Artificial Intelligence Nanodegree Syllabus

Contact Info

While going through the program, if you have questions about anything, you can reach us at support@udacity.com. For help from Udacity Mentors and your peers visit the Udacity Classroom.

Nanodegree Program Info

This program will teach you how to become a better Artificial Intelligence or Machine Learning Engineer by teaching you classical AI algorithms applied to common problem types. You will complete projects and exercises incorporating search, optimization, planning, and probabilistic graphical models which have been used in Artificial Intelligence applications for automation, logistics, operations research, and more. These concepts form the foundation for many of the most exciting advances in AI in recent years. Each project you build will be an opportunity to demonstrate what you've learned in your lessons, and become part of a career portfolio that will demonstrate your mastery of these skills to potential employers.

Prerequisite Skills

A well-prepared learner is able to:

- Intermediate Python Programming: experience with basic algorithms, common data structures, and Object Oriented Programming
- Intermediate Statistics and Linear Algebra: discrete and continuous distributions, vector spaces and matrices
- Basic knowledge of linear algebra and calculus.

Required Hardware

- 64-bit computer

Version: 3.0.0

Length of Program: 75 Days*

** This is a self-paced program and the length 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.*

Part 1: Introduction to Artificial Intelligence

Meet the instructional team including Sebastian Thrun, Peter Norvig, and Thad Starner who will be teaching you about the foundations of AI. Get acquainted with the resources available in your classroom & other important information about the program. Complete the lesson by building a Sudoku solver.

Project: Build a Sudoku Solver

Use constraint propagation and search to build an agent that reasons like a human would to efficiently solve any Sudoku puzzle.

Supporting Lessons

Lesson	Summary
Welcome to Artificial Intelligence	Welcome to the Artificial Intelligence Nanodegree program!
Knowledge, Community, and Careers	You are starting a challenging but rewarding journey! Take 5 minutes to read how to get help with projects and content.
Get Help with Your Account	What to do if you have questions about your account or general questions about the program.
Intro to Artificial Intelligence	An introduction to basic AI concepts and the challenge of answering "what is AI?"
Solving Sudoku With AI	In this lesson, you'll dive right in and apply Artificial Intelligence to solve every Sudoku puzzle.
Workspaces	Review the basic functionality of Workspaces—pre-configured development environments in the Udacity classroom for projects and exercises.
Setting Up Your Environment with Anaconda	If you do not want to use Workspaces, then follow these instructions to set up your own system using Anaconda, a popular tool to manage your environments and packages in python.

Project: Optimize Your GitHub Profile

Other professionals are collaborating on GitHub and growing their network. Submit your profile to ensure your profile is on par with leaders in your field.

Supporting Lessons

Lesson	Summary
Jobs in AI	Learn about common jobs in artificial intelligence, and get tips on how to stay active in the community.

Part 2: Constraint Satisfaction Problems

Take a deep dive into the constraint satisfaction problem framework and further explore constraint propagation, backtracking search, and other CSP techniques. Complete a classroom exercise using a powerful CSP solver on a variety of problems to gain experience framing new problems as CSPs.

Part 3: Classical Search

Learn classical graph search algorithms--including uninformed search techniques like breadth-first and depth-first search and informed search with heuristics including A*. These algorithms are at the heart of many classical AI techniques, and have been used for planning, optimization, problem solving, and more. Complete the lesson by teaching PacMan to search with these techniques to solve increasingly complex domains.

Part 4: Automated Planning

Learn to represent general problem domains with symbolic logic and use search to find optimal plans for achieving your agent's goals. Planning & scheduling systems power modern automation & logistics operations, and aerospace applications like the Hubble telescope & NASA Mars rovers.

Project: Build a Forward-Planning Agent

In this project you'll use experiment with search and symbolic logic to build an agent that automatically develops and executes plans to achieve their goals.

Part 5: Optimization Problems

Learn about iterative improvement optimization problems and classical algorithms emphasizing gradient-free methods for solving them. These techniques can often be used on intractable problems to find solutions that are "good enough" for practical purposes, and have been used extensively in fields like Operations Research & logistics. Finish the lesson by completing a classroom exercise comparing the different algorithms' performance on a variety of problems.

Part 6: Adversarial Search

Learn how to search in multi-agent environments (including decision making in competitive environments) using the minimax theorem from game theory. Then build an agent that can play games better than any human.

Project: Build an Adversarial Game Playing Agent

Extend classical search to adversarial domains, to build agents that make good decisions without any human intervention—such as the DeepMind AlphaGo agent.

Project: Improve Your LinkedIn Profile

Find your next job or connect with industry peers on LinkedIn. Ensure your profile attracts relevant leads that will grow your professional network.

Part 7: Probabilistic Models

Learn to use Bayes Nets to represent complex probability distributions, and algorithms for sampling from those distributions. Then learn the algorithms used to train, predict, and evaluate Hidden Markov Models for pattern recognition. HMMs have been used for gesture recognition in computer vision, gene sequence identification in bioinformatics, speech generation & part of speech tagging in natural language processing, and more.

Project: Part of Speech Tagging

In this project you will build a hidden Markov model (HMM) to perform part of speech tagging, a common pre-processing step in Natural Language Processing.

Part 8: After the AI Nanodegree Program

Once you've completed the last project, review the information here to discover resources for you to continue learning and practicing AI.

Udacity

Generated Fri May 1 15:53:56 PDT 2020