Deep Q-Network

Reinforcement Learning with TensorFlow&OpenAI Gym

Youngho Byeon - July 20, 2017

Domain Background

Deep Q-Network(DQN) is an algorithm developed by Google DeepMind, which uses deep learning network.[1] The algorithm was published in Nature in February 2015.[2] It's reinforcement learning network that can play Atary games like a human being. The amazing part of this algorithm is simple and powerful. If you just tell a agent that "high score is your goal", the agent will gradually understand game rules and find out a best strategy.

It is one of the most interesting reinforce learning algorithms for me, because in my country, South Korea, there was a Go competition with AlphaGo and Sedol Lee in last year. As you know, the result was a victory of AlphaGo. At that time, although I was a iOS programmer, I watched the game interesting. And, I though if I were going to study Machine learning later, I would dive into this topic deeply. So, I choose this algorithm for my last capstone project of the Machine Learning Nanodegree.

Problem Statement

Reinforcement network learning is not state of the art algorithm. But, it wasn't common before DQN because it has some problems: 1. Correlations between samples. 2. Non-stationary targets.[3] So, I will start with basic Q-learning. And I will find out why these are the problems and how DQN solves the problems.

Datasets and Inputs

[1] https://sites.google.com/a/deepmind.com/dqn/

[2] https://www.nature.com/nature/journal/v518/n7540/full/nature14236.html

[3] https://stackoverflow.com/questions/10722064/training-a-neural-network-with-reinforcement-learning

I will use OpenAI Gym. It is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Go.[4] Especially, I choose 'FrozenLake' and 'CartPole' games. Because they has very simple rules and easy to understand. So, I think it is appropriate for beginner.

FrozenLake is that a agent controls the movement of a character in a grid world. Some tiles of the grid are walkable, and others lead to the agent falling into the water. Additionally, the movement direction of the agent is uncertain and only partially depends on the chosen direction. The agent is rewarded for finding a walkable path to a goal tile.[5] So, in this game, inputs are movement directions of the agent (Up, Down, Left, Right). And, there are 4 states in the grid: Starting point, Frozen surface, Hole, Goal.

CartPole is that a pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The system is controlled by applying a force of +1 or -1 to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. A reward of +1 is provided for every timestep that the pole remains upright.[6] For this game, we have four inputs, one for each value in the state, and two outputs.

Solution Statement

In FrozenLake game, the episode ends when you reach the goal or fall in a hole. The agent receive a reward of 1 if it reach the goal, and zero otherwise. So, I will find a solution that receive the reward of 1. And then, I will make more complex environments. With more complex models, I will demonstrate why the previous algorithm had some problems.

And in CartPole game, the episode ends when the pole is more than 15 degrees from vertical, or the cart moves more than 2.4 units from the center. According to CartPole document, it defines "solving" as getting average reward of 195.0 over 100 consecutive trials.

[4] https://gym.openai.com/

[5] https://gym.openai.com/envs/FrozenLake-v0

[6] https://gym.openai.com/envs/CartPole-v0

CartPole is more complex than FrozenLake. So, I will build a deep learning reinforcement network by Tensorflow[7] and explain DQN model.

Benchmark Model

It is reinforcement model, so it doesn't have target label and it can be multiple correct answers. As I mentioned before, in FrozenLake, the agent will receive the reward of 1 when it reach the goal. So, I should check the reward every episode and I will make a plot how it would be optimized over time. And, I should need to make sure the solution is an optimized route. Because, although the route received the reward of 1, it couldn't be a efficient way. In this case, there are multiple answers. In conclusion, I should check the reward and optimization for benchmark model.

And in CartPole, we can check the time for benchmark. Because in this game, time is reward. And the pole is more than 15 degrees from vertical, the episode will be end. So, I can use the pole's degree as sub benchmark model.

Evaluation Metrics

FrozenLake model will be evaluated by the reward. And I will use Q-function, so I can evaluate by Q-table. For each state that has been recorded from the actions, I will check values for the each states. After some learning, the model should have the reward of 1.

And CartPole model will be evaluated by time. In this case, I will use Deep Q-leaning. After learning, the model should getting average reward of 195.0.

Project Design

I will build a Neural network by Tensorflow and OpenAI Gym.

1. Q-leaning by Q-table : Make simple FrozenLake agent based Q-tables. There are some problems in this model. So, I will improve Q-learning model in the next step.

[7] https://www.tensorflow.org/

2. Q-leaning by improved Q-table : Make FrozenLake agent based Q-tables. This model is very similar to first model. But, I could improve by decaying greedy. So, I will compare with each model, and rewrite my Q-learning algorithm.

3. Q-learning by Neural network : Make CartPole agent based Neural network by Tensorflow. It is a combination of Q-learning and Deep-learning. And, It has some problems too. So, I will improve this model in the next step.

4. Q-learning by DQN : This model is my final goal. I will build DQN model by Tensorflow. With this model, I will show how Google Deep Mind solved the problems of the previous model, and how the final result improved.