July 5, 2018

Advanced Micro Devices
Nicholas Malaya, Ph.D., nicholas.malaya@amd.com
Abhinav Vishnu, Ph.D., abhinav.vishnu@amd.com

Dear Dr. Nicholas Malaya and Dr. Abhinav Vishnu:

Enclosed are two copies of the Statement of Work (SOW) and two copies of this cover letter. The SOW outlines our team's current understanding of the problem and addresses our planned approach to a solution.

Please show your approval of the SOW by signing both copies of the cover letter in the space provided on this page, or by indicating your proposed changes, and returning one copy of each (SOW and signed cover letter) to me by Monday, July 16. Otherwise, after that date, we will assume Advanced Micro Devices' tacit approval.

Sincerely,

Jiajing Guan
RIPS Project Manager

Institute for Pure and Applied Mathematics (IPAM)
Attn RIPS: Jiajing Guan

Enclosure: Advanced Micro Devices RIPS Work Statement

Cc: Susana Serna, RIPS Program Director
Stacey Beggs, IPAM Assistant Director

Accepted this _____ day of July 2018

By: ____________________________

Research in Industrial Projects for Students

Sponsor

Advanced Micro Devices

Statement of Work

Exploration of Reinforcement Learning in Computer Games

Student Members

Jiajing Guan (Project Manager), *George Mason University*
jiajingguan@gmail.com
Patrik Gerber, *University of Oxford*
Elvis Nunez, *Brown University*
Kaman Phamdo, *University of Maryland*

Academic Mentor

Tonmoy Monsoor, mtonmoy@g.ucla.edu

Sponsoring Mentors

Nicholas Malaya, nicholas.malaya@amd.com
Abhinav Vishnu, abhinav.vishnu@amd.com
Alan Lee, alan.lee@amd.com

Date: July 5, 2018

Introduction

Advanced Micro Devices, Inc. (AMD) is an American multinational semiconductor company based in Santa Clara, California, that develops computer processors and related technologies for business and consumer markets. AMD's main products include microprocessors, motherboard chipsets, embedded processors and graphics processors for servers, workstations and personal computers, and embedded systems applications.

A major automotive company demoed an RC car at Conference NIPS 2017 that dynamically explored and interacted with its environment. As the car explored, it learned more about its environment and how to best navigate it. This is distinct from typical approaches in Artificial Intelligence, which operate in separate modes of training and inference. Often, an algorithm must first learn by viewing examples of a task being performed (training) before it is capable of making predictions for this same task (inference). The bifurcation between learning and action is distinct from naturally occurring intelligences, which acquire knowledge continuously through experience and trial-and-error. Mimicking the feedback loop between action and results is the motivation for the field of reinforcement learning, which provides a means for an agent to continually learn while dynamically interacting with an environment. Ultimately, the promise of reinforcement learning is the creation of general-purpose artificial intelligences that can interact with and learn from the world around them. However, reinforcement learning has numerous issues including intransigence towards changes in environment and the need for large training datasets. AMD's interests in addressing these technical difficulties resulted in this project.

Problem Description

Technical Background

Typically, supervised machine learning approaches operate in distinct modes of training and inference. During training, an algorithm is shown labeled data and learns a representation of the underlying distribution of the data. After training, the model can be deployed to provide predictions in a phase known as inference. When such a model is built from layers of computational units called neurons that interact with each other, we call it deep learning.

The field of deep reinforcement learning blends techniques from the areas of deep and reinforcement learning to provide a means for an agent to continually learn while dynamically interacting with an environment. A periodic "reward" signal provides the agent feedback to guide it towards behavior that is considered beneficial. Many complex tasks provide such feedback rarely and with large latency, resulting in a difficult learning task. Furthermore, classical reinforcement learning had serious limitations when applied to high-dimensional data. There have been recent developments in incorporating deep neural networks into reinforcement learning frameworks. Deep Q-Network (DQN) is one of the common deep reinforcement learning algorithms. Studies have shown DQN to be successful in some formerly intractable problems.

However, DQN comes with many issues, one of which is its sensitivity to the environment. In other words, a change in the environment could result in a catastrophic failure of the algorithm. We attempt to tackle this problem by exploring DQN's robustness and solutions to desensitize the algorithm.

