WILDML

Artificial Intelligence, Deep Learning, and NLP

OCTOBER 2, 2016 BY DENNY BRITZ

Learning Reinforcement Learning (with Code, Exercises and Solutions)

Skip all the talk and go directly to the Github Repo with code and exercises.

Why Study Reinforcement Learning

Reinforcement Learning is one of the fields I'm most excited about. Over the past few years amazing results like learning to play Atari Games from raw pixels and Mastering the Game of Go have gotten a lot of attention, but RL is also widely used in Robotics, Image Processing and Natural Language Processing.

Combining Reinforcement Learning and Deep Learning techniques works extremely well. Both fields heavily influence each other. On the Reinforcement Learning side Deep Neural Networks are used as function approximators to learn good representations, e.g. to process Atari game images or to understand the board state of Go. In the other direction, RL techniques are making their way into supervised problems usually tackled by Deep Learning. For example, RL techniques are used to implement attention mechanisms in image processing, or to optimize long-term rewards in conversational interfaces and neural translation systems. Finally, as Reinforcement Learning is concerned with making optimal decisions it has some extremely interesting parallels to human Psychology and Neuroscience (and many other fields).

With lots of open problems and opportunities for fundamental research I think we'll be seeing multiple Reinforcement Learning breakthroughs in the coming years. And what could be more fun than teaching machines to play Starcraft and Doom?

How to Study Reinforcement Learning

There are many excellent Reinforcement Learning resources out there. Two I recommend the most are:

- David Silver's Reinforcement Learning Course
- Richard Sutton's & Andrew Barto's Reinforcement Learning: An Introduction (2nd Edition) book.

The latter is still work in progress but it's ~80% complete. The course is based on the book so the two work quite well together. In fact, these two cover almost everything you need to know to understand most of the recent research papers. The prerequisites are basic Math and some knowledge of Machine Learning.

That covers the theory. But what about practical resources? What about actually implementing the algorithms that are covered in the book/course? That's where this post and the Github repository comes in. I've tried to implement most of the standard Reinforcement Algorithms using Python, OpenAI Gym and Tensorflow. I separated them into chapters (with brief summaries) and exercises and solutions so that you can use them to supplement the theoretical material above. All of this is in the Github repository.

Some of the more time-intensive algorithms are still work in progress, so feel free to contribute. I'll update this post as I implement them.

Table of Contents

List of Implemented Algorithms

- Dynamic Programming Policy Evaluation
- Dynamic Programming Policy Iteration
- Dynamic Programming Value Iteration
- Monte Carlo Prediction
- Monte Carlo Control with Epsilon-Greedy Policies
- Monte Carlo Off-Policy Control with Importance Sampling
- SARSA (On Policy TD Learning)
- Q-Learning (Off Policy TD Learning)
- Q-Learning with Linear Function Approximation
- Deep Q-Learning for Atari Games
- Double Deep-Q Learning for Atari Games
- Deep Q-Learning with Prioritized Experience Replay (WIP)
- Policy Gradient: REINFORCE with Baseline
- Policy Gradient: Actor Critic with Baseline
- Policy Gradient: Actor Critic with Baseline for Continuous Action Spaces
- Deterministic Policy Gradients for Continuous Action Spaces (WIP)
- Deep Deterministic Policy Gradients (DDPG) (WIP)
- Asynchronous Advantage Actor Critic (A3C) (WIP)

DEEP LEARNING, REINFORCEMENT LEARNING