SPEEDING UP DEEP REINFORCEMENT LEARNING VIA TRANSFER AND MULTITASK LEARNING

Speaker: Yunshu Du
Host: Gail Murphy
Yunshu Du

Yunshu Du is a third year PhD student at the School of Electrical Engineering and Computer Science at Washington State University, under the supervision of Dr. Matthew E. Taylor. From 2010 to 2012, she majored in software engineering at Wuhan University in China. Yunshu transferred to Eastern Michigan University to study computer science for her junior year. After two years of study, she obtained the Bachelor of Science degree of computer science, with a minor of Geographical Information System in 2014.

Gail Murphy

Dr. Gail Murphy is a Professor in the Department of Computer Science and Associate Dean (Research & Graduate Studies) in the Faculty of Science at the University of British Columbia. She is also a co-founder and Chief Scientist at Tasktop Technologies Incorporated. Her research interests are in software engineering with a particular interest in improving the productivity of knowledge workers, including software developers. Dr. Murphy’s group develops tools to aid with the evolution of large software systems and performs empirical studies to better understand how developers work and how software is developed.
SPEEDING UP DEEP REINFORCEMENT LEARNING VIA TRANSFER AND MULTITASK LEARNING

Yunshu Du
Intelligent Robot Learning Laboratory
Washington State University

CRA-W Undergraduate Town Hall
July 27, 2017
ABOUT ME

• Born and raised in Wuhan, China
  – Capital city of Hubei province, “the Chicago of China”
  – Must-see: the Yellow Crane Tower, the Yangtze river
• Came to the US in 2012
  – A short visit in Texas in 2009
  – Eastern Michigan University, BS in Computer Science 2014
• Joined Washington State University 2014
  – PhD in Computer Science. Advisor: Dr. Matt Taylor
  – Current Research: reinforcement learning, applied data science
OUTLINE

• AI and Machine Learning
• Reinforcement Learning
• Deep Reinforcement Learning
• Transfer and Multi-task Learning
Artificial Intelligence
Artificial Intelligence

AGENT

What is the world like now

Condition-action (if-then) rules

Action to be done

Actuators

Sensors

percepts

ENVIRONMENT

actions
Why is learning important?

- Unanticipated situations
- Faster for human programmer
- Better than human programmer/user
Machine learning is one of many approaches to achieve AI.

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
Reinforcement Learning (RL)

- Inspired by behaviorist psychology
- An agent explores an environment and decide what action to take
- Learn from reward signal, but it is often delayed/limited
- State changes upon the action took
- Things happen in a sequential way
  - The Markov Decision Process: \{s, a, r, s’\}
  - The goal is to find an optimal policy so that the agent maximize the reward accumulated
Example: teaching a dog to lie down
Action: up, down, or stay
Example: teaching a dog to lie down

Action: up, down, or stay
Example: teaching a dog to lie down

Policy: state-action mapping

standing → down → sitting → down → lying

GOOD DOG + bones
<table>
<thead>
<tr>
<th>Game</th>
<th>Board size</th>
<th>State space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go</td>
<td>19 × 19</td>
<td>$10^{172}$</td>
</tr>
</tbody>
</table>

What if states are huge?
Function Approximator

What you see → Process in brain → Output actions
Deep Learning

- Inspired from neuronal responses in the brain, a tool to implement machine learning algorithms
- Use deep neural network as *function approximator* to represent *features* in an environment

An agent processes what it “sees” with a neural network
Deep Reinforcement Learning

(Any) RL algorithms

DeepRL in Google DeepMind:
- Deep Q-network: general Atari game playing agent
- Gorila: distributed deep RL system
- Asynchronous deep RL: Atari + continuous control
- AlphaGo: defeated world’s No. 1 professional Go player

DeepMind Blog:
https://deepmind.com/blog/deep-reinforcement-learning/
Deep Q-network (DQN)

- An artificial agent for general Atari game playing
- Learn to master 49 different Atari games directly from game screens
- Excel human expert in 29 games
- Q-learning + convolutional neural network
Deep Q-network (DQN)

Network Architecture

- Input: Atari image 84x84
- Convolutional
- Extract Features
- Q values for each action
- Fully Connected
- Output actions
- Reward signal: score + life

Input Atari image
Deep Q-network (DQN)

Techniques to Help Stabilize Learning

- Reinforcement learning is known to be unstable or even to diverge when use neural network as function approximator

- Main solution: save experiences first, then learn from them later
Deep Q-network (DQN)

Techniques to Help Stabilize Learning

- Reinforcement learning is known to be unstable or even to diverge when use neural network as function approximator.

- Main solution: save experiences first, then learn from them later.

Experience Replay Memory

Randomly pick a set of experience

Input to network
Deep Q-network (DQN)

Techniques to Help Stabilize Learning

• Reinforcement learning is known to be unstable or even to diverge when use neural network as function approximator

• Main solution: save experiences first, then learn from them later
My Research

Problem
• DeepRL is slow in learning: 10 days to learn one game
  – A RL agent needs time to explore the environment
  – A deep neural network has millions of parameters
  – This is problematic in real-world, e.g., train a program to drive a car

Solution
• Transfer Learning
• Multi-task Learning
My Research

Transfer Learning (TL) in DQN

- Task Selection
  - Source task: task(s) the agent has already learned
  - Target task: task(s) to be learned
  - Usually select by a human based on task similarities, similar tasks are more likely to transfer well

A trick to increase task similarity

Breakout

Pong
My Research

Transfer Learning (TL) in DQN

- Weight Transfer
  - Copy weights
  - Fine-tune
  - Transfer in CNN layers only
My Research

