INDIVIDUAL LEARNERS

SCHOOL OF ARTIFICIAL INTELLIGENCE

Artificial Intelligence

Nanodegree Program Syllabus

Overview

This program will teach learners how to become a better artificial intelligence or machine learning engineer by teaching them classical AI algorithms applied to common problem types. They 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 students have learned in their lessons and become part of a career portfolio that will demonstrate their mastery of these skills to potential employers.

Program information

Estimated Time

3 months at 12-15hrs/week*

Skill Level

Advanced

Prerequisites

A well-prepared learner should have knowledge of algebra, calculus, statistics, and Python.

Required Hardware/Software

Learners 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 3.5 and supporting packages. Their network should allow secure connections to remote hosts (like SSH). We will provide learners with instructions to install the required software packages. Udacity does not provide any hardware.

*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. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Introduction to Artificial Intelligence

In this course, students will learn about the foundations of AI. They'll configure their programming environment to work on AI problems with Python. At the end of the course, they'll build a Sudoku solver and solve constraint satisfaction problems.

Course Project

Build a Sudoku Solver

Humans use reason to solve problems by decomposing the problem statement and incorporating domain knowledge to limit the possible solution space. Use a technique called constraint propagation together with backtracking search to make an agent that only considers reasonable solution candidates and efficiently solves any Sudoku puzzle. This approach appears in many classical AI problems, and the solution techniques have been extended and applied to diverse problems in bioinformatics, logistics, and operations research. Demonstrate some basic algorithms knowledge and learn to use constraint satisfaction to solve general problems.

Lesson 1

Welcome to Artificial Intelligence

- Learn about the resources available to help one succeed.

Lesson 2

Intro to Artificial Intelligence

- Consider the meaning of "artificial intelligence."
- Be able to define core concepts from AI including "agents," "environments," and "states."
- Learn the concept of "rational" behavior for AI agents.

Lesson 3

Solving Sudoku with AI

- Express logical constraints as Python functions.
- Use constraint propagation and search to solve all Sudoku puzzles.

Lesson 4

Setting Up Your Environment

- Use Conda to configure and manage Python packages and dependencies.

Lesson 5

Constraint Satisfaction Problems

- Learn to represent problems in terms of logical constraints.
- Use constraint propagation to limit the potential solution space.
- Incorporate backtracking search to find a solution when the set of constraints is incomplete.
- Use Z3 Solver to solve constraint satisfaction problems.

Course 2

Classical Search

In this course students will 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.

Lesson 1

Introduction

- Learn about the significance of search in AI

Lesson 2

Uninformed Search

- Learn uninformed search techniques including Depth-First, Search, Breadth-First Search, and Uniform Cost Search.

Lesson 3

Informed Search

- Learn informed search techniques (using heuristics) including A* Search.
- Understand admissibility and consistency conditions for heuristics.

Lesson 4

Classroom Exercise: Search

- Implement informed and uninformed search for Pacman.

Lesson 5

Additional Topics: Search

- List of external resources for one to continue learning about search.

Course 3

Automated Planning

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

Course Project

Build a Forward Planning Agent

Intelligent agents are expected to act in complex domains where their goals and objectives may not be immediately achievable. They must reason about their goals and make rational choices of actions to achieve them. Build a system using symbolic logic to represent general problem domains and use classical search to find optimal plans for achieving the agent's goals. Demonstrate an understanding of classical optimization and search algorithms, symbolic logic, and domain independent planning.

Lesson 1

Symbolic Logic & Reasoning

- Learn propositional logic (propositions and statements).
- Learn first-order logic (quantifiers, variables, and objects).
- Encode problems with symbolic constraints using First Order Logic.

Lesson 2

Introduction to Automated Planning

- Learn to define planning problems.

Lesson 3

Classical Planning

- Learn high-level features of automated planning techniques using search and symbolic logic including forward planning, backwards planning, and hierarchical planning.
- Explore planning heuristics and planning graphs.

Lesson 4

Additional Topics: Search

- List of external resources for you to continue learning about search.

Course 4

Optimization Problems

In this course students will 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 and logistics. Students will finish the lesson by completing a classroom exercise comparing the different algorithms’ performance on a variety of problems.

Lesson 1

Introduction

- Introduce iterative improvement problems that can be solved with optimization.

Lesson 2

Hill Climbing

- Learn random hill climbing for local search optimization problems.

Lesson 3

Simulated Annealing

- Learn to use simulated annealing for global optimization problems.

Lesson 4

Genetic Algorithms

- Explore and implement genetic algorithms that keep a pool of candidates to solve optimization problems.

Lesson 5

Classroom Exercise: Optimization Problems

- Compare optimization techniques on a variety of problems.

Lesson 6

Additional Optimization Topics

- Learn about improvements and optimizations to optimization search including late acceptance hill climbing, basin hopping, and differential evolution.

Course 5

Adversarial Search

In this course students will 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.

Course Project

Build an Adversarial Game Playing Agent

AI agents acting in the real world have to "hope for the best, but prepare for the worst." Write an agent that can play games with superhuman performance in an adversarial environment. The principles of adversarial search provide a foundation for autonomous agents acting in the real world, and for understanding modern advances in AI like DeepMind's AlphaGo Zero.

Lesson 1

Search in Multi-Agent Domains

- Understand "adversarial" problems and applications (e.g., multi-agent environments).
- Extend state space search techniques to domains the agents do not fully control.
- Learn the minimax search technique.

Lesson 2

Optimizing Minimax Search

- Apply depth-limiting to overcome limitations in basic minimax search.
- Apply alpha-beta pruning to overcome limitations in basic minimax search.