Problem Statement

This project will focus on improving the flexibility of reinforcement learning algorithms, allowing them to adapt to perturbations in the environment. The project will allow the RIPS team to explore common frameworks and algorithms in reinforcement learning and investigate techniques to reduce the power consumption and latency of the deployed model without compromising the accuracy of the system. A possible avenue of exploration includes enriching the feedback received by the agent to guide it towards more effective learning. Moreover, analysis of model efficiency will revolve around characterizing the influence of hyperparameters, precision of numerical operations, and sensitivity to hardware devices.

Objectives

Main Objective

The focus will be to characterize common reinforcement learning approaches, and implement a state-of-the-art learning algorithm using techniques described in recent literature. The RIPS team will then test the algorithm on simple computer games, such as Snakes or the classic Atari game NeonRacer.

Additional Objectives

If time permits, the RIPS team will attempt the following objectives:

- Reduce the power consumption and latency of the learning algorithm without compromising the accuracy of the system. A possible approach to this includes analyzing the impact of parameters such as the frame rate, learning rate and precision of numerical operations on the algorithm's performance.
- Improve existing approaches by augmenting them with curriculum and imitation learning (providing simple examples for the agent to learn). Observe the robustness of this update and its flexibility.

Approach

Reinforcement learning agents learn by exploring their environment freely and receiving rewards for ideal behavior. A natural venue that provides such a sandbox is computer games. While employing deep learning methods, the RIPS team will implement reinforcement learning algorithms to train agents to successfully solve computer games. Due to low cost in training, algorithms will be trained on the games provided by OpenAI Gym. Given a trained model, the team will explore the effects of changing parameters in the environment, and if time permits, provide a theoretical justification for these results.

Deliverables

- From the RIPS team to Sponsors
  - Weekly conference call meetings
  - Midterm oral presentation
  - Midterm progress report
  - Oral presentation at AMD site
  - Projects Day presentation
  - Final report of findings
  - Software
    - Any code used to study the problem as per request
- From Sponsors to the RIPS team

- Set up AMD site visit (to be arranged)
- Weekly conference call meetings
- Project and platform specifications

Timetable

- Week 1 - 3
  - Read reinforcement learning and deep learning texts
  - Work on reproducing results from the paper *Playing Atari with Deep Reinforcement Learning* by Mnih., et al. by implementing Q-learning on a simple computer game
- Week 4 - 6
  - Analyze the impact of changes in various parameters on the performance and results of Q-learning model
  - Prepare and present midterm written report and oral presentation
- Week 7 - 9
  - Site visit
  - Perform any further experiments based on previous results
  - Complete final report and Projects Day presentation

Reference

[1] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, *A brief survey of deep reinforcement learning*, arXiv preprint arXiv:1708.05866, (2017).

[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, *Openai gym*, arXiv preprint arXiv:1606.01540, (2016).

[3] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1, MIT press Cambridge, 2016.

[4] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, arXiv preprint arXiv:1502.03167, (2015).

[5] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, *Teacher-student curriculum learning*, arXiv preprint arXiv:1707.00183, (2017).

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, *Playing atari with deep reinforcement learning*, arXiv preprint arXiv:1312.5602, (2013).

[7] Y. Tian, Q. Gong, W. Shang, Y. Wu, and C. L. Zitnick, *Elf: An extensive, lightweight and flexible research platform for real-time strategy games*, in Advances in Neural Information Processing Systems, 2017, pp. 2656–2666.

IPAM Software Disclaimer for RIPS Sponsors

June 14, 2017

We want our RIPS sponsors to be aware of the nature of software developed by RIPS project teams. IPAM does not regard RIPS software as anything more than a prototype developed as a proof-of-concept only, and it is never developed for commercial use nor is it warranted by IPAM in any way. Here are some points to remember:

1. Software developed by a RIPS project team that appears to have been created wholly by a project team, may in fact contain proprietary codes borrowed from other sources; the sponsor must assume all risk for using such software.

2. IPAM makes every effort to discourage misuse of proprietary software by RIPS project participants; IPAM cannot be held responsible for such misuse.

3. As participants in an academic program, RIPS students will at times be permitted to use software that cannot be used by sponsors without a license.

4. Any restriction required by the sponsor on the use of special software, or platform needed to run the software, should be declared by the sponsor at the time of negotiating the project Work Statement. Otherwise the project team is free to choose software solutions as they see fit.