Model AI Assignments 2018

Todd W. Neller
Gettysburg College
tneller@gettysburg.edu

Zack Butler
Rochester Institute of Technology
zjb@cs.rit.edu

Nate Derbinsky
Northeastern University
n.derbinsky@northeastern.edu

Heidi Furey
Manhattan College
hfurey01@manhattan.edu

Fred Martin
University of Massachusetts Lowell
fred_martin@uml.edu

Michael Guerzhoy
University of Toronto
guerzhoy@cs.toronto.edu

Ariel Anders
Massachusetts Institute of Technology
aanders@csail.mit.edu

Joshua Eckroth
Stetson University
jeckroth@stetson.edu

Abstract

The Model AI Assignments session seeks to gather and disseminate the best assignment designs of the Artificial Intelligence (AI) Education community. Recognizing that assignments form the core of student learning experience, we here present abstracts of seven AI assignments from the 2018 session that are easily adoptable, playfully engaging, and flexible for a variety of instructor needs. Assignment specifications and supporting resources may be found at http://modelai.gettysburg.edu.

Go For a Walk! Pedestrian-Friendly A* with Learned Cost Functions

Zack Butler

This assignment is a two-part project based on path planning for pedestrians in an urban environment. In the first part of the project, students use real-world data collected from OpenStreetMap and digital elevation data to build an A* path planner. In particular, they need to consider the distance as well as elevation gained or lost on each edge, and provide a cost function and admissible heuristic derived from their ad-hoc estimates and/or published pedestrian models. Extensions use OpenStreetMap metadata to consider planning for bicycles and cars. In the second part of the project, the students in the course walk through the environment to collect actual data on the time taken to follow different paths. This data is shared to the entire class to build an overall data set. Students then perform regression over the features in their cost function using this data set to develop a new cost function. They can also compare the cost function derived from only their individual data to that derived from the overall class data set, and as an extension to the assignment, compare different types of regression algorithms.

Solve a Maze via Search

Nate Derbinsky

An important aspect of an introductory AI course is exposure to the complexities and tradeoffs involved in deploying an application that integrates one or more intelligence techniques. This project assumes students have studied (un)informed search algorithms (e.g. via Pacman Projects; (Neller et al. 2010)), and guides them through the process of developing an image-based maze solver. The assignment begins by assuming a clean grid-based representation, and has students formulate maze-solving as a search problem. They are then tasked with applying basic computer-vision and image-processing techniques (via OpenCV; (Bradski 2000)) to convert an input image into the assumed problem representation, and then project the solved path onto the picture for the user to see. Throughout the project, students are pushed to visualize their work, verify incremental steps, and analyze tradeoffs between algorithmic performance and solution feasibility/quality. The assignment concludes with a variety of extensions, allowing students to pursue more challenging aspects that suit their interests.

A Module on Ethical Thinking about Autonomous Vehicles in an AI Course

Heidi Furey and Fred Martin

A computer science faculty member and a philosophy faculty member collaborated in the development of a one-week introduction to ethics which was integrated into a traditional AI course. The module provided a brief introduction to ethics and utilitarianism, and then engaged students in considering the classic "Trolley Problem," which has gained contemporary relevance with the emergence of autonomous vehicles. Students used this introduction to ethics in thinking through the implications of their final projects. Results from the course final examination showed that nearly all students in the course understood the Trolley Problem (62 of 63 students). This Model AI assignment submission includes the presentation materials used in lecture, student exercises (recommended to be done in class, in pairs), and suggested readings.

Understanding How Neural Networks Recognize Faces

Michael Guerzhoy

In this assignment, students build a feedforward neural network for face recognition using TensorFlow. Students then visualize the weights of the neural networks they train. The visualization allows students to understand feedforward one-hidden layer neural networks in terms of template matching, and allows students to explore overfitting. Using a framework such as TensorFlow allows for the students to be able to run a variety of experiments in order to obtain interesting visualizations.

Understanding How Recurrent Neural Networks Model Text

Michael Guerzhoy

In this assignment, students are introduced to Recurrent Neural Networks (RNNs) and explore the mechanism that allows RNNs to model English text character-by-character. Students learn to think of an RNN as a state machine. The assignment was originally used in a third-year neural networks and machine learning course, and could be adapted to an Intro AI course.

Recurrent Neural Networks have recently been shown to be remarkably effective at modelling complex time-series, including English text, chacater by character. Realistic-seeming "fake" English text can be generated using RNN models. Many students and practitioners tend to view RNNs as black boxes. Thsi assignment forces students to understand how RNNs model complex time-series by having students explain the some properties of the outputs generated by the model in reference to the specific weights of the model.

In this assignment, students are provided with a "vanilla" RNN model that was trained on a corpus of Shakespeare play. The students are asked to write code to generate "fake" text from the model, and to then explain several properties of the text. Students are encouraged to find new interesting properties of the model for bonus marks.

Robot Juggling

Ariel Anders

In this assignment students learn how to control a robot to juggle a ball. To achieve this goal, students will program a velocity controlled robot, such that it causes the ball to bounce with some desired periodic motion. Specifically, students will implement a hybrid controller that uses a *mirror control law* within the framework of a 2D physics simulator.

Similar to many control assignments, the theoretical learning objectives deal with analyzing system responses and stability criterion. Additionally, designing and implementing the controller will give students a chance to translate theoretical knowledge into real-world practice. This assignment differentiates from most introductory control assignments because the dynamics of robot juggling is highly nonlinear, due to the collisions with the robot.

This assignment is tailored to Engineering Sophomore or Junior undergraduate students, with minimal programming experience. The assignment is implemented with Python, using readily accessible packages from Anaconda Cloud. Another highlight of this assignment is that it is free, open source, and capable of running across popular operating systems, like Windows, Mac, and Ubuntu.

Biductive Computing: Several Variants of a Universal Paradigm

Joshua Eckroth

This Model AI Assignment allows students to practice with logic programming and constraint programming in Prolog and ProbLog using a paradigm we call "biductive computing," i.e. supporting both deductive and abductive inferences with the same code. This assignment includes four variants of biductive computing: database querying, planning, parsing, and probabilistic reasoning. In each variant, we describe a computational problem in a real-world domain, explain the biductive aspects of the desired implementation, provide test cases to measure correct solutions, and suggest possible enhancements. Each assignment variant may be used in an introductory AI course and, in the case of the database querying variant, may serve as a first assignment in Prolog.

References

Bradski, G. 2000. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer* 25(11):120–123.

Neller, T. W.; DeNero, J.; Klein, D.; Koenig, S.; Yeoh, W.; Zheng, X.; Daniel, K.; Nash, A.; Dodds, Z.; Carenini, G.; Poole, D.; and Brooks, C. 2010. Model AI assignments. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, July 13-14, 2010, Atlanta, Georgia, USA.*, 1919–1921.