UDACITY

NANODEGREE PROGRAM SYLLABUS

Deep Reinforcement Learning Expert

Overview

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.

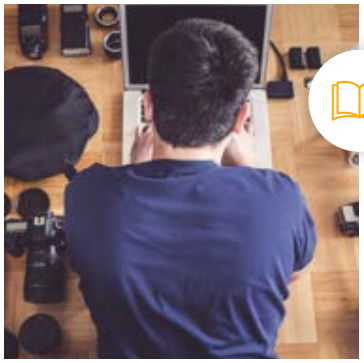
The term is comprised of 4 courses and 3 projects, which are described in detail below. Building a project is one of the best ways to demonstrate the skills you've learned, and each project will contribute to an impressive professional portfolio that shows potential employers your mastery of reinforcement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

IN COLLABORATION WITH

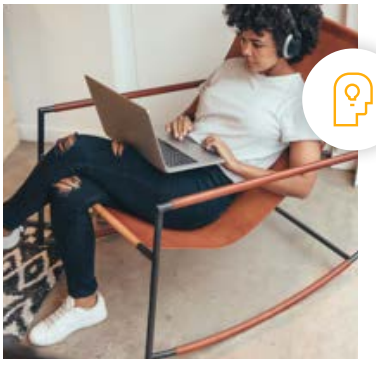
unity

NVIDIA DEEP LEARNING INSTITUTE

Estimated Time: 4 Months at 10-15hrs/week



Prerequisites: Experience with Python, Probability, Machine Learning, & Deep Learning.



Flexible Learning: Self-paced, so you can learn on the schedule that works best for you

Need Help? udacity.com/advisor Discuss this program with an enrollment advisor.

Course 1: Foundations of Reinforcement Learning

Master the fundamentals of reinforcement learning by writing your own implementations of many classical solution methods.

	LEARNING OUTCOMES	
LESSON ONE	**Introduction to RL**	• A friendly introduction to reinforcement learning.
LESSON TWO	**The RL Framework: The Problem**	• Learn how to define Markov Decision Processes to solve real-world problems.
LESSON THREE	**The RL Framework: The Solution**	• Learn about policies and value functions. • Derive the Bellman equations.
LESSON FOUR	**Dynamic Programming**	• Write your own implementations of iterative policy evaluation, policy improvement, policy iteration, and value iteration.
LESSON FIVE	**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.
LESSON SIX	**Temporal - Difference Methods**	• Learn the difference between the Sarsa, Q-Learning, and Expected Sarsa algorithms.

LESSON SEVEN	**Solve openai Gym's Taxi - V2 Task**	• Design your own algorithm to solve a classical problem from the research community.
LESSON EIGHT	**RL In Continuous Spaces**	• Learn how to adapt traditional algorithms to work with continuous spaces.

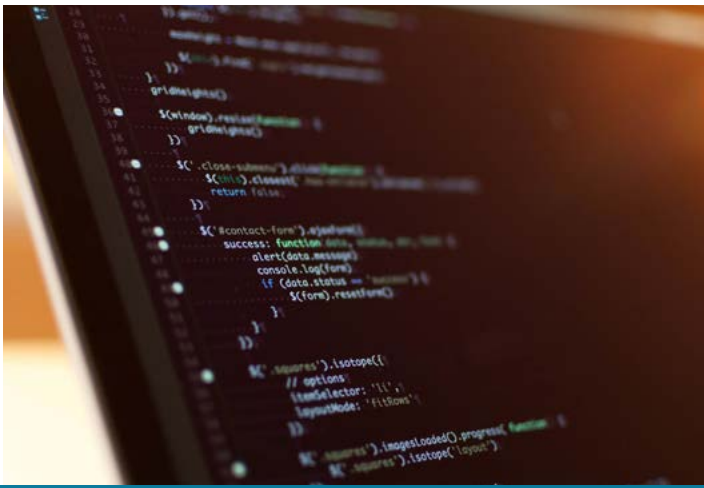
Course 2: Value-Based Methods

Apply deep learning architectures to reinforcement learning tasks. Train your own agent that navigates a virtual world from sensory data.

Course Project:
Navigation

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

	LEARNING OUTCOMES	
LESSON ONE	**Deep Learning in PyTorch**	• Learn how to build and train neural networks and convolutional neural networks in PyTorch.
LESSON TWO	**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.
LESSON THREE	**Deep RL for Robotics**	• Learn from experts at NVIDIA how to use value-based methods in real-world robotics.



