to behavioral cloning.

A. *Markov Decision Processes*

We define the MDP parameters as {S, A, P, R, I}, where:

- S is the set of states
- A is the finite set of actions
- P = P(s, a, s') is the state transition probability which denotes the probability to transition to state s' give than the previous state was s and action a was taken.

- R = R(s,a) is the reward in state s given action a was taken
- I is the initial state distribution

Additional parameters are specified as follows:

- N denotes the number of episodes
- T denotes the time horizon
- π denotes the policy that determines which action is taken at state s
- $\pi^{'}$ denotes the experts policy
- $\pi^{''}$ denotes the optimal policy

Given the above we can make the following conclusions:

- $V(\pi)=T*E\ R(s,\pi(s))$

  denotes the total rewards of all trajectories given the initial state I
- $V(\pi^{'})-V(\pi)$ denotes the imitation regret
- $V(\pi^{''})-V(\pi)$ denotes the reinforcement regret
- $V(\pi^{''})-V(\pi^{'})$ denotes the expert regret

The goal of augmented reinforcement learning is to accelerate reduction of reinforcement regret to the point where it is below expert regret. That is using imitation learning to reach the point where

$$V(\pi^{''})-V(\pi)\leq V(\pi^{''})-V(\pi^{'})$$

as fast as possible.

III. Implementation

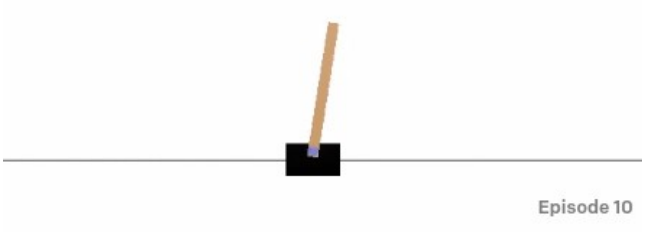
A. Agent-Environment Interaction

We implement our experiment in CartPole-v1 Gym environment from the OpenAI gym. CartPole is a conventional controls problem and is suitable for imitation learning since an expert model is readily available in the form of PID controllers.

The mechanics of CartPole consist of a pole attached by a joint to a cart, which is controlled by applying a force of +1 or -1. The pole initially starts upright and the goal is to prevent the pole from falling over. Each episode ends, when the pole is more than theta degrees from the vertical or the cart moves more than 2.4 units away from the center. For this experiment, we set theta to 50 degrees to reduce learning time of each agent.

Figure 1. View of Cartpole-v1 environment.

We first train the model via imitation learning by modeling the PID. Then we train the model using deep reinforcement learning directly from the environment. We compare these results to a model trained via deep reinforcement learning alone.



In both imitation learning and deep reinforcement learning we used the q-learning methodology for training. During the imitation learning process, the q-learning model optimizes the reward based on following the expert input. The expert input was taking by implementing a simple PID controller to control the cart. The PID system was tuned to score much higher than an average human player. The proportional, integral and derivative parameters are as follows:

$$P = 0.6,\ I = 0.00625,\ D = 0.8$$

The reward during imitation learning is a Gaussian function defined below. The reward depends on the difference between the expert action and the model action as well as the difference between the optimal and actual pole angles. The reward function is highest when the pole angle is optimal and the model action matches the expert action.

$$R(\theta,a_{PID},a_{model})=0.2e^{-\frac{1}{2}\left(\frac{\theta_{optimal}-\theta}{\sigma_1}\right)^2}+0.8e^{-\frac{1}{2}\left(\frac{a_{PID}-a_{model}}{\sigma_2}\right)^2}$$

$$\theta_{optimal}=0\wedge\sigma_1=10\wedge\sigma_2=0.5$$

For deep reinforcement learning the Gaussian reward function becomes

$$R(\theta)=e^{-\frac{1}{2}\left(\frac{\theta_{optimal}-\theta}{\sigma_1}\right)^2}$$

$$\theta_{optimal}=0\wedge\sigma_1=10$$

This reward function is based on the difference between the target and actual pole angles

We define the loss function for the model as follows:

$$L_i(\theta)=E_{a\sim u}[(y_i-Q(s,a;\theta_i))^2]$$

where

$$y_i=E_{a'\sim\pi}[r+\gamma max\,Q(s',a';\theta_{i-1})|S_t=s,A_t=a]$$

$$L_i(\theta_i)=\mathbb{E}_{a\sim\mu}\left[(y_i-Q(s,a;\theta_i))^2\right]$$

$$\text{where}\quad y_i:=\mathbb{E}_{a'\sim\pi}\left[r+\gamma\max_{a'}Q(s',a';\theta_{i-1})|S_t=s,A_t=a\right]$$

B. Model Architecture

The model architecture is shown in Figure 1.We use a fully connected neural network model for the Deep Q-Learning and Imitation Learning agents. We use ReLU activation for the inner layers and linear activation for the output layer.

