Work on AIMA-Python

Finish AIMA-python algorithms and add explanatory notebooks

Aman Deep Singh
Github: ad71
Email: ads6may@gmail.com

ABSTRACT

The aim of this project is to finish incomplete implementations of the algorithms in aima-python, add tests for them and add detailed jupyter notebooks explaining the functioning and the usage of the algorithms.

PROJECT PROPOSAL

1. The aima-python repository currently has 6 incomplete algorithms, one of which is currently being worked on. I propose to complete the implementation of these algorithms and add tests for them.
2. I will write detailed jupyter notebooks for the algorithms which do not have notebook sections, explaining the intuition behind the algorithms and providing graphing and visualization wherever necessary. These notebooks will include guides and examples for using the algorithms on real-world problems. The notebooks will also address the advantage of certain algorithms over others in different scenarios.
3. Notebook sections for some algorithms are inadequately or incomprehensibly written and require a revisit. I will rewrite some parts to accomodate the ideas in the previous point.
4. I will add tests for the newly added algorithms as well as for the ones that are missing tests. Some tests for the existing algorithms are quite trivial and serve no real purpose. I will add tests for these algorithms too in an attempt to increase coverage.
5. Lower down the priority list, I will write simple GUI applications for some of the classic algorithms as I believe certain concepts are explained best when the student can interact with an application.
6. Some notebooks have a corresponding `apps` notebook. I plan to extend this idea for a few notebooks. For example a `search-apps` notebook will be very helpful as search algorithms can be applied to almost all optimization problems.
7. If time permits, I also plan to remove dependencies on third-party libraries like `networkx` . I

will try my best to rewrite these parts using the classes already built into `aima-python`.

8. I have quite a lot of experience with JavaScript as well and I am willing to work on `aima-exercises` too, if required. I have included links to a few JavaScript projects in this document.

In case of unforeseen problems, I might be unable to work on all these points, but I will try my best to complete the tasks with the highest priority. I do not have other commitments this summer and I am willing to work 45-50 hours a week on this project. I do not yet have a plan on how to start working on this project as I have a decent knowledge about almost all the concepts in the book and am willing to work however my mentors see fit. Given a choice, I would probably start with the `logic`, or `mdp` modules.

MY WORK ON THE REPOSITORY

The following is a summary of my contributions to the project.

1. Genetic Algorithm section in `search.ipynb` #702

2. Value Iteration explanation in `mdp.ipynb` #736

3. Policy Iteration explanation and various enhancements to `mdp.ipynb` #743

4. Created `mdp_apps.ipynb` for applications of MDPs. #743 and #782

5. Hill-Climbing section in `search.ipynb` #787

6. TT-Entails and TT-Check-All section in `logic.ipynb` #793

7. To-CNF explanation in `logic.ipynb` #802

8. PL-FC-Entails section in `logic.ipynb` #818

9. DPLL section in `logic.ipynb` #823

10. WalkSAT section in `logic.ipynb` #823

11. Min-Conflicts section in `logic.ipynb` #841

12. Simulated-Annealing section in `search.ipynb` #866

13. Rewrote `EightPuzzle` and `NQueensProblem` sections in `search.ipynb` #848

14. Refactored `EightPuzzle` class to work with search algorithms #807

15. Refactored `NQueensProblem` class to work with search algorithms #848

16. Added tests for `mdp.py` #722

17. Added tests for `EightPuzzle` and `astar_search` #807

18. Added tests for `pl_fc_entails` #818

19. Added tests for `NQueensCSP` and `min_conflicts` #841

20. Added tests for `NQueensProblem` , `breadth_first_tree_search` , `depth_first_tree_search` , `astar_search` and `uniform_cost_search` #848

21. Added tests for `unify` , `tt_entails` , `pl_resolution` , `WalkSAT` and `dpll_satisfiable` #854

22. Changed plotting function for `NQueensCSP` #847

23. `recombine_uniform` function in `search.py` #704

24. Genetic Algorithm GUI example `gui/genetic_algorithm.py` #702

25. Grid-MDP Editor GUI `gui/grid_mdp.py` #719 and #734

26. Eight-Puzzle GUI `gui/eight_puzzle.py` #861

27. TSP algorithm selection menu (GUI) `gui/tsp.py` #706

28. Minor bugfixes #715, #771, #790, #794, #801, #832, #864 and #867

29. Solved issues: #696, #709, #710, #711, #713, #766, #785, #792, #824 and #826

30. Helped users solve issues: #786, #800, #819, #824, #826, #865

31. Gave feedback and suggestions to users: #705, #735, #814, #830, #856, #858, #860, #855, #846, #845, #842, #837, #835, #827, #825, #815, #810, #808, #791, #781, #776, #742 and #737

PAST PROJECTS

1. **CNNs learn to drive**: An ensemble of convolutional neural networks were trained on an image dataset to drive a car on a thin track. The environment was built in Unity-C#. The dataset was generated from a screencast of the game being played by actual humans. The dataset was appropriately preprocessed and fed into an ensemble of convolutional neural networks which was then trained to control a car (using virtual keypresses) on the track using the real time screencast as test data. Source code links: environment, project.