Course 3: Policy-Based Methods

Learn the theory behind evolutionary algorithms and policy-gradient methods. Design your own algorithm to train a simulated robotic arm to reach target locations.

Course Project:
Continuous Control

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

		LEARNING OUTCOMES
LESSON ONE	**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.
LESSON TWO	**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).
LESSON THREE	**Actro-Critic Methods**	• Study cutting-edge algorithms such as Deep Deterministic Policy Gradients (DDPG).
LESSON FOUR	**Deep RL for Financial Trading**	• Learn from experts at NVIDIA how to use actor-critic methods to generate optimal financial trading strategies.

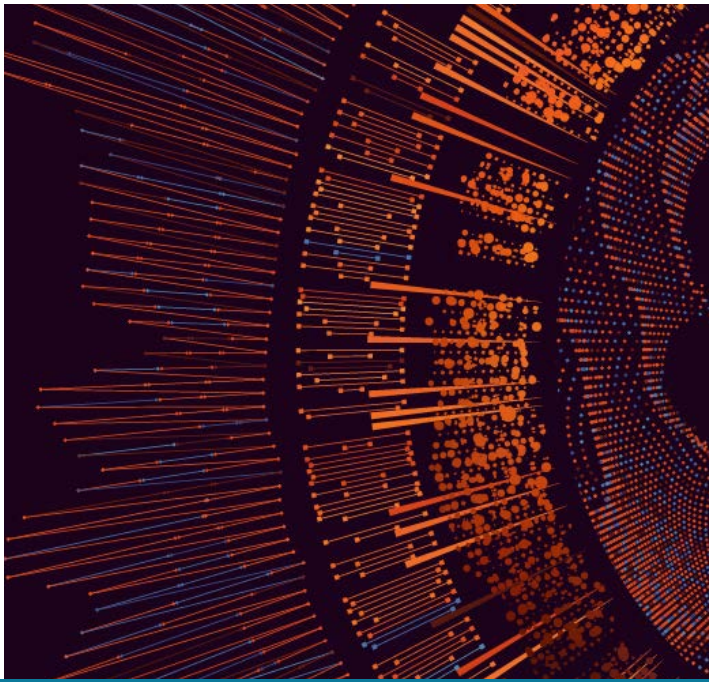
Course 4: Multi-Agent Reinforcement Learning

Learn how to apply reinforcement learning methods to applications that involve multiple, interacting agents. These techniques are used in a variety of applications, such as the coordination of autonomous vehicles.

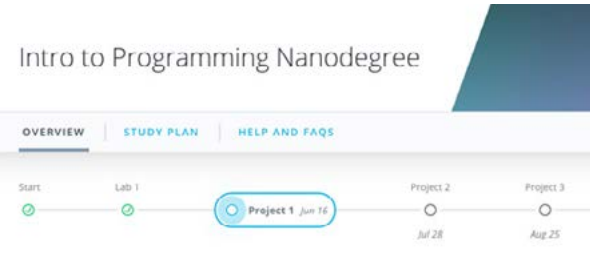
Course Project: Collaboration and Competition

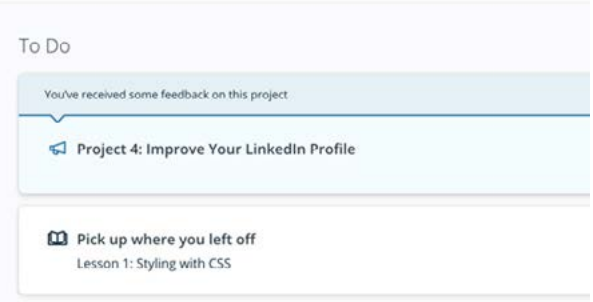
Train a system of agents to demonstrate collaboration or cooperation on a complex task.

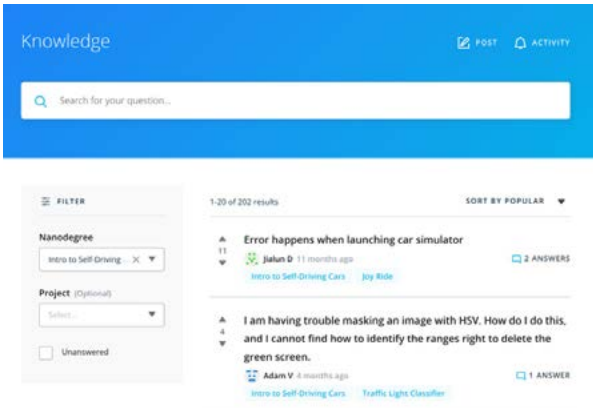
	LEARNING OUTCOMES	
LESSON ONE	**Introduction Multi-Agent RL**	• Learn how to define Markov games to specify a reinforcement learning task with multiple agents. • Explore how to train agents in collaborative and competitive settings.
LESSON TWO	**Case Study: Alphazera**	• Master the skills behind DeepMind's AlphaZero.