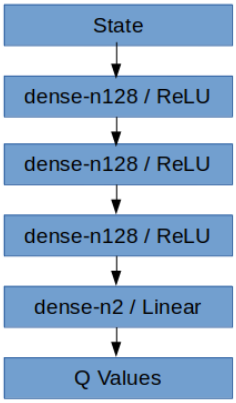


Figure 1. DQN Model Architecture. Dense layers are described by number of features (n).

C. Imitation Training Methodology

For imitation training we use the forward training methodology used by Ross and Bagnell (2010) [2] modified to use a stationary policy. We use a stationary policy since the T is unbounded in the training environment.

The training methodology is summarized as follows:

Algorithm 1: Imitation Training

1: Initialize π

2: **For** I = 1 to num_epochs **do**

3: Execute x trajectories using π'

4: Sample dataset D = {states, action} taken by expert

5: Train π using DQN[2] with $Reward = R(\theta, a_{PID}, a_{model})$

6: **End for**

7: **Return** π

D. Reinforcement Learning Methodology

The reinforcement training is similar to imitation training except that the training classifier uses a reward function directly from the environment rather than the expert player.

Algorithm 2: Reinforcement Training

1: Initialize π

2: **For** I = 1 to num_epochs **do**

3: Execute x trajectories using π'

4: Sample dataset D = {states, action} taken by expert

5: Train π using DQN[2] with $Reward = R(\theta)$

6: **End for**

7: **Return** π

IV. Results

A. CartPole-v1 with Deep Q Learning

The first model is trained using deep reinforcement learning alone, denoted as RL500. The model loss and reward curves are shown in Figure 2. The summary results are shown in Table 2. The model achieves an average score of 331.63 and a peak reward of 1949.39 after 500 episodes of training. Figure 3 shows the model weights across the 4 layers of the model in normalized indices.

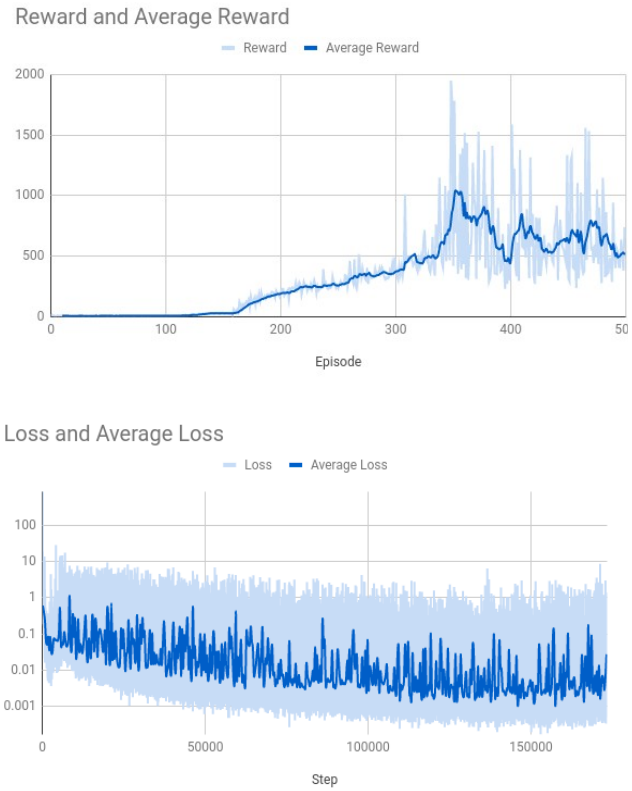
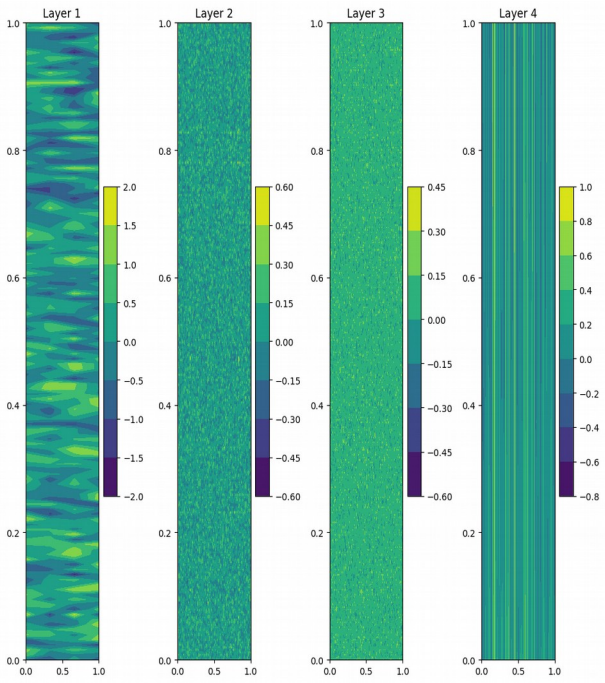


