Model AI Assignments 2020

Todd W. Neller,[1] Stephen Keeley,[2] Michael Guerzhoy,[2,3,4] Wolfgang Hoenig,[5] Jiaoyang Li,[6] Sven Koenig,[6] Ameet Soni,[7] Krista Thomason,[7] Lisa Zhang,[3] Bibin Sebastian,[3] Cinjon Resnick,[8] Avital Oliver,[9] Surya Bhupatiraju,[10] Kumar Krishna Agrawal,[11] James Allingham,[12] Sejong Yoon,[13] Jonathan Chen,[14] Tom Larsen,[14] Marion Neumann,[14] Narges Norouzi,[15] Ryan Hausen,[15] Matthew Evett[16]

[1]Gettysburg College, [2]Princeton University, [3]University of Toronto, [4]Li Ka Shing Knowledge Institute, [5]California Institute of Technology, [6]University of Southern California, [7]Swarthmore College, [8]New York University, [9]Google, [10]Massachusetts Institute of Technology, [11]Indian Institute of Technology, Kharagpur, [12]University of Cambridge, [13]The College of New Jersey, [14]Washington University in St. Louis, [15]University of California, Santa Cruz, [16]Eastern Michigan University

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 nine AI assignments from the 2020 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.

Predicting and Preventing Deaths in the ICU: Designing and Analyzing an AI System

Stephen Keeley, Michael Guerzhoy

In this assignment, students build a system for predicting patient deaths in the Intensive Care Unit (ICU) using the large PhysioNet Computing in Cardiology Challenge 2012 dataset. For each patient in the dataset, demographic variables and time series of physiological variables were collected during their stay in the ICU. Students use simulation and model the impact of the system's flagging higher-risk patients for intervention by physicians on saving lives in the ICU. The assignment illustrates the process of building and evaluating the potential impact of an AI system when only observational data is available.

The assignment is designed in R/tidyverse and is suitable for students who are taking their first programming course. We provide a translation to Python/Pandas/scikit-learn.

A Project on Multi-Agent Path Finding (MAPF)

Wolfgang Hoenig, Jiaoyang Li, Sven Koenig

We provide project material for the emerging topic of multi-agent path finding (MAPF), where agents (typically: robots) operate in a known environment and are tasked with moving from their current locations to their respective goal locations without colliding with the environment or each other. MAPF is a key task for autonomous warehousing and just-in-time manufacturing. Traditional search algorithms in the joint location space, such as A*, scale poorly in the number of agents. Therefore, one needs to develop search algorithms that exploit the problem structure better to gain efficiency. We provide project material for two such search algorithms, namely the incomplete and suboptimal prioritized planning and the complete and optimal (but slower) conflict-based search.

The project material is designed for students with prior knowledge of A* and working knowledge of Python. We provide a textbook-style overview of MAPF for the teacher and students that describes the MAPF problem and two solution approaches; lecture slides for the teacher; a code framework that includes result visualization and a generator for random MAPF instances; a handout for the students that guides them to finish the implementations with a focus on single-agent path finding, constraint generation and resolution; and sample solutions for the teacher.

A Module for Introducing Ethics in AI: Detecting Bias in Language Models

Ameet Son, Krista Thomason

The widespread adoption of machine learning algorithms has raised awareness of the potential biases these systems can encode and perpetuate. This course module introduces students to this concept and has them detect biases in widely used language models. In this lab, students first work with programs that demonstrate the usefulness of word embedding algorithms in finding relationships between words. Students use an implementation of the Word Embedding Association Test (WEAT) (Caliskan, Bryson, and Narayanan 2017) to detect gender and racial bias encoded in word embeddings. Students devise their own bias test, experiment with different data sources (e.g., Twitter, Wikipedia) and

interpret their findings. Finally, this assignment presents a real-world scenario that uses natural language systems in a health care setting. Students must apply what they learned in this module to discuss ethical concerns with the proposed system. This assignment was designed for a non-majors course on Ethics and Technology but can be used in any course looking to introduce discussions of ethical issues in the field of computer science.

Gesture Recognition using Convolutional Neural Networks

Lisa Zhang, Bibin Sebastian

In this assignment, students build a Convolutional Neural Network (CNN) to recognize American Sign Language (ASL) hand gestures. While doing so, students experience the entire machine learning workflow, and learn best practices for debugging neural networks. In particular, students collect and clean their own photos demonstrating ASL gestures. The teaching team pools together data collected by the entire class and provides it to students. In the meantime, students build a CNN by first having a simple model "overfit" or memorize a small dataset to show their network's correctness. Once the entire class's dataset is available, students train their CNN, tune hyperparameters, and report results. There is also a module where students use pre-trained AlexNet weights to understand transfer learning to obtain better performance. This assignment leads naturally to a discussion about fairness in machine learning, since the training data excludes demographics not represented by students in the class. It is also possible to run a competition on an unseen test set. Interested instructors can contact the authors for a secret test set not previously shared with students.