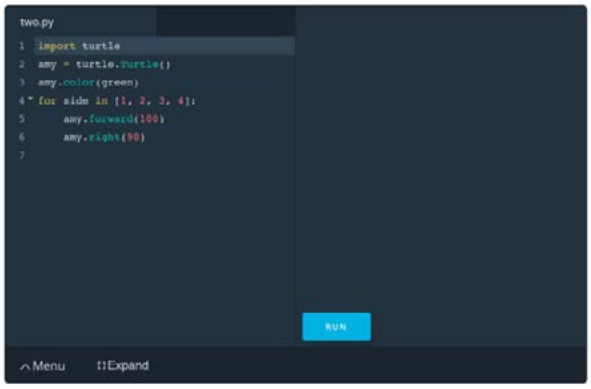


Our Classroom Experience









REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover in real-time how to solve the challenges that you encounter.

WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS

Create a custom study plan to suit your personal needs and use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

Luis Serrano

CURRICULUM LEAD

Luis was formerly a Machine Learning Engineer at Google. He holds a PhD in mathematics from the University of Michigan, and a Postdoctoral Fellowship at the University of Quebec at Montreal.

Alexis Cook

CURRICULUM LEAD

Alexis is an applied mathematician with a Masters in Computer Science from Brown University and a Masters in Applied Mathematics from the University of Michigan. She was formerly a National Science Foundation Graduate Research Fellow.

Arpan Chakraborty

INSTRUCTOR

Arpan is a computer scientist with a PhD from North Carolina State University. He teaches at Georgia Tech (within the Masters in Computer Science program), and is a coauthor of the book Practical Graph Mining with R.

Mat Leonard

CONTENT DEVELOPER

Mat is a former physicist, research neuroscientist, and data scientist. He did his PhD and Postdoctoral Fellowship at the University of California, Berkeley.

Learn with the Best

Cezanne Camacho

CURRICULUM LEAD

Cezanne is a machine learning educator with a Masters in Electrical Engineering from Stanford University. As a former researcher in genomics and biomedical imaging, she's applied machine learning to medical diagnostic applications.

Dana Sheahan

CURRICULUM LEAD

Dana is an electrical engineer with a Masters in Computer Science from Georgia Tech. Her work experience includes software development for embedded systems in the Automotive Group at Motorola, where she was awarded a patent for an onboard operating system.

Chhavi Yadav

CONTENT DEVELOPER

Chhavi is a Computer Science graduate student at New York University, where she researches machine learning algorithms. She is also an electronics engineer and has worked on wireless systems.

Juan Delgado

CONTENT DEVELOPER

Juan is a computational physicist with a Masters in Astronomy. He is finishing his PhD in Biophysics. He previously worked at NASA developing space instruments and writing software to analyze large amounts of scientific data using machine learning techniques.

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

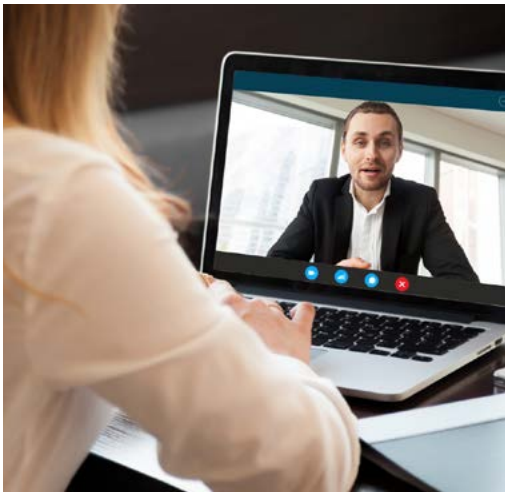
PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW



WHY SHOULD I ENROLL?