Figure 2. Model Loss and Reward for CartPole-v1 with Deep Q Learning, denoted as RL500.



B. Expert Player (PID) with Behavioral Cloning

The second model is trained by imitation learning via the expert PID system, referred to as IL250. Figure 4 shows the loss and reward of the expert. The model imitation loss after 250 episodes of training is also shown in Figure 4. Table 2 shows the expert producing an average score of 593.1 and a peak score of 6082.91. Model weights are shown in Figure 5.

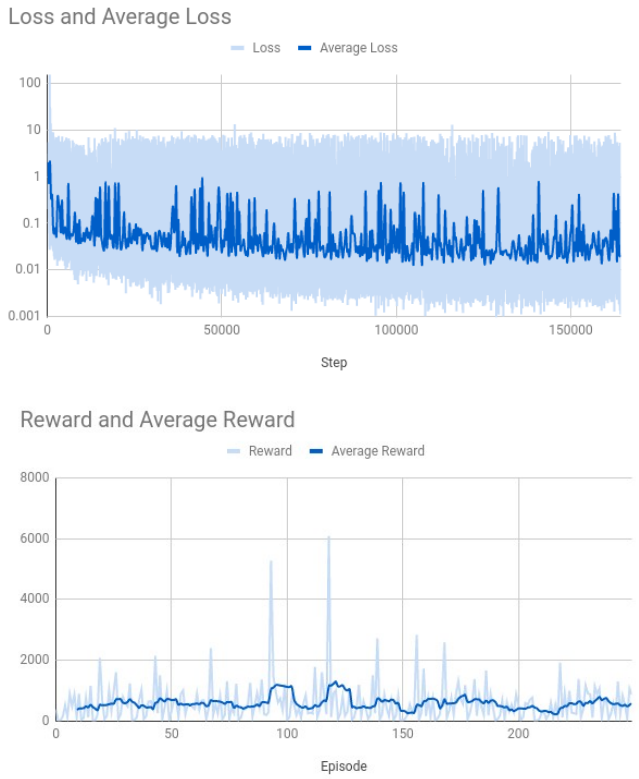


Figure 4. Model Loss and Reward for Behavioral Cloning using PID expert player, denoted as IL250.

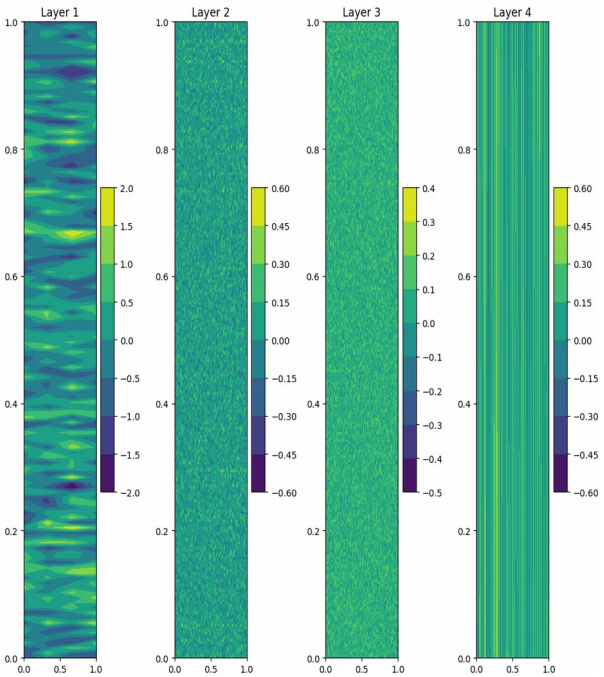


Figure 5. Model Weights for IL250 Model.

