Deep Reinforcement Learning Nanodegree Program Syllabus

Become a Deep Reinforcement Learning Expert

Welcome to the Deep Reinforcement Learning Nanodegree program!

Before You Start

Educational Objectives: In this program, you'll learn the theory and practice driving recent advances in deep reinforcement learning. This program will cover the latest techniques used to create artificially intelligent agents that can solve a variety complex tasks, with applications ranging from gaming to finance to robotics. With the practical skills you gain in this program, you'll be able to understand the most cutting-edge research papers, and build an impressive portfolio containing your own coding implementations.

Prerequisite Knowledge: In order to succeed in this program, we recommend having significant experience with Python, and entry-level experience with probability and statistics, and deep learning architectures. Specifically, we expect you to be able to write a class in Python and to add comments to your code for others to read. Also, you should be familiar with the term "neural networks" and understand the differential math that drives backpropagation. If you feel you need to add to your Python and statistics skills, we suggest our Machine Learning program. If you'd like to learn more about neural networks and backpropagation, consider our Deep Learning program.

Length of Program: The program is comprised of 1 term, lasting 4 months. We expect students to work 10-15 hours/week on average. Make sure to set aside adequate time on your calendar for focused work.

Instructional Tools Available: Video lectures, Jupyter notebooks, personalized project reviews.

Contact Info

While going through the program, if you have questions about anything, you can reach us at drlnd-support@udacity.com.

Nanodegree Program Information

This program is designed to enhance your existing machine learning and deep learning skills with the addition of reinforcement learning theory and programming techniques. This program will not prepare you for a specific career or role, rather, it will grow your deep learning and reinforcement learning expertise, and give you the skills you need to understand the most recent advancements in deep reinforcement learning, and build and implement your own algorithms.

The term is comprised of 4 courses and 3 projects, which are described in detail below. Building a project is one of the best ways to demonstrate the skills you've learned, and each project will contribute to an impressive professional portfolio that shows potential employers your mastery of reinforcement learning and deep learning techniques.

Length of Program: 160 Hours*
Number of Reviewed Projects: 3

* The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.

Projects

Throughout this Nanodegree program, you'll master valuable skills by building the following projects:

- Navigation
- Continuous Control
- Collaboration and Competition

In the sections below, you'll find a description of each project, along with the skills you'll learn along the way.

Project: Navigation

Leverage neural networks to train an agent that learns intelligent behaviors from sensory data.

Course: Foundations of Reinforcement Learning

Lesson Title	Learning Outcomes
INTRODUCTION TO RL	• A friendly introduction to reinforcement learning.
The RL FRAMEWORK: THE PROBLEM	• Learn how to define Markov Decision Processes to solve real-world problems.
THE RL FRAMEWORK: THE SOLUTION	• Learn about policies and value functions. • Derive the Bellman equations.
DYNAMIC PROGRAMMING	• Write your own implementations of iterative policy evaluation, policy improvement, policy iteration, and value iteration.
MONTE CARLO METHODS	• Implement classic Monte Carlo prediction and control methods. • Learn about greedy and epsilon-greedy policies. • Explore solutions to the Exploration-Exploitation Dilemma.
TEMPORAL-DIFFERENCE METHODS	• Learn the difference between the Sarsa, Q-Learning, and Expected Sarsa algorithms.
SOLVE OPENAI GYM'S TAXI-V2 TASK	• Design your own algorithm to solve a classical problem from the research community.
RL IN CONTINUOUS SPACES	• Learn how to adapt traditional algorithms to work with continuous spaces.

Course: Value-Based Methods

Lesson Title	Learning Outcomes
DEEP LEARNING IN PYTORCH	• Learn how to build and train neural networks and convolutional neural networks in PyTorch.

DEEP Q-LEARNING	• Extend value-based reinforcement learning methods to complex problems using deep neural networks. • Learn how to implement a Deep Q-Network (DQN), along with Double-DQN, Dueling-DQN, and Prioritized Replay.
DEEP RL FOR ROBOTICS	• Learn from experts at NVIDIA how to use value-based methods in real-world robotics.

Project: Continuous Control

Train a robotic arm to reach target locations, or train a four-legged virtual creature to walk.

Course: Policy-Based Methods

Lesson Title	Learning Outcomes
INTRODUCTION TO POLICY-BASED METHODS	• Learn the theory behind evolutionary algorithms, stochastic policy search, and the REINFORCE algorithm. • Learn how to apply the algorithms to solve a classical control problem.
IMPROVING POLICY GRADIENT METHODS	• Learn about techniques such as Generalized Advantage Estimation (GAE) for lowering the variance of policy gradient methods. • Explore policy optimization methods such as Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO).
ACTOR-CRITIC METHODS	• Study cutting-edge algorithms such as Deep Deterministic Policy Gradients (DDPG).
DEEP RL FOR FINANCIAL TRADING	• Learn from experts at NVIDIA how to use actor-critic methods to generate optimal financial trading strategies.

Project: Collaboration and Competition

Train a system of agents to demonstrate collaboration or cooperation on a complex task.

Course: Multi-Agent Reinforcement Learning

Lesson Title	Learning Outcomes
INTRODUCTION TO MULTI-AGENT RL	• Learn how to define Markov games to specify a reinforcement learning task with multiple agents. • Explore how to train agents in collaborative and competitive settings.
CASE STUDY: ALPHAZERO	• Master the skills behind DeepMind's AlphaZero.