Lesson 3

Extending Minimax Search

- Extend adversarial search to non-deterministic domains and domains with more than two players.

Lesson 4

Additional Adversarial Search Topics

- Understand other adversarial search techniques such as Monte Carlo Tree Search.
- List of external resources for one to continue learning about adversarial search.

Course 6

Fundamentals of Probabilistic Graphical Models

In this course students will 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 and part of speech tagging in natural language processing, and more.

Course Project

Part of Speech Tagging

Probabilistic models allow agents to better handle the uncertainty of the real world by explicitly modeling their belief state as a distribution over all possible states. Use a Hidden Markov Model (HMM) to perform part of speech tagging, a common pre-processing step in natural language processing. HMMs have been used extensively in NLP, speech recognition, bioinformatics, and computer vision tasks.

Lesson 1

Introduction to Probabilistic Models

- Model probability distributions based on a given set of parameters in a real-world use case using discrete distributions.

Lesson 2

Probability

- Review key concepts in probability including discrete distributions, joint probabilities, and conditional probabilities.

Lesson 3

Bayes Nets

- Efficiently encode joint probabilities in Bayes networks.

Lesson 4

Inference in Bayes Nets

- Learn about inference in Bayes networks through exact enumeration with optimizations.
- Learn techniques for approximate inference in more complex Bayes networks.

Lesson 5

Hidden Markov Models

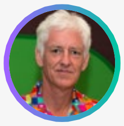
- Learn parameters to maximize the likelihood of model parameters to training data.
- Determine the likelihood of observing test data given a fixed model.
- Learn an algorithm to Identify the most likely sequence of states in a model given some data.

Lesson 6

Dynamic Time Warping

- Learn the dynamic time warping algorithm for time-series analysis.

Meet your instructors.



Peter Norvig

Research Director at Google

Peter Norvig is a director of research at Google and is co-author of Artificial Intelligence: A Modern Approach, the leading textbook in the field.

Sebastian Thrun

Founder and Executive Chairman at Udacity

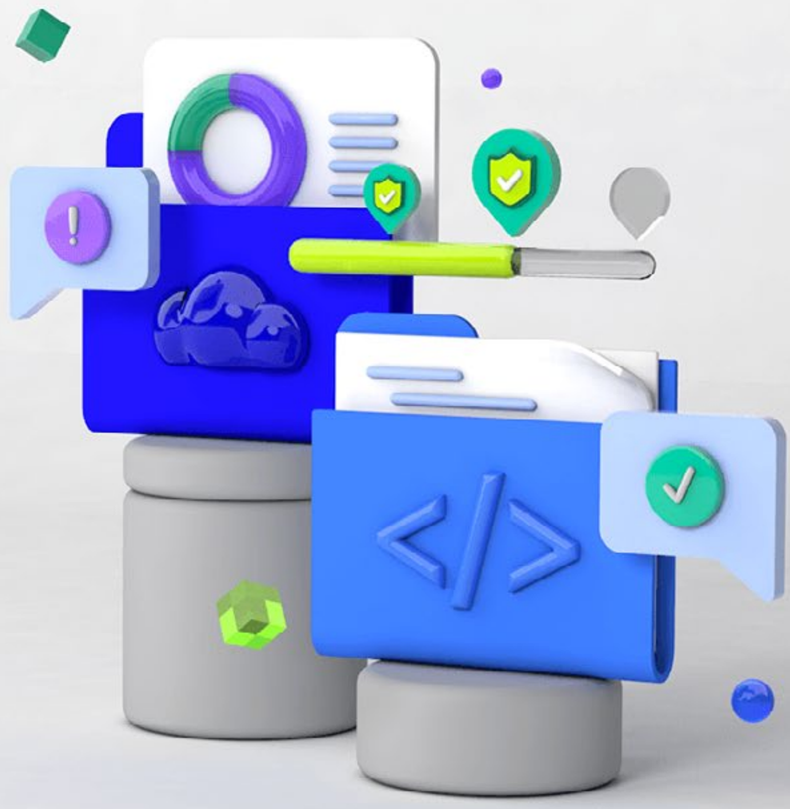
As the founder and president of Udacity, Sebastian's mission is to democratize education. He is also the founder of Google X, where he led projects including the self-driving car, Google Glass, and more.

Thad Starner

Professor of Computer Science at Georgia Tech

Thad Starner is the director of the Contextual Computing Group (CCG) at Georgia Tech and is also the longest-serving technical lead/manager on Google's Glass project.

Udacity’s learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren’t just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

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

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

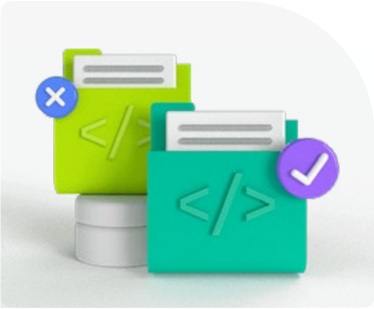
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

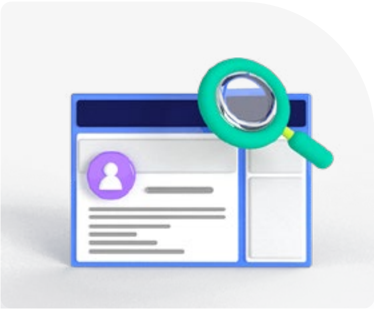
Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

Learn more at

www.udacity.com/online-learning-for-individuals →