Neuro-Symbolic Reinforcement Learning: A Reinforcement Learning Platform & Neuro-Symbolic Agent

Hunter W. Ellis

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

Thinh T. Doan, PhD, Chair

Michael S. Hsiao, PhD, Co-chair

Ryan K. Williams, PhD

March 23, 2016

Blacksburg, Virginia

Keywords: Reinforcment Learning, Neuro-symbolic, Neuro-symbolic Concept Learner, Multi-task Reinforcement Learning, Robotics

Neuro-Symbolic Reinforcement Learning: A Reinforcement Learning Platform & Neuro-Symbolic Agent

Hunter W. Ellis

(ABSTRACT)

Neuro-Symbolic Reinforcement Learning: A Reinforcement Learning Platform & Neuro-Symbolic Agent

Hunter W. Ellis

(GENERAL AUDIENCE ABSTRACT)

Dedication

This is where you put your dedications.

Acknowledgments

This is where you put your acknowledgement.

Contents

List of Figures

List of Tables

NLP is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages.

σ is the eighteenth letter of the Greek alphabet, and carries the 's' sound. In the system of Greek numerals, it has a value of 200.

Chapter 1

Introduction

Neuro-symbolic learning methods and concepts have been used in recent years to achieve results that standalone deep learning and symbolic programming methods have not been able to achieve. [EXAMPLES]. The emerging developments in the field of neuro-symbolic learning has created opportunities to explore applications and adaptations of these methods. The field of Reinforcement Learning (RL) has also made advances – demonstrating the paradigms effectiveness in creating agents that can perform complex tasks autonomously. The intersection of these two research areas has led to developments that have allowed agents to perform tasks with with both programatic interpretability and learned performance. [EXAMPLES].

1.1 Objectives

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.1.1 A sub-section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

1.2 Applications

1.3 Challenges

1.4 Contributions

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Chapter 2

Background

2.1 Markov Decision Process

2.2 Reinforcement Learning

2.3 Neuro-symbolic Architectures

Neuro-symbolic models can be broadly defined as models which focus on merging both *neural* and *symbolic* AI approaches to add value to the system. The term *nerual* refers to the use of artificial neural networks and the term *symbolic* typically refers to approaches based on explicit symbol manipulation. (CITATION taxonomy.pdf) The fundamental desire of neuro-symbolic models is to leverage both neural and symbolic approaches in a way that is favorable to the strengths of each approach and unfavorable to their weaknesses. The strengths neural models would include the ability to leverage raw data that may be too dificult/complex to semantically reason about, while the weaknesses may include the challenges faced when attempting to reason neural models. Conversely, the strengths of symbolic models include their ability to be highly explainable and verifiable, but they may have weaknesses in their reliance human input/understanding of a system. Thus, the promise of a neuro-symbolic architecture is a system which would be robust from training data, symbolically explainable, and be able to leverage human expert knowledge in its design.

In recent years many different neuro-symbolic models have been designed and deployed. Because the intersection of these two approaches may be quite broad and models can take many different forms be it is important to make distinctions between the various incarnations of neuro-symbolic architectures. Henry Kautz, in a 2005 article presents "six possible designs" patterns classifying each method in reference to their neural and symbolic interactions:(CITATION KAUTZ)

Neuro-symbolic Design Pattern	**Definition**	**Example**
Symbolic Neural symbolic	A symbolic input is fed into a neural network producing a symbolc output.	GPT
Symbolic[Neural]	A neural subroutine is evoked by a symbolic strategy.	AlphaGo
Neural[Symbolic]	A symbolic subroutine is evoked by a neural strategy.	ChatGPT accessing Wolfram for computations
Neural|Symbolic	A neural network converts a non-symbolic input into symbolic data to be symbolically processed.	Neuro-Symbolic Concept Learner
Neural:Symbolic → Neural	A symbolically represented dataset is used to train an neural network to predict a symbolic output	ANN-MPC
Neural_{Symbolic}	Symbolic rules are used to define the structures making up the neural network.	Logic Tensor Networks

2.4 Sim-to-Real

2.4.1 Transfer Learning

2.4.2 Multitask Learning

Chapter 3

Reinforcement Learning Platform

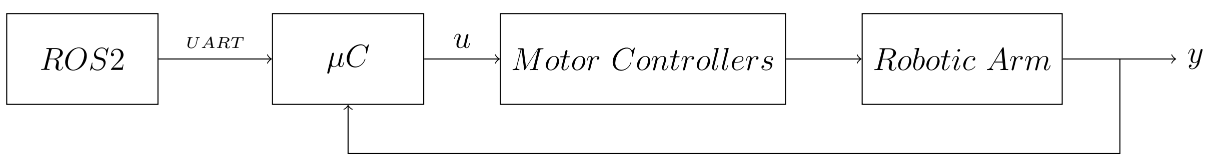
Given the problem space two environments were created to validate the neuro-symblic alogirithms mentioned in the previous sections

Custom real and cooresponding simulated environments were created to demonstrate the possibilities of the Neuro-Symbolic Reinforcement Learning Methods presented in the previous chapter. The environment was set up for basic object manipulation tasks. The cooresponding dataset used to train the model that can be seen as an extension of the CLEVR dataset – which instead prompts the agent to act on the objects in the environment.

