Reinforcement Learning
Today, Direct Policy Optimization

- Policy may be a simpler function to learn
- More naturally deal with stochastic policies
Plan for Today

• PPO
• Alternate methods
• Course Projects
• Model building overview
• ME-TRPO
Policies may be simpler

Do Fielders Know Where to Go to Catch the Ball or Only How to Get There?

Peter McLeod
Oxford University

Zoltan Dienes
Sussex University

Skilled fielders were filmed as they ran backward or forward to catch balls projected toward them from a bowling machine 45 m away. They ran at a speed that kept the acceleration of the tangent of the angle of elevation of gaze to the ball at 0. This algorithm does not tell fielders where or when the ball will land, but it ensures that they run through the place where the ball drops to catch height at the precise moment that the ball arrives there. The algorithm leads to interception of the ball irrespective of the effect of wind resistance on the trajectory of the ball.

Modulate running speed to maintain angle between ball and ground.

If ball rises in the field of view, slow down.
If ball drops, speed up.
Thus, position of the ball in the field of view is maintained.

Using this heuristic, human catcher arrives at landing point exactly when the ball lands.
Stochastic policies

Rock, paper, scissors

Image Credit: David Silver
Directly Optimize Policies?

\[ \pi_\theta(s, a) \text{ policy (parameterized by } \theta) \]

\[ J(\theta) = \text{Average return when acting as per } \pi_\theta(s, a) \]

\[ \theta_* = \arg\max_\theta J(\theta) \]
The policy gradient has many equivalent forms

\[ \nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(s, a) \ n_t] \quad \text{REINFORCE} \]
\[ = \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(s, a) \ Q^w(s, a)] \quad \text{Q Actor-Critic} \]
\[ = \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(s, a) \ A^w(s, a)] \quad \text{Advantage Actor-Critic} \]
\[ = \mathbb{E}_{\pi_\theta} [\nabla_\theta \log \pi_\theta(s, a) \ \delta] \quad \text{TD Actor-Critic} \]
Many successes in simulation

[Image of robot hands with colored blocks]

Summary

Model-Free RL

Policy Optimization
- Policy Gradient
  - A2C / A3C
  - PPO
  - TRPO
- DDPG
- TD3
- SAC

Q-Learning
- DQN
- C51
- QR-DQN
- HER

Source: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#id20
Policy gradient really a gradient?

Evolution Strategies as a Scalable Alternative to Reinforcement Learning

Tim Salimans  Jonathan Ho  Xi Chen  Szymon Sidor  Ilya Sutskever

Hypercheetah

Hopper

Walker

See also: http://www.argmin.net/2018/02/20/reinforce/
Solving MDPs

**Policy:** \( a_t \sim \pi(o_t) \)

**Most General Case**

**More Specific Case**

**Fully Observed System**

\( o_t = s_t \)

**Known Transition Function**

\( s_{t+1} \sim T(s_t, a_t) \)

**Known Reward Function**

\( R(s_{t+1}, s_t, a_t) \)
So, are we done?

- Exploration is challenging
- Credit assignment problem

Poor sample complexity
Sample Complexity

Effect of Scale in Simulation

Solving a RL Problem

Better reward signals | Sim2Real

Better optimization | Convert into a supervised training problem

Solve a related but supervision rich problem | Build models and plan with them

Model-free RL with sparse rewards | Known reward, known model. Model-based RL