CS221 Project Proposal
Deep Reinforcement Learning for Atari Games

Yinglan Ma, Hongyu Xiong, Chuan Tian

October 29, 2016

1 Task Definition

1.1 Scope

Recently deep reinforcement learning has shown promising results in playing Atari games. Mnih et al. proposed deep Q-network (DQN) approach that used a convolutional neural network as a function approximator to estimate the Q values [1]. DQN was tested on several Atari games, and outperformed competing methods and human players in some cases. However, DQN failed to beat human players in almost half of the games and there is still much room for improvement [2].

Our ultimate goal is to come up with a general deep reinforcement learning model that outperforms human baseline across different games. The input is raw pixels of the game environment, and the output is the total reward after a certain number of steps.

1.2 Data

We use OpenAI Gym as the platform for data gathering, environment setup, code development and result comparison [3]. The data are raw frames of Atari games, 210 by 160 RGB images. The data can be easily accessed by importing the gym library in Python. There might be some issues working with the full size frames, since training on large number of frames can be computational costly, but we will take the full size frames to begin with, and make adjustments on data as needed along the progress. Fig. 1 shows the environment of Atari game Tutankham, set up by OpenAI Gym

1.3 Evaluation

The key metric for performance evaluation is the average reward after a certain number of steps. Average reward of our algorithm can be obtained by running simulation using gym platform. In terms of comparing with other methods or human players, there are two sources with comparing information available. First is from literature where average reward provided with specific number of steps. We can run our algorithm under the same specifications as the ones in literature for reward comparison. Another source is OpenAI Gym with learning performance (100-episode average reward) given for various algorithms. Besides average reward, some other considerations for evaluation include complexity analysis, computational time, etc.

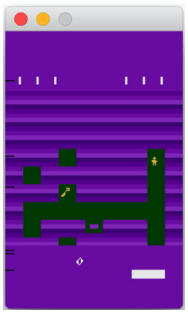

Figure 1: Environment of Tutankham-v0 set up by OpenAI Gym package.

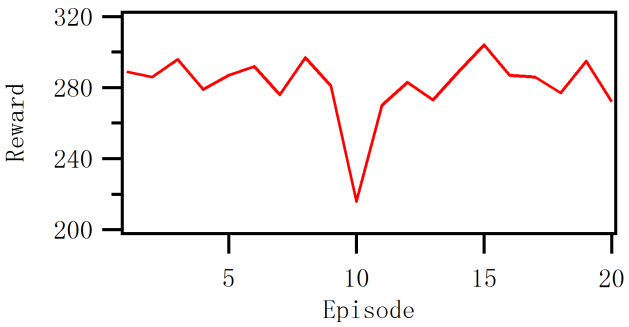

Figure 2: Reward obtained by an AI agent playing the game for multiple episodes, who is trained by Batch-A3C algorithm. The average reward is 289.

2 Approach

We started by gathering Batch-A3C algorithm baseline and human player oracle on Tutankham, one of the Atari games. Our inputs are raw frames, i.e. RGB images, of Tutankham game. And the output is the total reward gained after a certain number of steps.

2.1 Baseline

We ran a baseline algorithm based on Batch A3C [4]. The complete training process requires days (over 1 million iterations), but after a 24-hour training we can see the AI agent starts to know how to play the game; after 12000 iterations (two epochs), the agent got a maximum score of 3. We ended up using pre-trained baseline model to play the game [5], and we plotted the reward obtained for different episodes, which is shown in Fig. 2. A short piece of video is also generated to show the behavior of the AI agent trained by Batch A3C algorithm (Baseline on Tutankham).

2.2 Oracle

Ideally speaking, as long as the explorer in Tutankham is able to kill the creatures chasing him, and collect the keys throughout each level before time runs out, it is theoretically possible to reach a score of infinity. That provides an upper bound for the reward that can be obtained by playing the game. Thus we chose such imaginary grandmaster that can keep the explorer alive forever in Tutankham as our oracle. In reality, however, human players may feel tired or be influenced by the environment. That explains the finite reward of human players in literature.

3 Preliminary Analysis

As stated earlier, theoretically oracle is able to achieve a score of infinity, while our baseline gets an average score of 289, and in the video the explorer tends to "stuck" at certain places for a long time. There is a huge gap between baseline and oracle, in both score and performance. In this project, we aim to improve deep reinforcement learning for Atari games, by exploring different variants of DQN, e.g. input representation, choice of activation functions, number of hidden layers. Mathematically, to explore appropriate neural network approximations of $Q_{opt}(s, a; w)$ to minimize the objective function:

$$\min \sum_{s,a,r,s\prime} ((Q_{\text{opt}}(s, a; w) - (r + \gamma V_{\text{opt}}(s\prime))))^2. \tag{1}$$

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. *arXiv preprint* arXiv:1312.5602 (2013).

[2] Volodymyr Mnih, et al. *Nature* 518.7540 (2015): 529-533.

[3] OpenAI Gym. `https://gym.openai.com/`

[4] Volodymyr Mnih, et al. arxiv.1602.01783 (2016).

[5] https://github.com/ppwwyyxx