3.1 Real-Time Hardware Environment

The physical hardware used to test the algorithms is made up of a 6-axis robot arm communicating over serial (UART) with a PC running the algorithms in a ROS2 Jazzy workspace. The rigid components of the arm including gears and structural pieces are made of 3D-printed Polylactic Acid (PLA). The robotic arm uses stepper motors, belts, and pulleys to articluate the 6 joints. The first 5 joints (J1 to J5) use bipolar NEMA 17 stepper motors, while the last joint J6, responsible for manipulating the end effector, uses a bipolar NEMA 8 stepper motor. The belts and pulleys are "off-the-shelf" GT2 timing belts and pulleys of varying sizes, used to the torque applied to each joint (belts and pulleys connect every motor to a joint with the exception of J6). Ball and shunt bearings of verious sizes are also used to reduce friction in the joints.

The electronic hardware is controlled by an ATmega328p microcontoller and 6 A4988 stepper motor drivers originally setup for controlling the stepper motors of a 3D-Printer. Custom firmware was written for the microcontroller to run the 6 stepper motor drivers simultaineously with a serial (UART) inturrupt to recieve control commands from the host PC.



Feedback system

3.2 Virtual Environment

The environment consists of the same six axis robot arm described in the previous section set up in the Gazebo Robotic Simulator (using a ROS Universal Robot Description File and Gazebo Simulation Description File). Within the arm's working envelop various basic 3-D shapes (i.e. spheres, cubes, cylinders, etc.) are present.

3.3 Robot Operating System

The Robot Operating System is a middleware suite used for robot software development. ROS workspaces consist of packages that interface with ROS libraries.

3.3.1 Description Package

The `arm_description` package is a ROS package that contains various files used to describe the physcial characteristics of the robot including its visual, collision, control, and forward kinematics.

The six-axis robotic arm model deployed in this package is a slightly modified version of an open source six-axis robot design with modfications made to some of the pulleys and end-effector design. The robot arm is made up of six joints (J1-J6) all described as revolute joints in the `arm_description`'s Universal Robot Description File (`.urdf`). Meshes for rendering the robot imported as `.stl` files and the meshes for collision areas are described by COLLADA (`.dae`) files. These meshes are linked together in the URDF in a kinematic chain to form the robotic arm manipulator. Control of the robot is accomplished through the ROS Jazzy control package.