The demand for engineers with reinforcement learning and deep learning skills far exceeds the number of engineers with these skills. This program offers a unique opportunity for you to develop these in-demand skills. You'll implement several deep reinforcement learning algorithms using a combination of Python and deep learning libraries that will serve as portfolio pieces to demonstrate the skills you've acquired. As interest and investment in this space continues to increase, you'll be ideally positioned to emerge as a leader in this groundbreaking field.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

This program is designed to build on your existing skills in machine learning and deep learning. As such, it doesn't prepare you for a specific job, but instead expands your skills in the deep reinforcement learning domain. These skills can be applied to various applications such as gaming, robotics, recommendation systems, autonomous vehicles, financial trading, and more.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

This program offers an ideal path into the world of deep reinforcement learning—a transformational technology that is reshaping our future, and driving amazing new innovations in Artificial Intelligence. If you're interested in applying AI to fields such as gaming, robotics, autonomous systems, and financial trading, this is the perfect way to get started.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

To succeed in this Nanodegree program, we recommend you first take any course in Deep Learning equivalent to our Deep Learning Nanodegree program. You also need to be able to communicate fluently and professionally in written and spoken English.

Additionally, you should have the following knowledge:

Intermediate Python programming knowledge, including:

- Strings, numbers, and variables
- Statements, operators, and expressions
- Lists, tuples, and dictionaries

FAQs Continued

- Conditions & loops
- Generators & comprehensions
- Procedures, objects, modules, and libraries
- Troubleshooting and debugging
- Research & documentation
- Problem solving
- Algorithms and data structures

Basic shell scripting:

- Run programs from a command line
- Debug error messages and feedback
- Set environment variables
- Establish remote connections

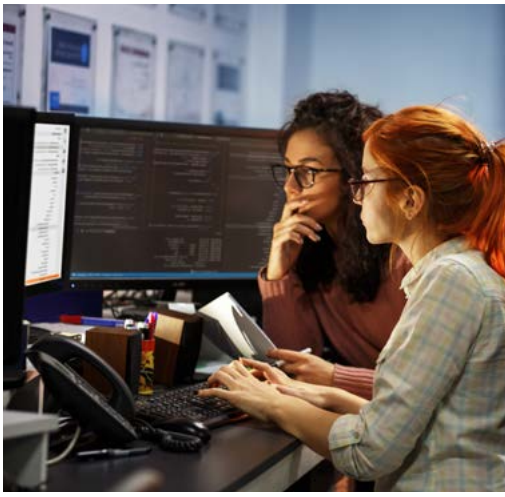
Basic statistical knowledge, including:

- Populations, samples
- Mean, median, mode
- Standard error
- Variation, standard deviations
- Normal distribution

Intermediate differential calculus and linear algebra, including:

- Derivatives & Integrals
- Series expansions
- Matrix operations through eigenvectors and eigenvalues

You will need to be able to communicate fluently and professionally in written and spoken English.



IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

We have a number of courses and programs we can recommend that will help prepare you for the program, depending on the areas you need to address. For example:

- **Intro to Machine Learning**
- **Artificial Intelligence Programming with Python Nanodegree program**
- **Deep Learning Nanodegree program**
- **Machine Learning Engineer Nanodegree program**

TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

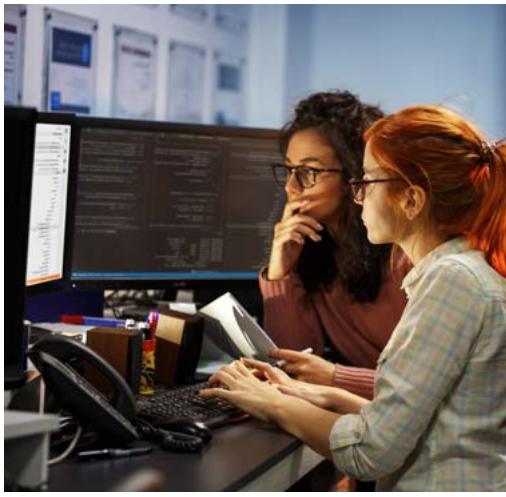
The Deep Reinforcement Learning Nanodegree program is comprised of content and curriculum to support three (3) projects. We estimate that students can complete the program in four (4) months working 10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

FAQs Continued

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.



SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

You will need a computer running a 64-bit operating system (most modern Windows, OS X, and Linux versions will work) with at least 8GB of RAM, along with administrator account permissions sufficient to install programs including Anaconda with Python 3.6 and supporting packages. Your network should allow secure connections to remote hosts (like SSH). We will provide you with instructions to install the required software packages.