Introduction to Artificial Intelligence

Fall 2021

Prof. Gilles Louppe g.louppe@uliege.be



Goals and philosophy

Thorough and detailed

- Understand the landscape of artificial intelligence.
- Be able to write from scratch, debug and run (some) Al algorithms.

Well established algorithms and state-of-the-art

- Well-established algorithms for building intelligent agents.
- Introduction to materials new from research (\leq 5 years old).
- Understand some of the open questions and challenges in the field.

Practical

• Fun and challenging course projects.

Us

This course is given by:

- Theoretical lectures: Gilles Louppe
- Exercise sessions: François Rozet
- Programming projects: Arnaud Delaunoy, Gaspard Lambrechts

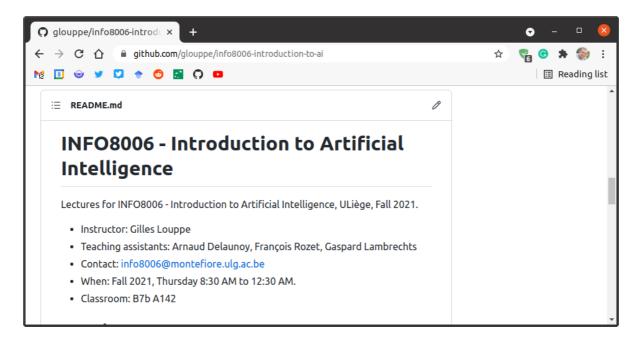
Feel free to contact us at info8006@montefiore.ulg.ac.be for help.



Materials

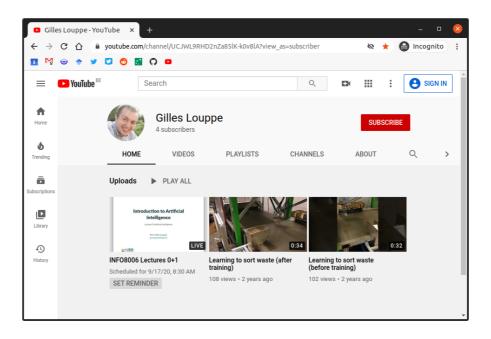
The schedule and slides are available at github.com/glouppe/info8006-introduction-to-ai.

- In HTML and in PDFs.
- Posted/updated online the day before the lesson (hopefully).
- Minor improvements/fixes from previous years.

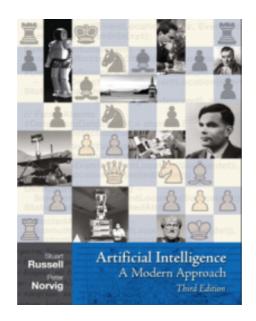


Videos

Videos from Fall 2020 are available at https://youtube.com/playlist? list=PLLqXZ_E-UXlybvRU7vgaYMTbxZdT73ZFD.



Textbook



The core content of this course is based on the following textbook:

Stuart Russel, Peter Norvig. "Artificial Intelligence: A Modern Approach", Third Edition, Global Edition.

This textbook is recommended. It covers both the theory and the exercises.

CS188

- Some lessons and materials are partially adapted from "CS188 Introduction to Artificial Intelligence", from UC Berkeley.
- Cartoons that you will see in those slides were all originally made for CS188.



Projects

Reading assignment

Read a major scientific paper in Artificial Intelligence. (Paper to be announced later.)

ARTICLE

Mastering the game of Go with deep neural networks and tree search

Julian Schrittwieser¹, Joannis Antonoglou¹, Veda Panneershelvam¹, Marc Lanctot¹, Sander Dieleman¹, Dominik Grewe¹, John Nham², Nal Kalchbrenner¹, Ilya Sutskever², Timothy Lillicrap¹, Madeleine Leach¹, Koray Kavukcuoglu¹, Thore Graepel1 & Demis Hassabis

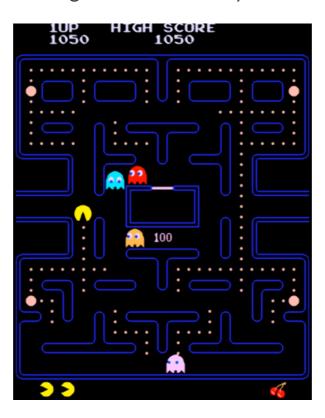
The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to compate Go that tasks "value networks' to evaluate board positions and policy networks' to select naives. These deep to the compater Go that tasks "value networks' to evaluate board positions and policy networks" to select naives. These deep learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the- art Monte Carlo free search programs that similate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, or program Alpháco ochieved a 93% witning intel against other Go porgrams, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

All games of perfect information have an optimal value function, v (i), which determines the octations of the game, from every based for a flower of the object of the octation of the game, from every based on a flower or thin the object of by extraction by comparing our septimes a manifest mean in a section of the presence of the present of the presence of the presence of the present of the presentation of the presentation of the presentation of the presenta

Scrabble," and lowest ensistent level play in Ga¹⁰
Monte Carlo tree search (MCTS)¹¹ uses Monte Carlo rollouts
to estimate the value of each state in a search tree. As more simutalians are executed, the search tree gooss larger and the result as lates to the search as a long soon larger and the result as lates to the search as a long soon larger and the result as lates to the search as a long soon larger and the result as lates and the search as a long search as long search as a long search as a long search as long

Programming projects

Implement an intelligent agent for playing Pacman. The project will be divided into three parts, with increasing levels of difficulty.



Evaluation

- Written exam (60%)
 - Short questions on the reading assignment will be part of the exam.
- Programming projects (40%)
 - Project 0: 0%
 - Project 1: 20%
 - Project 2: 20%
 - Programming projects are mandatory for presenting the exam.

Let's start!