Enduro : Imitation Learning

Dhruva Kashyap, Sumukh Aithal K

07-10-2021

Contents

1 Introduction

Reinforcement Learning has been very successful in Video games. The agent is able to beat humans in challenging games like Go (AlphaGO) and Starcraft.

In Imitation Learning, the model learns from the actions taken by the human. Imitation Learning can be seen as supervised learning problem with the data being generated by the expert human. The model tries to "imitate" the human by learning what action the human has taken at a particular state. In essence, the model tries to learn the mapping between observation space and actions by looking at the data. This mode of learning is essential in sensitive tasks where simulation is hard.

2 About the Environment

Enduro consists of maneuvering a race car in the National Enduro, a long-distance endurance race. The object of the race is to pass a certain number of cars each day.

Doing so will allow the player to continue racing for the next day. The driver must avoid other racers and pass 200 cars on the first day, and 300 cars with each following day. An episode ends after 150 seconds per level. The game also ends and resets if the player reaches 999.99KM.

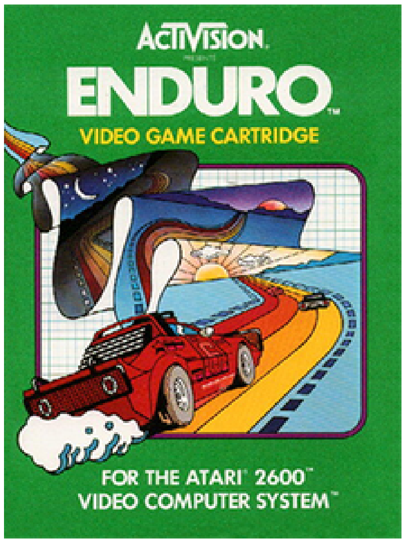

Figure 1: Activision Enduro Poster [7]

Figure 2: Enduro Gameplay [8]

2.1 Observation Space

In this environment, the observation is an RGB image of the screen, which is an array of shape (210, 160, 3). Each action is repeatedly performed for a duration of k frames, where k is uniformly sampled from {2,3,4}. The game is running at 30fps.

2.2 Action Space

A total of 9 actions are defined in this environment.

The actions are as follows

1. NOOP
2. FIRE/ACCELERATE
3. RIGHT
4. LEFT
5. DOWN/DECELERATE
6. DOWNRIGHT
7. DOWNLEFT
8. RIGHTFIRE
9. LEFTFIRE

2.3 Reward

In the gym[3] package a reward of +1 is given for each car passed and -1 for each car that passes the agent. However, the net reward cannot drop below 0.

3 Dataset

We played the Enduro game and recorded some gameplay for the model to learn. We had three different gameplays with a total of 15 minutes of gameplay. This corresponds of 26000 observations and corresponding actions taken by the “expert” human. Trials t1 and t2 had completed only level 1 and t3 had completed level 2.

Table 1: Number of actions per Game

Name	Duration	Datapoints
t1	4:57	8194
t2	2:43	4911
t3	7:23	13316

The action space and state space is stored in a compressed numpy .npz format, reducing 100s of MB to 10s of MB.

4 Model Architecture

We tried 3 different architectures:

1. SimpleNet (LeNet Architecture)
2. BigNet
3. ResNet18

We started with the LeNet Architecture [1], which has been famously used in MNIST. We started with this architecture as it seemed like a simple starting point.

We moved on to a new architecture we call BigNet, which is a modification to the LeNet architecture but with an additional convolution layer.

Lastly, we train the model on a standard ResNet18 [2] architecture, which is commonly used in image recognition tasks.

Table 2: Number of parameters per model

Model	Number of parameters
Simple	2639801
BigNet	1007193
ResNet18	11284041

5 Training

Enduro is available in the Arcade Learning Environment[4]. We tried with both SGD with Momentum and Adam as our optimizer and tuned the learning rate using a library called optuna[5]. We used optuna to run 100 trials with different learning rates that would minimize the loss, where the loss is a measure of closeness between the model and the expert. Optuna automatically prunes runs and provides the best parameters. Each run was trained for about 50 epochs, this was a balance we observed between training time and performance. A batch size of 64 was used in all the experiments. The model is built on PyTorch[6]. For all the observations, we cropped the image to 160 x 160 and then converted it to a Tensor. We modeled the problem as a 9 class classification (9 actions) problem. Since it is a classification problem, we used the Cross Entropy Loss as our objective function to train the model.

6 Results

It can be seen that the SimpleNet beats around 198 cars in the first level which is impressive. This model took less time to train compared to the ResNet model. The best ResNet model completed level 1 and defeated around 50% of cars in challenging Level 2 of the Enduro game. Our best performing model is BigNet which finishes level 1 and $2/3^{\text{rd}}$s of level 2.

It is important to note that only one of the gameplays had completed Level 2 and two other gameplays had completed Level 1 only. The model reaching the first position is a really positive result showing the effectiveness of the Imitation Learning in the Enduro game. In the model's gameplay, we observed that the model maneuvered the racing track extremely well, overtaking multiple cars with ease. Specifically, during the most challenging time, the night time when the visibility is affected, the model is able to play the game well. When we observed the gameplay of the model, we saw true reflections of ourselves in the model.

We observed that cropping the state space to 160x160, which shows only

Table 3: Results for best runs of different models

Model	Optimizer	Learning Rate	Rank (In Level 1)	Rank (In Level 2)	Number of cars passed
SimpleNet	Adam	10^{-5}	2		198
BigNet	SGD	50^{-3}	1	100	400
ResNet18	Adam	10^{-3}	1	150	350

the race and not the information such as the travel meter and level number, improves the performance significantly.

7 Challenges

The main problem that we faced was of compute, especially when it came to training the ResNet model, we ended up using Colab's GPU runtime to train the model. As the training set was not generated by a true expert of the game, we were only able to provide 2 levels to train the model. We did not play a perfect game, and the model learned from bad teachers.

8 Conclusion

We observed that imitation learning is a powerful method to solve challenging reinforcement learning tasks. A simple model with very limited training data was able to play well.

9 References

1. LeCun, Yann. "LeNet-5, convolutional neural networks." URL: `http://yann/`. lecun. com/exdb/lenet 20.5 (2015): 14.

2. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

3. Brockman, Greg, et al. "Openai gym." arXiv preprint arXiv:1606.01540 (2016)

4. Bellemare, Marc G., et al. "The arcade learning environment: An evaluation platform for general agents." Journal of Artificial Intelligence Research 47 (2013): 253-279.

5. Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In KDD arXiv).

6. Paszke, Adam, et al. “Pytorch: An imperative style, high-performance deep learning library.” Advances in neural information processing systems 32 (2019): 8026-8037.

7. Wikipedia

8. Wikipedia