

Introduction to AI and Autonomous Systems

Lab Assignments



December 11, 2020

Instructions

To deepen your understanding of the two areas of reasoning and learning, we have designed two lab assignments in the form of experiments. The details are below. Please hand in a short written report (2-3 pages) for each of the two labs stating what techniques you used, what experiments you did, the results you got, a discussion of these results, and the conclusions you draw.

You may complete the lab assignment either individually or in pairs. The main purpose is not to write a lot of code, but rather to get some understanding both of what it takes to actually realize some AI methods and to make modifications to them in order to be able to experiment with them. It is therefore allowed to use existing code as long as it is clear where it is from and what you have contributed.

The deadline is **January 11**.

Software

The recommended software is Python3 and Anaconda. It can easily be installed on all major operating systems. The following is good instruction: <https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/>

Lab 1 – Reasoning

The purpose of this lab is to explore how to make decisions in a simulated world playing Pac-man, see <http://ai.berkeley.edu/search.html>. Since the original code is for Python2, you probably want to use the Python3 version for example <https://github.com/jspacco/pac3man>. You can download the files here <https://github.com/jspacco/pac3man/archive/master.zip>, the Pac-man files are in the directory search.

You are expected to implement at least two different methods and compare them. It is sufficient to complete the first three questions. Suitable algorithms to implement and evaluate are different versions of Breadth-First Search, Depth-First Search, A*, and Monte-Carlo Tree Search. Suitable things to measure are for example average score and nodes expanded.

Lab 2 – Learning

The purpose of this lab is to explore learning methods. You are expected to implement at least two different learning methods and do at least three experiments with these. In each experiment you should vary at least one variable and measure at least one variable. Suitable methods to implement and evaluate are different versions of neural networks, regression, support vector machines, and Gaussian Processes.

- Time series prediction. Train at least two models to predict the next k steps of a known function f given n samples from the function. Try different values of k and n as well as different functions and sampling strategies. Good functions are polynomials and trigonometric functions.
- Image classification. Train at least two models to classify MNIST digits or CIFAR images and compare the accuracy. You can vary the training time and the number of examples.
- Reinforcement learning. Apply reinforcement learning to Pac-man from lab 1.

A good starting point, especially if you are a beginner, is the following step-by-step tutorial for implementing a convolutional neural network for learning MNIST digits: <https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network>. It works out of the box on a decent laptop taking a few minutes to train. It has concrete suggestions for experiments to do. The simplest is probably to study the influence of the learning rate or the model depth.

For reinforcement learning in a simple grid world, the following is a good starting point that can be modified or extended: <https://towardsdatascience.com/training-an-agent-to-b>. You can also start from the Berkeley AI Pac-man project http://ai.berkeley.edu/project_overview.html.