C. Deep Reinforcement Learning after Behavioral Cloning

We train the third model using Deep Q Learning with the imitation learning model from B. This model is referred to as IL250 + RL250. Model loss and reward plots are shown in Figure 6. As shown in Table 2, the model achieves an average reward of 2000 and a peak reward of 13000 after 250 episodes of training. The reward of this model is much higher than RL500. Model weights are shown in Figure 7.

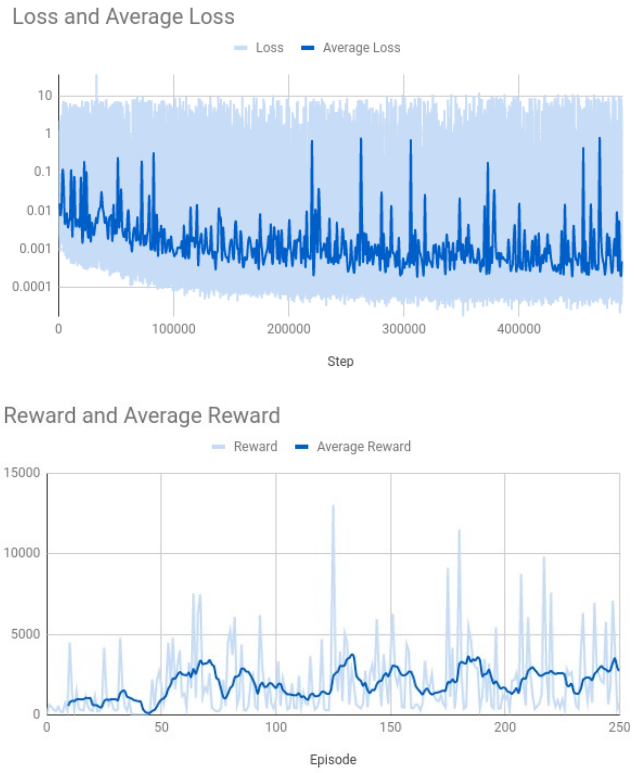


Figure 6. Model Loss and Reward for CartPole-v1 with Deep Q Learning after Behavioral Cloning, denoted as IL250 + RL250.

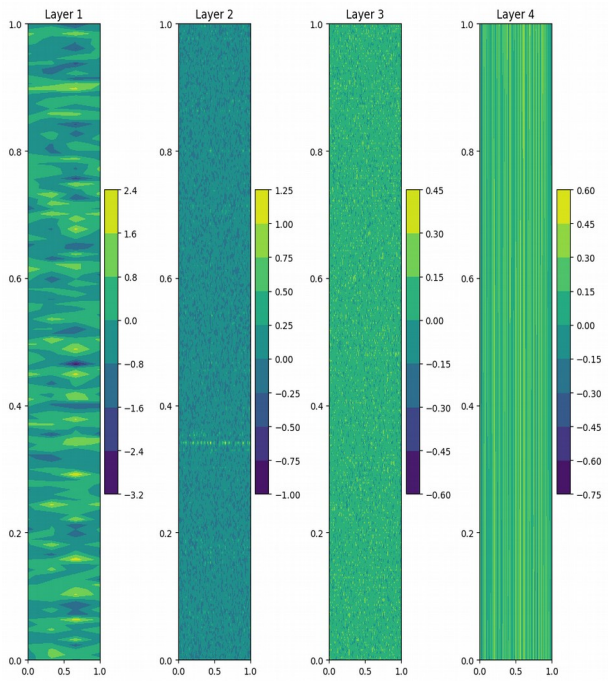


Figure 7. Model Weights for IL250 + RL250 Model.

D. Comparison of Rewards

	CartPole-v1
RL500	331.63
IL250	593.1
IL250 + RL250	**1937.61**
RL500 Best	1949.39
IL250 Best	6082.91
IL250 + RL250 Best	**13043.02**

Table 2. Comparison of average reward and best reward for each learning method. Best reward measures the single best performing episode for each learning method.

V. Conclusion

We have trained a model using imitation learning via PID agent followed by deep reinforcement learning (RL250 + IL250) and compared the result with a model trained only using deep reinforcement learning (RL500). RL250 + IL250 achieves significantly higher reward than RL500. Future works include testing this method on other Gym environments and measuring its performance on other deep learning model architectures such as CNN, RNNs.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*