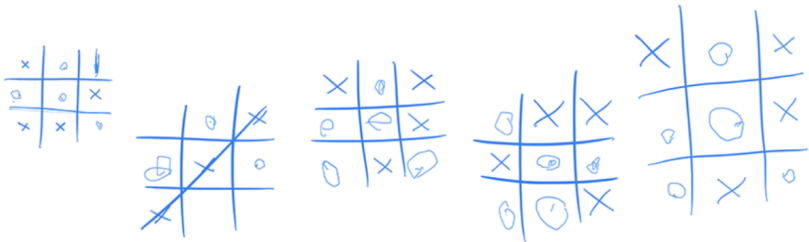
CS-C1000 – Introduction to Artificial Intelligence

Computer Assignment C

William Wilkinson

March 30, 2021

Computer Exercise C:
Reinforcement learning for Tic-Tac-Toe

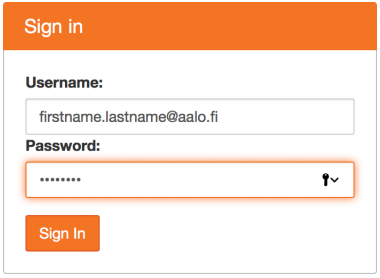


Train your own game-playing AI.

Log into JupyterHub (a quick reminder)

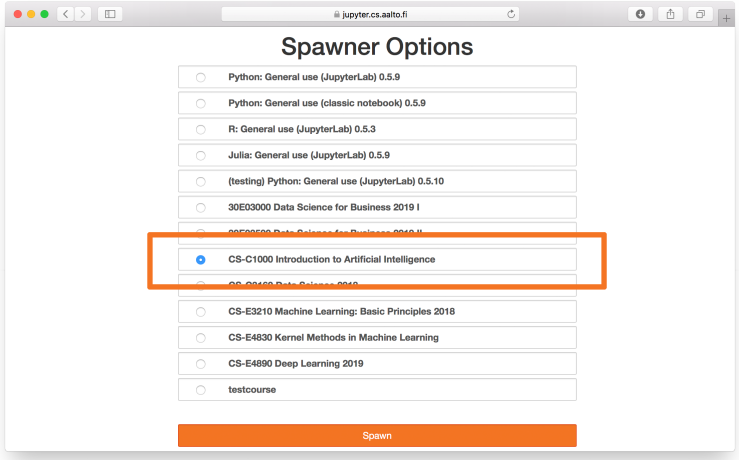
Log in to:

`https://jupyter.cs.aalto.fi`

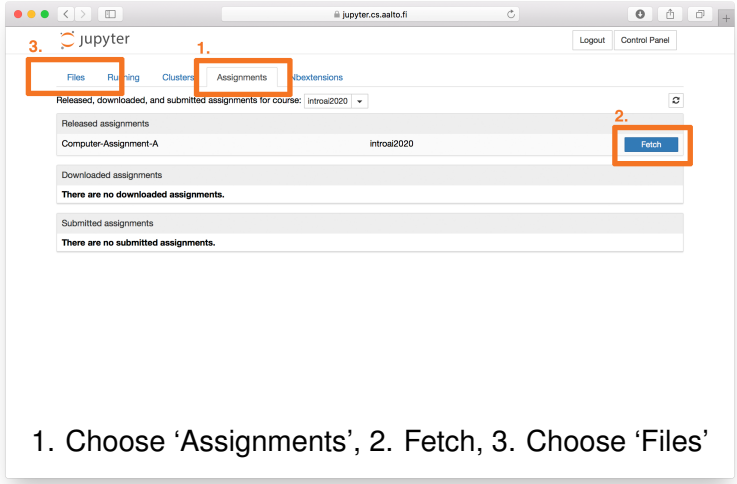


Using your Aalto account.

Choose the course

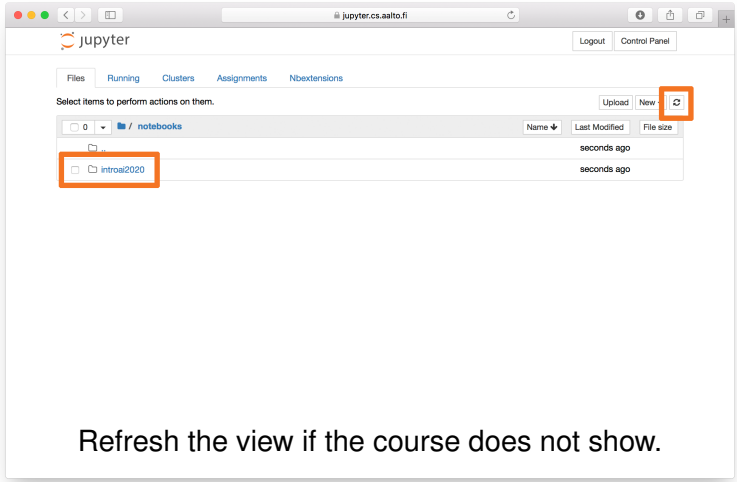


Fetch the exercise (well, C this time, not A!)