3.4 Gazebo Robotics Simulator

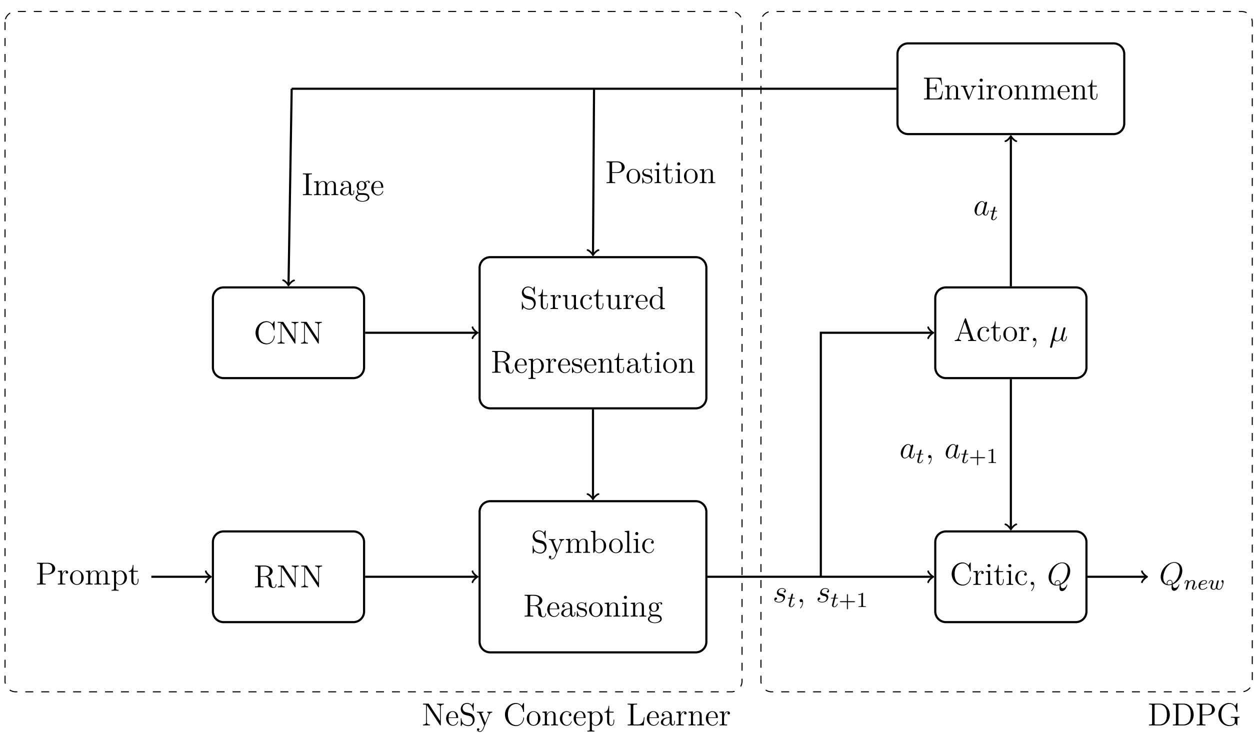
Gazebo is a robotic physics simulator developed by Open Robotics which integrates the ODE physics engine, ORGE rendering engine, and support code for sensor and actuator control integration. This environment leverages Gazebo Harmonic (the latest release of Gazebo at the time of writing) to visually render and simulate the physics of the robotic manipulator and the objects in its environment.

3.4.1 Simulation Package

3.5 Dataset

Chapter 4

Neuro-symbolic Reinforcement Learning Model



./images/architecture_overview.png

Figure 4.1: Overview of proposed architecture.

Shape	**Color** (r, g, b)	**Position** (x, y)	**Rotation** θ	**Velocity** (v_x, v_y, v_z, ω)
J1	(0, 0, 0)	(0, 0, 0)	0°	(0, 0, 0, 0)
J2	(0, 0, 0)	(0, 0.1, 0)	0°	(0, 0, 0, 0)
J3	(0, 0, 0)	(0, 0.2, 0)	0°	(0, 0, 0, 0)
J4	(0, 0, 0)	(0, 0.4, 0)	0°	(0, 0, 0, 0)
J5	(0, 0, 0)	(0, 0.4, 0)	0°	(0, 0, 0, 0)
J6	(0, 0, 0)	(0, 0.4, 0)	0°	(0, 0, 0, 0)
cube	(255, 0, 255)	(0.75, 0.75, 0)	0°	(0, 0, 0, 0)
cylinder	(255, 255, 0)	(-0.25, 1, 0)	0°	(0, 0, 0, 0)
rectangle	(0, 0, 255)	(0.5, 0.5, 0)	90°	(0, 0, 0, 0)
sphere	(0, 255, 0)	(0.6, -0.1, 0)	0°	(0, 0, 0, 0)

Label	**Color** (r, g, b)	**Position** (x, y)	**Rotation** θ	**Velocity** (v_x, v_y, v_z, ω)	**Symbol**
J1	(0, 0, 0)	(0, 0, 0)	0°	(0, 0, 0, 0)	null
J2	(0, 0, 0)	(0, 0.1, 0)	0°	(0, 0, 0, 0)	null
J3	(0, 0, 0)	(0, 0.2, 0)	0°	(0, 0, 0, 0)	null
J4	(0, 0, 0)	(0, 0.4, 0)	0°	(0, 0, 0, 0)	null
J5	(0, 0, 0)	(0, 0.4, 0)	0°	(0, 0, 0, 0)	null
J6	(0, 0, 0)	(0, 0.4, 0)	0°	(0, 0, 0, 0)	null
cube	(255, 0, 255)	(0.75, 0.75, 0)	0°	(0, 0, 0, 0)	null
cylinder	(255, 255, 0)	(-0.25, 1, 0)	0°	(0, 0, 0, 0)	avoid
rectangle	(0, 0, 255)	(0.5, 0.5, 0)	90°	(0, 0, 0, 0)	move
sphere	(0, 255, 0)	(0.6, -0.1, 0)	0°	(0, 0, 0, 0)	null
goal	(0, 0, 0)	(-1, 1, 0)	0°	(0, 0, 0, 0)	goal

Chapter 5

Simulations

Chapter 6

Discussion

Chapter 7

Conclusions