USING REINFORCEMENT LEARNING TO PLAY ANGRY BIRDS

colloquium

Peter Hirsch

Dresden, 2017/9/26
Table of Contents

1  Angry Birds
2  Neural Networks
3  Reinforcement Learning
4  Deep Deterministic Policy Gradient
5  RL Agent: Dr. L. Bird
6  Results
output = \sum \text{weights} \cdot \text{inputs}
Deep Neural Networks

A mostly complete chart of Neural Networks

©2016 Fjoed van Veen - asinovinstitute.org
RL in a Nutshell

Environment

Agent

state, reward

action

TU Dresden, 2017/9/26

Reinforcement Learning
RL in a Nutshell

Angry Birds

Agent

image, score

shoot bird
RL: Policy-based Methods

deterministic policy

stochastic policy

(with state approximation, gray states not distinguishable)
RL: Value-based Methods

Value Estimator:
- table
- neural network
- ...

Action 1: $<\text{value1}>$
Action 2: $<\text{value2}>$
Action 3: $<\text{value3}>$
...

a greedy or $\epsilon$-greedy policy is used to act
(a.k.a. go to neighboring state with highest (Q-)value)
Combination of policy-based and value-based
Deep Deterministic Policy Gradient

**Actor**
- State $s$ → Action $a$
- $Q(s, a)$

**Critic**
- $Q(s, a)$ → State $s$

**Evaluation**
- Blue arrows

**Optimization**
- Red arrows

TU Dresden, 2017/9/26
DDPG: Acting

Actor

Experience Buffer

Evaluation
Optimization

state s → action a

store

execute

store
Experience Buffer

- state s
- action a
- next state s'
- reward r
Experience Buffer

sample $s, a, s', r$ → $s'$

Actor

Q($s', a'$)

Critic

TU Dresden, 2017/9/26
DDPG: Learning - Part 2

Experience Buffer

sample → s, a, s', r

Q(s, a) ← Actor

Critic

← s, a

Evaluation
Optimization

TU Dresden, 2017/9/26
Optimization: Backpropagation using chain rule across the two networks
Dr. L. Bird

state $s$

$Q(s, a)$

action $a$

Evaluating

Optimizing

Actor

Critic

TU Dresden, 2017/9/26
Dr. L. Bird - Agent

Evaluation
Optimization

state s

Q(s, a)

action a

state s

Actor

Critic

TU Dresden, 2017/9/26 Dr. L. Bird slide 20
Results

Score per Game [in 10000]

Games

Score per Game [in 10000]

Games

Score per Game [in 10000]

Games

Score per Game [in 10000]

Games

TU Dresden, 2017/9/26
Tested Versions

• DDPG loss function:
  – Q-Learning-based (off-policy)
  – Sarsa-based (on-policy)
  – TD-based (estimated value)
  – Monte-Carlo-based (cumulative return)

• Stochastic Policy Gradient:
  – policy-based
  – uses statistics of probability distribution
  – output (sampled for action):
    • mean
    • variance

• A3C (asynchronous advantage actor-critic)
  – actor-critic
  – stochastic policy
  – parallel (asynchronous) execution of multiple agents
  – advantage instead of Q-value (relative value of actions)

→ no success so far
Sources neural network schematics:


Source policy-based method example:
http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
Discussion