IDATT2502 Final Project
Weekly Summaries

Erling Sletta

November 2021

Week 1 (01.11 - 07.11)

While the project formally started on the 19th of October, I was not able to properly get started with actual work before the beginning of November due to coursework in other courses among other things. The time was not spent idly, however, as I was carefully considering each project idea, so that I could get started right away as soon as I had the time to.

This first week was spent mostly researching different environments and then trying to get it running. Originally, I would've liked to create one from scratch, but doing the project alone I figured I might not have enough time to so do.

Chose the environment found here: https://github.com/Kautenja/gym-tetris, and started experimenting with it. Mapped out the action space:

0 - NOOP	No Operation, agent idles
1 - A	Agent rotates piece clockwise
2 - B	Agent rotates piece counterclockwise
3 - Right	Agent moves piece to the right
4 - Left	Agent moves piece to the left
5 - Down	Agent moves the piece down

Looked back on the 8th exercise and attempted improving and testing features on this. Uncovered a few flaws with my solution and attempted to fix them, one being that the exploration rate was broken, so the agent always explored. Started looking into how to save and load models with PyTorch.

Week 2 (08.11 - 14.11)

Kept trying to improve on the DQN agent used in the 8th exercise, found a few more flaws to bed fixed. Looked up relevant material for creating Tetris AI. Attempted to fit my CartPole model to the Tetris environment, but struggled somewhat due to lacking documentation for the latter.

Could not figure out what the state represented, and instead started using the info dict that was returned. After a lot of trial and error, I managed to get a model that ran, but it didn't seem to be learning anything worthwhile.

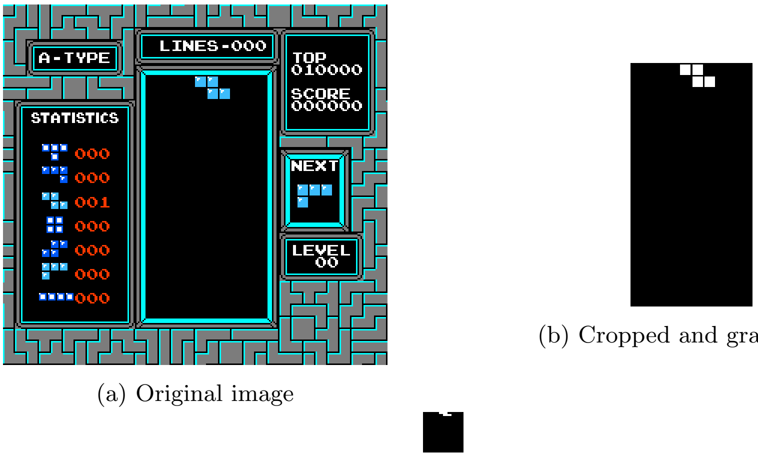
Week 3 (15.11 - 21.11)

Kept experimenting with the environment and eventually figured out that the state represented the environments raw pixel data. A 240 x 256 window with RGB color.

Met with T.A., who recommended switching to a different environment due to raw pixel data not being too convenient to work with.

Wanted to experiment a bit longer with my current environment before moving on, so conducted a few more tests and let the model run for longer periods to see what happened. Unfortunately the code crashed when the agent managed to get it's first line clear. Spent some time troubleshooting this issue. Attempted rewarding the agent for staying alive longer, which seemed to give some results.

Found a useful article about playing Atari games in an OpenAI gym environment that returns raw pixel data. In the article, image pre-processing was used simplify the raw pixel data into much more manageable states. After implementing these methods to my own model, I let it ran for a while to see if it had changed anything. Images were at this point in time processed as such:



(a) Original image

(b) Cropped and gray-scaled

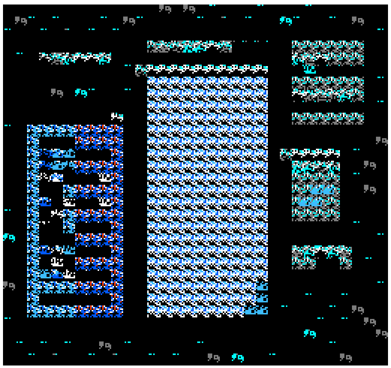
(c) Resized

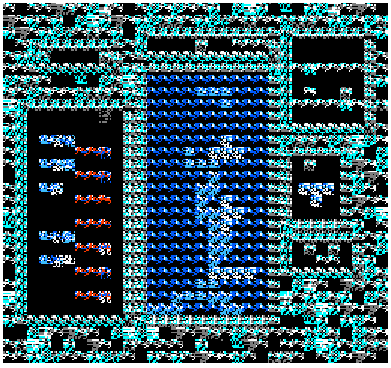
Kept tweaking the model for a long time. Experimented with different reward schemes etc. to see how it would affect the agent. For example, agent becomes very hyper when rewarded for every action made. Observed that over time, the agent started to have some very short episodes. Typically they lasted around 5000 to 10000 steps, but it was consistently ending the game within 1000, with the all time low being 428. Assumed that maybe it didn't know that the negative reward given at game over was bad, since it never received any positive rewards anyways?

Met with T.A. again, discussed the complexity of the environment etc., and possible alternatives. Mentioned genetic algorithms and heuristics, and decided too apply some finishing touches on the DQN models before moving on to these.

Previously, images were processed to be 20x20 images. This could possibly cause lossy input, so the processing was revamped so that rectangular true-to-original images were created. Experimented with different state shapes, how many sequential frames were sent at a time to be exact.

Encountered some bugs with the environment, where it would turn into one of the following:





Week 4 (22.11 - 26.11)

The DQN model had the advantage of taking long times to train, and the bugs above as well as the time used per episode in the chosen environment gave me little hope of achieving an agent trained to the point where it could perform reasonably well. Nonetheless, the model did seem to be somewhat successful, as some signs of intelligence and patterns could be seen.

Decided to move onto heuristics and genetic algorithms to see how they would perform in comparison while leaving the DQN model training to generate graphs.

Started implementing analytical methods to be used for heuristics, and started looking into genetic algorithms once they were completed. Got an agent running using random heuristics, and it was already looking pretty promising.

Met with T.A., got some pointers on the implementation of the genetic algorithm.

Finished up implementing the genetic algorithm, and referenced another project my T.A. introduced to me to implement parallellization since I don't have much experience in the field.

Started focusing on finishing the final report while running the genetic algorithm in the background to generate graphs. Genetic algorithm trained **very** slowly. Also found and fixed some bugs and implemented some optimizations while working on the final report.