Wasserstein GAN — Depth First Learning

Cinjon Resnick, Avital Oliver, Surya Bhupatiraju, Kumar Agrawal, James Allingham

We present a Model AI assignment targeting the Wasserstein GAN papers (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017). It consists of five sessions, each of which roughly correspond to 6-10 hours of work (a 1.5 hour class, readings, and assignments). The sessions have both required and optional readings, as well as problems where students must demonstrate their understanding by either proving a mathematical statement, answering a question, or implementing code. The assignment is modeled after the Depth First Learning approach and was originally created and run by a South African group organized via the Deep Learning Indaba and led by James Allingham. Starting from source material, it builds up the requisite tree of knowledge that then enables students to attain a strong grasp of the papers when they finally read them in weeks four and five. Secondarily, we will also present how our pedagogy has evolved since last year's submission to EAAI (DeepStack). In the intervening period, we have accrued three more study guides and are in the midst of running a fellowship to both spread the gospel as well as build our compendium. We will cover in our talk how each iteration has allowed us to run experiments that have successively improved the outcome.

PyPlat: A Flexible Platform Game Project

Sejong Yoon

PyPlat is a small, game-based AI project that requires students to design and implement game-playing agents in Python. It is a platform game, thus agents must jump over pits and/or climb the ladder to collect reward items while avoiding obstacles and adversarial agents. The project has two design goals: (a) The game domain should be sophisticated enough to challenge advanced students while simple enough for beginners to understand the material, stay motivated, and complete the assignments. Unlike a well-known prior art of The Mario Project (Taylor 2011), PyPlat provides a relatively small search space without sacrificing the complexity and interestingness as a full, complete platform game. The source code is only about 1,000 lines in total thus students without much programming experience can understand the entire system quickly. (b) Instructors can easily modify artistic aspects of the game while maintaining the core mechanics, without investing excessive effort. The whole level design (platforms, obstacles, reward items, number, speed, and location of adversarial agents) can be done in a single Python script file in the plain text format. The assignment comes with two presets of game levels and artworks, as well as sample handouts with setup instruction, and suggestions for customized assignments.

Exploring Unfairness and Bias in Data

Jonathan Chen, Tom Larsen and Marion Neumann

It is natural to assume that a model built from "real-world" data will inherently represent the world-at-large. However, this is not the case as seen in recent instances of models behaving in unexpectedly biased ways. We believe that one long term solution to this problem is to build a curriculum that inspires students to actively think about their data and the potential for it to be biased. Here, we present a Jupyter Notebook-based assignment exploring how bias can be introduced into a model using an example of gender-bias in credit history used to predict creditworthiness. We use an unbalanced data set to demonstrate how model evaluation methods like classification accuracy can be misleading even with standard procedures like dataset splitting and cross-validation, allowing bias to remain undetected. Then we consider whether pre- and post-processing strategies such as ignoring gender altogether will help improve the fairness of our predictions. We conclude by prompting students to discuss whether these methods can mitigate unfairness in AI and to contribute their ideas about how to tackle this problem.

Playing Against Adversary and Stochastic Agents in Connect Four Game

Narges Norouzi and Ryan Hausen

We designed a variation of the Connect Four game for students to implement and understand the scope of adversarial and stochastic games. The assignment is designed to be adaptable, modular, interactive, and effective in showing the

differences and similarities between adversarial games and stochastic games. Topics covered in this assignment include Adversarial Games, Stochastic Games, the Minimax Algorithm, and the Expectimax Algorithm. Throughout this assignment, students will implement Alpha-Beta pruning and Expectimax Search algorithms and are provided with a GUI to test the performance of their algorithms. The target audience of this assignment is undergraduate Computer Science or Computer Engineering students taking an Artificial Intelligence course. This assignment is easily adaptable by instructors and is implemented in a modular structure. Instructors will have the flexibility of assigning any of the two modules in the assignment (AIPlayer playing against a RandomPlayer (stochastic agent) or a HumanPlayer (adversary agent)). The assignment also provides the structure that can be modified by instructors to work with any other board game.

Graphical Networked Checkers Bots Assignment

Matthew Evett

In this assignment, intended for an introductory AI course, students develop an autonomous bot that plays a game of checkers against other bots across a network. The assignment provides skeleton code of a Player module, which provides a graphical rendering of each game as it is played. The complexity of the networking component is mostly hidden from the students as they need only to make use of a provided intermediary module that uses Java's remote method invocation (RMI) technology to allow the student bots to compete with each other. This allows the students to concentrate on the core AI component, a heuristic search algorithm.

The assignment is the culmination of the study of search techniques, especially A* and minimax, and methods for comparing the effectiveness of heuristics. Students are encouraged to conduct their own research to identify strong heuristics for checkers, and to incorporate that research into their bots.

This assignment has been used for several years and students regularly report in their course reviews that it was their favorite assignment across all of their CS courses. During the class-wide tournament at the end of the assignment, the cheers and corresponding groans of dismay are heartening for any educator.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In Precup, D., and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 214–223. International Convention Centre, Sydney, Australia: PMLR.

Caliskan, A.; Bryson, J. J.; and Narayanan, A. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356(6334):183–186.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of Wasserstein GANs. *CoRR* abs/1704.00028.

Taylor, M. 2011. Teaching reinforcement learning with Mario: An argument and case study. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-2011)*, 1737–1742. AAAI Press.