Deep Reinforcement Learning with Atari: Q-Learning vs. Policy Gradient

Zhe Cai, Silvia Ionescu, Monil Jhaveri, Andrew Levy
{s20525xx,mjhaveri,sgionescu,levya}@bu.edu

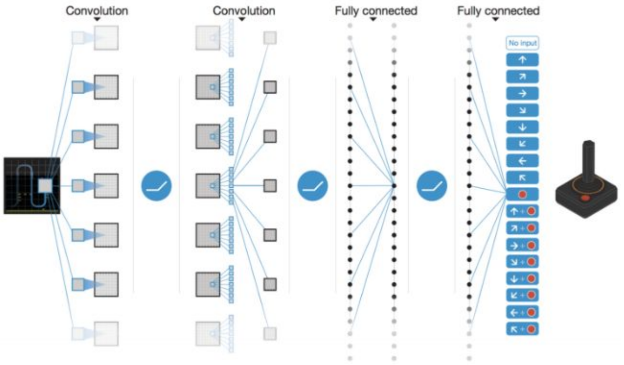

Figure 1. Deep Q-Learning Neural Network Mapping

1. Task

The main task of our project is to implement agents that can learn how to play Atari games using deep reinforcement learning techniques. More precisely, we plan to compare and contrast agents that use the two basic deep reinforcement learning methods, Q-Learning and policy gradient, within the Pong and Breakout Atari game environments. Time-permitting, we would like to implement the more advanced deep reinforcement learning algorithms that combine both Q-Learning and policy gradient methods in more complex game environments.

Two key challenges that our group will face in implementing these agents are the exploration-exploitation dilemma and the credit assignment problem. The goal in reinforcement learning is to find the optimal policy, or sequence of actions, that maximizes long-term reward. The exploration-exploitation dilemma represents the difficult tradeoff that occurs when trying to find the optimal policy. The agent can choose to more thoroughly explore the state-action space to try to find a policy that yields a better reward but at the cost of slower convergence. Or the agent can exploit what it has learned so far and explore a more limited region of the state-action space, thereby converging faster to a non-optimal policy. The credit assignment problem refers to the issue of determining which actions within a series of actions were most responsible for causing an outcome. This difficulty arises because rewards are often sparse and time-delayed.

2. Related Work

Mnih et al. provide one of the first successful applications of deep reinforcement learning to a complex environment. The authors show that by adapting the classical Q-Learning technique in a few important ways, agents can learn control policies at or exceeding human level in several different Atari games using only the pixels and the game scores as input [1]. Two of the key changes to the Q-Learning technique include convolutional input layers to better understand the input image and Experience Replay, a “memory” device that improves learning by enabling the neural network to learn from past experience that is not limited to the most recent actions. The author’s results also show one of the main flaws of Q-Learning -- long training times.

Several other papers provide theory and best practices for implementing policy gradient methods. Sutton et al. prove the convergence properties of the policy gradient method [2]. The authors show that if the parameters of the neural network that approximates the policy are updated in a way that is proportional to the product of the gradient of the policy and the estimated future reward of the policy, a local optimum in expected reward can be attained. Sutton and Barto, however, discuss one of the main drawbacks to generic policy gradient methods [3]. Convergence to a local optimum can be difficult if the future reward of an action is not properly estimated. Schulmann et al. discuss various ways to more effectively estimate the future reward, resulting in faster convergence to a local optimum in expected reward [4].

3. Approach

The goal of our project is to compare the effectiveness of the two main deep reinforcement learning algorithms: Q-Learning and policy gradient.

Q-Learning finds the optimal policy indirectly by determining the Q-value, or average future reward, for every state-action pair. A policy is then backed into by choosing the action with the highest Q-value for every state. The deep Q-Learning technique uses a neural network to approximate Q-values. The network maps the state (i.e., game image) to the Q-values for each action (see Figure 1).

On the other hand, the policy gradient technique optimizes a policy directly. Under this method, a neural network is used to map the state to the probability of each action. The policy gradient method then optimizes the network through essentially trial-and-error. The actions that produce positive rewards are made more likely to occur in the future while those that produce negative rewards are made less likely to occur.

Time-permitting, our plan is to also assess actor-critic methods, which are the state-of-art deep reinforcement learning techniques that combine Q-Learning and policy gradient algorithms. Similarly to policy gradient methods, actor-critic methods optimize a policy directly. However, the degree to which each action is made to occur more or less often in the future is dependent on the Q-value of the state-action pair. These methods thus use two neural networks. One networks maps from input image to probability of each action. Another maps from input image to the value of each action.

We plan to use convolutional layers with potentially some preprocessing steps for the input layers of our neural networks. As described above, the inputs to our networks are images. Convolutional layers should help the network achieve a stronger and more efficient understanding of the features in an input image. We will likely model the sequence and size of the convolutional layers after those used by DeepMind in [1].

4. Dataset and Metric

We will use the Atari Pong and Breakout environments provided by OpenAI Gym to train the agents. These game environments provide the state, action, and reward data that are necessary to implement the Q-Learning and policy gradient algorithms. The OpenAI Gym software also provides video that shows how an agent performs in an environment with the current code, making it easier to debug.

Implementing the deep reinforcement algorithms will require a small amount of data preprocessing. The inputs to the neural networks will likely be the differences between consecutive game images rather than the game image itself. Also, certain nonessential components of the game image, such as the scoreboard, will likely be removed.

The key metrics that we will use to assess the performance of the different algorithms are the average score per game and the number of training epochs required to achieve this average. We hope to achieve reward averages that meet or surpass the average human scores detailed in [1].

5. Approximate Timeline

Task	**Deadline**
Implement Q-Learning and P.G.	04/01/17
Implement advanced RL/Games	04/15/17
Prepare report and presentation	05/02/17

References

1) Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *Nature* 518.7540 (2015): 529-533.
2) Sutton, McAllester, et al. “Policy Gradient Methods for Reinforcement Learning with Function Approximation.” *Advances in Neural Information Processing Systems* (2000)
3) Sutton and Barto. *Reinforcement Learning.* MIT Press (1998)
4) Schulmann, Moritz, et al. “High-Dimensional Continuous Control using Generalized Advantage Estimation.” (2015)