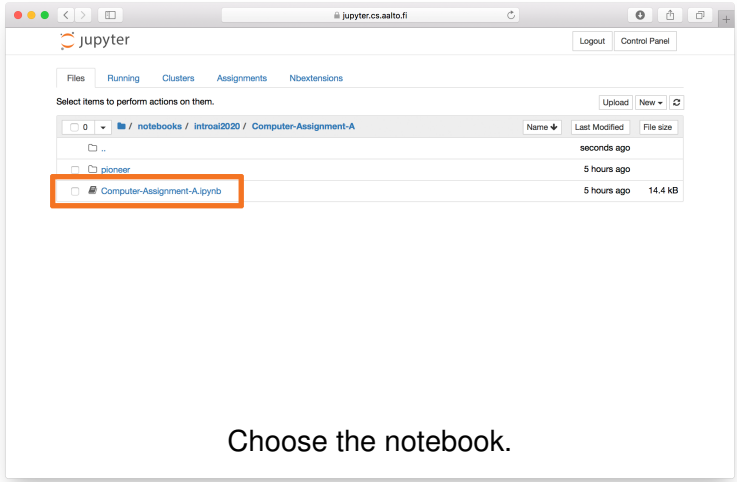
1. Choose ‘Assignments’, 2. Fetch, 3. Choose ‘Files’

Find the notebook



Refresh the view if the course does not show.

Fire it up (again, C this time)



Choose the notebook.

Ready to run (a cell at a time)

jupyter Computer-Assignment-C (unsaved changes)

Logout

File Edit View Insert Cell Kernel Help

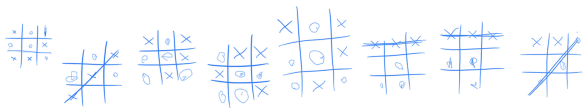
Not Trusted | Python 3

Run | Markdown

Computer Assignment C

Teaching an Artificial Intelligence to play Tic-Tac-Toe

For developing systems that can make informed decisions, board games provide good benchmarks. Advances in AI methodology (not just the speed of computers!) has brought us to a point where game-playing AIs can beat human players not only in **Chess** but also in **Shogi** and **Go** [1]. However, this is narrow artificial intelligence, methods trained for a specific purpose and that do not directly generalize to other tasks. For example, the example code below will not learn to drive a car or cook dinner however much you show it driving or cooking.



Reinforcement learning for game play

In this exercise, you get to train a game-playing AI from scratch for the classic game of Tic-Tac-Toe (also known as Noughts and Crosses) [2]. We consider the simplest version of the game, where only a `3x3` grid is used. The AI will know the hard-coded rules of the game, but simply knowing the rules does not make the AI great at playing. One option would be to show the AI a lot of human-played games as *training data*, but instead we employ a method called *reinforcement learning* (RL). We will put the RL agent to play games agaist itself such that it learns how to play the game efficiently.

We will consider Q-learning, a model-free reinforcement learning (RL) algorithm. The goal of Q-learning is to learn a policy (what to do), which tells the game-palying agent what action to take under what circumstances (depending on where the opponent puts their mark). It can learn purely by self-play once it has been given the rules of the game. The agent will receive a reward of `+1` if it wins and `-1` if it loses a game. We use a modified implementation following [3], and all code needed for this task is presented below.

Tasks (read the notebook for exact instructions)

- Set up an agent.
- Try the agent by playing against it yourself.
- Train the agent by self-play.
- Try the agent by playing against it yourself.
- Continue training and playing.

Tasks (read the notebook for exact instructions)

- Set up an agent.
- Try the agent by playing against it yourself.
- Train the agent by self-play.
- Try the agent by playing against it yourself.
- Continue training and playing.

Tasks (read the notebook for exact instructions)

- Set up an agent.
- Try the agent by playing against it yourself.
- Train the agent by self-play.
- Try the agent by playing against it yourself.
- Continue training and playing.

Tasks (read the notebook for exact instructions)

- Set up an agent.
- Try the agent by playing against it yourself.
- Train the agent by self-play.
- Try the agent by playing against it yourself.
- Continue training and playing.

Tasks (read the notebook for exact instructions)

- Set up an agent.
- Try the agent by playing against it yourself.
- Train the agent by self-play.
- Try the agent by playing against it yourself.
- Continue training and playing.

Tasks (read the notebook for exact instructions)

- Set up an agent.
- Try the agent by playing against it yourself.
- Train the agent by self-play.
- Try the agent by playing against it yourself.
- Continue training and playing.

How to get points?

- To get the points for this week's exercise, answer the questions in MyCourses related to this Computer Exercise C.