Transfer Learning (TL) in DQN

- Weight Transfer
  - Copy weights
  - Fine-tune
  - Transfer in CNN layers only
My Research

Transfer Learning (TL) in DQN

• How to evaluate
  – *Jumpstart*: the agent's initial performance on the target task was improved by transferring source task knowledge
  – *Final performance*: the agent's final performance on the target task was improved via transfer
  – *Total reward*: the accumulated reward (the area under the curve) on the target task was improved compared to no-transfer learning (within the same learning time period),
My Research

Transfer Learning (TL) in DQN

Pong Transfer to Breakout

- Baseline
- PtB, 1 Layer
- PtB, 2 Layers
- PtB, 3 Layers

1.25 million steps
My Research

Transfer Learning (TL) in DQN

Pong Transfer to Breakout

Final Performance

Jumpstart
My Research

Transfer Learning (TL) in DQN

Pong Transfer to Breakout

- Baseline
- PtB, 1 Layer
- PtB, 2 Layers
- PtB, 3 Layers

Total Reward

Final Performance

Jumpstart
My Research

Transfer Learning (TL) in DQN
My Research

Transfer Learning (TL) in DQN

Pong Transfer to Breakout

- Baseline
- PtB, 1 Layer
- PtB, 2 Layers
- PtB, 3 Layers
My Research

Transfer Learning (TL) in DQN

Pong Transfer

Breakout Transfer to Pong

- Baseline
- PtB, 1 Layer
- PtB, 2 Layers
- PtB, 3 Layers

Reward

Epochs

Epoch
My Research

Multi-task Learning (MTL) in DQN

- Task Selection: related tasks are more likely to help each other
- Modify the DQN’s architecture to enable multiple game inputs
My Research

Multi-task Learning (MTL) in DQN

• Design Choices
  – How often should games be switched
    • Every 1 step? Every 10,000 steps? Until one agent lose?
  – Should experience replay memory (EM) be shared

At what point to split the original DQN network
My Research

Multi-task Learning (MTL) in DQN

- How to evaluate
  - Final performance
  - Total reward
My Research

Multi-task Learning (MTL) in DQN

- How often should games be switched
- Should experience replay memory be shared

Switch every step, share EM vs. not share EM
My Research

Multi-task Learning (MTL) in DQN

- How often should games be switched: more frequent (switch1) seems better
- Should experience replay memory be shared: no sharing (sep) seems better

Breakout

Pong

Switch every 1,250 step, share EM vs. not share EM
My Research

Multi-task Learning (MTL) in DQN

- At what point to split the original DQN network
  - at higher level (more sharing) seems better for Breakout, but worse for Pong

Split the network at different layers

[Graph showing comparison between Breakout and Pong for different network splitting strategies]
My Research

Take Away

• TL and MTL shows the potential of speeding up learning in DQN

• However, empirical results were not enough to draw a solid conclusion

• Future study
  – Test in more domains
    • Atari games: does not help all games and uncertain why
    • Continuous control problems
  – Knowledge selection for each layer in DQN
    • How to Interpret neural networks
  – Robust source/target task selection mechanism
    • How to measure the similarity between games
    • Can we automate the selection process
HOW TO CHOOSE A RESEARCH DIRECTION AS AN UNDERGRADUATE

Yunshu Du
Intelligent Robot Learning Laboratory
Washington State University

CRA-W Undergraduate Town Hall
July 27, 2017
Outline

• I assume you have already know about how to participate in research
  – If not, there are previous VUTH sessions that provide great resources:
  – 12/1/16 Katherine Sittig-Boyd: Getting Involved in Undergraduate Research
  – 4/18/17 Rebecca Wright: Getting Involved in CS Extra-curricular Activities

• This is a general guide on how to pick a research project

• I will talk about how did I end up in my current direction

• Followed by a mini discussion panel with current undergraduate researchers in our department, we will cover:
  – How to find a project
  – What to expect
  – Other things to be considered
How did I pick my direction

Software Engineering

Computer Science
Geographic Information System

Bioinformatics
Reinforcement Learning
Deep Learning
Data Science
Voice From Current UG researchers
How to find a project

• Find what field are interested in
  – Machine Learning, Robotics, Software Security, Teaching Programming, etc.
  – Even if you don’t have a specific interest, you can still try something new
  – It is totally okay to change directions

• Find a professor/lab that does what you are interested in
  – Your current CS professor can point you in the right direction
  – Browse faculty page, email professors
  – Visit lab/office

• Discuss with professor
  – Brainstorm possible projects, it can be you own idea or pick from a list of what the professor is doing
  – If you want to work on an ongoing project, learn the current status of the project
What to expect

• You may have to do some self study on prerequisites
  – Understand what you need to know and build up from small pieces
  – Self study is research too (e.g., take an online class)!
  – You don’t need to be great to start, but you have to start to be great

• Research is high variance
  – Things might not work
  – Things might work extremely well

• Self motivation
  – Professors are not responsible for providing you a to-do list
  – Be proactive
  – Remember it is for your own development

• Have fun 😊
  – Not under high pressure
  – Meet new friends
Other considerations

• The people in the group
  – Not only knowing what research the professor/group members are doing, but also their personality

• Location
  – Will you be willing to spend months in that city
  – Do you like the lab environment

• If you will get paid
  – You could be more productive if there is a paycheck

• Time management
  – How long can you work per week and how is the hours allocated
  – Is the project feasible within the timeline you have