CS 638 Final Project Proposal

Team members:

Zhanpeng Zeng (zzeng38), Qinyuan Sun (qsun28), Jingyi Zhao (jzhao62)

Task:

Using TensorFlow to develop deep reinforcement learning agents that can learn directly from high dimensional input (raw pixels) and can learn to play different games without modifying the network architecture. Then, we may also explore transfer learning methods and ensemble of multiple RL agents.

Software Used:

TensorFlow, OpenAI Gym, OpenAI Universe

Data and Testbed:

We will use OpenAI Universe and OpenAI Gym as environments for our RL agents. We may first try to use OpenAI Gym. This Gym contains RL tasks including classic control, Atari games, board games, and robots. Since it uses more meaningful state representations and provides a diverse set of environments ranging from easy to difficult, this Gym allows us to quickly build some simple working AI agents to get some experience on RL. Then, we will try a collection of flash games in OpenAI Universe to directly use high-dimensional raw input as agent state. This collection of flash games provides diverse tasks for agent to learn and allows agent to learn at human level through looking at screen and using keyboard and mouse.

CPU Cycle:

We will first test our agents on local GPU machine and profile the training time, then train our agents on AWS or Google Cloud Machine Learning Platform. We know that TensorFlow has support for distributed system, but we are not sure about OpenAI Universe. As a result, the backup plan is to train all of our AI agents on local GPU machine.

Project Plan:

1. Learn TensorFlow
2. Implement some simple working RL agents and test them on OpenAI Gym.
3. Explore the current approaches for deep RL.
4. Implement several current approaches using TensorFlow, train them using OpenAI Universe, and train them on distributed system.
5. Profile the convergence rate and performance of these agents.
6. Explore and implement some fine tuning methods to improve agents' performance. We will find some tuning methods from some papers.
7. (Optional if time permit) Try transfer learning methods to facilitate agent's learning.

8. (Optional if time permit) Try ensemble of multiple AI agents.

Experimental Methodology:

All RL agents in all experiments will be trained in online learning setting. We are not clear about the network configuration, such as number of layers or number of hidden units, for each agent yet. The best way to determine good network configuration is to experiment with each configuration for each agent in all environment, but due to computational expense, this may not be feasible. Thus, we may read some deep RL papers and try to determine some good network configurations for each agent.

Depending on the difficulty of implementation, we will implement at least 3 different RL agents. Two of these agents will be Double Q Learning and Model-based Learning, and the rest agents are not decided yet. And also we will select a subset of environments in OpenAI Universe and the size of subset is depending on the how long to takes to train. We will try to choose as many environments as possible to make the experimental results more reliable.

Then, we will train each RL agent on all environments in this subset and record the convergence rate and training time. Depending on how much time it takes to train the agents, we may run each experiment multiple times and take average performance. **Experimental Evaluation:** We will compare the convergence rate among all agents, and compare the performance of these agents by letting the agents interact with each environment and comparing the total reward them get.

(Optional) After evaluation of these agents, we will try some transfer learning methods. Although we have general understanding of transfer learning, we do not know how transfer learning is applied in RL. One possible method is to use agents trained for other tasks to give advice to currently training agents. We will read some transfer learning papers to explore some possible methods. **Experimental Evaluation:** We will compare convergence rate and performance of transfer-learnt agents to the agents we previously trained.

(Optional) We may also try ensemble of multiple RL agents, but we are not clear how to do ensemble besides just gathering the agents we trained previously. As a result, we will not discuss this experiment in the proposal.

Subset of Current Approaches That We May Explore:

1. Policy-based Learning: Use a deep neural network to represent policy and directly learn the optimal policy.
2. Model-based Learning: Use a deep neural network to represent model and learn the model by learning the transition of state and use this information to indirectly choose optimal action.
3. Value-based Learning: Use a deep neural network to represent Q value, approximate the actual Q value, and use the approximated value to choose optimal action. Current approaches: Deep Q Learning, Double Q Learning, and Deep Recurrent Q Learning