2. **Island project**: This project was done as part of a Hackathon last year. It consisted of an environment with a track on an island. The track was marked with waypoints. Cars with random physical properties were run on it and optimized using modified genetic algorithms in an attempt to find the values that minimize the track-time. There were 18 variables to optimize over, including max steer-angle, max braking torque, throttle delay, etc. Several generations were run and the data was fed into a neural network and trained to predict the track time given the values of the 18 variables. A data visualization tool was also built. A lot of interesting results were found. Repository links for the environment and the network model.

3. **BoxCar2D**: This is a computation intelligence car evolution project, a replica of BoxCar2D. A 'car' is generated by drawing a polygon using eight randomly selected points as vertices. Each vertex has a probability of spawning a wheel of random size. A population of 20 'cars' are run on a track and their characteristics are optimized using a genetic algorithm to get the optimum model. The environment was completely built on Processing, a Java-based environment and the Box2D physics engine was used. Directory

4. **Genetic Algorithms**: I have numerous smaller projects based on Genetic Algorithms, the latest one involving NEAT, where we try to find the best neural network architecture for a particular problem using genetic algorithms. Details are in this repository.

5. **Deep Learning**: Deep Learning interests me the most. Following are the topics and repository links of some of my work in this field.
   1 Analysis of linear models using `scikit-learn`. notebook and notebook
   2a MNIST dataset image classification using CNNs. program
   2b MNIST dataset image classification using RNNs. notebook
   2c MNIST dataset analysis using the `keras` api. notebook
   2d Effect of Adversarial Noise on the MNIST dataset. notebook
   3a An AI to balance a pole on a cart (the `CartPole-v0` environment of OpenAI-gym) using densely connected neural networks. notebook
   3b An AI for the `CartPole-v0` environment using vanilla policy-gradients. notebook and notebook
   4 Statistical Learning (statistical inference using machine learning) using `scikit-learn`. notebook

5 Data analysis on the Iris dataset. notebook
6 Clustering and compression analysis using `scikit-learn`. notebook
7 KMeans clustering analysis. notebook
9 Data analysis on the titanic-dataset using clustering methods. notebook, algorithm
10 Support Vector Machines in `numpy`. program
11 Effect of Adversarial Noise on the `Inception-v3` model. notebook
12 Principal component analysis. notebook
13 Python implementation of Geoffrey Hinton's Capsule Network concept using `tensorflow`. notebook
14 Sentiment Analysis on IMDb movies dataset. notebook
15 Music generation from MIDI sequences using RNN (LSTM) networks. directory, model
16 Transfer Learning with `tensorflow` on the CIFAR-10 dataset using the `inception-v3` model. notebook and another related notebook.
17 Style transfer on images with `keras` on `tensorflow` using the VGG16 model. notebook
18 AI for the Halite-2 competition. directory and directory.

6. **Game development**: I spent a few months learning to make games on the Unity engine using C#. Simple examples of my work are included in this repository. Unity was also used to create the environment for the "Island project" and the track for the "CNNs learn to drive" project. The source for these environments and a few other simpler games are included in this repository. I also have experience with Python's game engine `pygame` and the physics engine `pymunk`. Directory.

7. **Natural Language Understanding**: This was a collaborative NLU project (written in C++) based on a problem statement in the latest Hackathon that I participated in, where we were expected to classify input sentences into their corresponding 'Grammars' and to find the boundaries for placeholders and open-phrases. Given an input string, we were supposed to find out what the user wants and call the respective function with the correct arguments as part of a larger NLU project. Implementational details and the source code are in this repository.

8. **Diabetic Retinopathy Detection**: Diabetic retinopathy is the leading cause of blindness in the working-age population. This is an ongoing collaborative project aimed at detecting the presence and severity of this disease using retina scans. This project is incomplete and still in the preliminary phase at the time of writing. Repository

9. **Web development**: This project was aimed at developing a website for a contemporary Indian artist. Repository.

10. **JavaScript applets**: I sometimes make simple JavaScript applets in my leisure time, some of which are deployed as webapps. This repository contains more details.

11. **Processing sketches**: I sometimes make simple Processing (a Java-based environment) applications. This repository and this repository contain some examples.

12. **Android Application Development**: This was a college-management application developed by me and another contributor during the annual Hackathon in my college last year. Though the app hasn't been deployed, it is functionally complete. Repository for android app, Repository for web app.

ABOUT ME

I am an undergraduate student pursuing Bachelor of Technology in Electronics and Communication at the Indian Institute of Technology (ISM), Dhanbad. My main areas of interest are Artificial Intelligence, Deep Learning, and Calculus. I am fluent in Python, C++ and JavaScript and spend all my free time exploring the different domains of Artificial Intelligence. I have also completed several MOOCs on Machine Learning and Deep Learning. The certificates for the same are linked here.
It's been quite a long time since I am contributing to this project and in my opinion, I have made some sizeable contributions to it. I have learnt a lot from the book, the codebase and my fellow contributors. It would be a great opportunity for me if I am selected to continue working on this project as a part of GSoC this year. I plan to be a long-time contributor and will support the organization and new contributors even after GSoC and when AIMA-4e is eventually released. I would love to help this project under a mentor.