REINFORCEMENT LEARNING

Connect4

Max de Goede & Rik van Spreuwel

Fontys Hoge school ICT
Data Science Minor

Introduction

Max and Rik worked together for the Reinforcement Learning (RL) exercise. They chose to work on a different environment than the standard Gym environment.

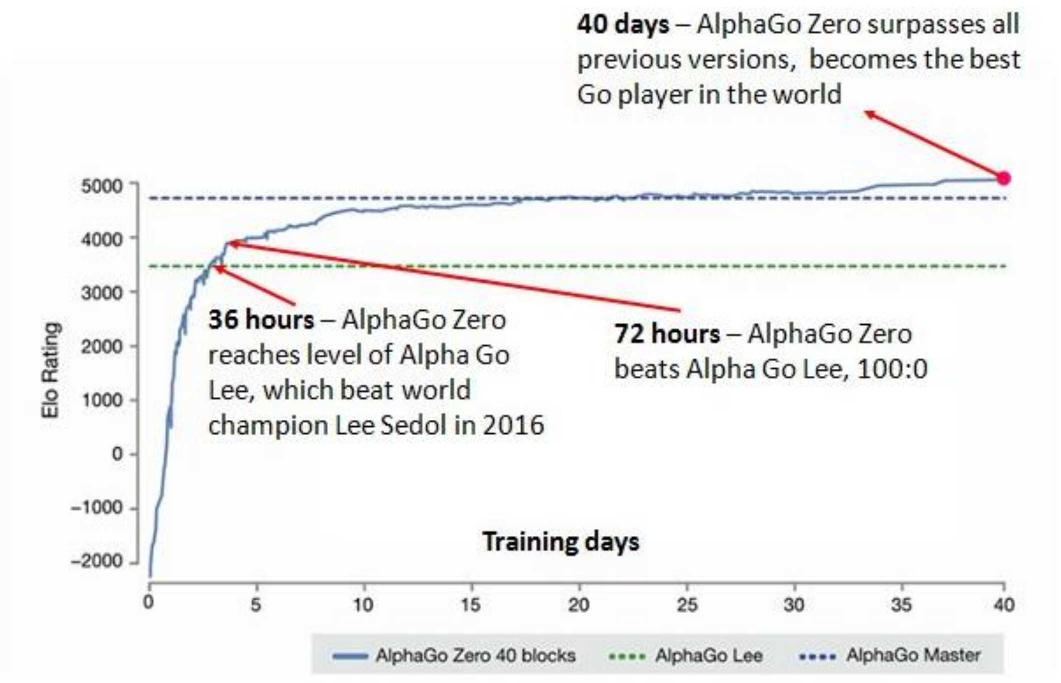
Used Case

The case is letting a network train and play the game connect 4 (In dutch known as “Vier op een rij”) similarly to the way Deepmind’s AlphaZero learned to play chess and go, through self-play. The inspiration came from the following tutorial by David Foster. The code used came from his Git repository.

Alpha Zero

The Alpha zero algorithm a neural network that takes the current game state and decides what action to take. The algorithm starts with a blank slate, playing completely random moves. It learns from its games and adjusts the weights of the neural network, making it more likely to make better moves. After it’s trained, it plays by calculating what action has a highest success rate (in other words, the moves with the highest chance to win).

Something really interesting about Alpha zero when used in chess was the moves it made. Chess has had machines playing games for a long time and what moves were best to make were not always keen on the eyes of a human. But the moves alpha zero was making were more relatable to humans than the moves stockfish would make. This was groundbreaking, it was nearly impossible for humans to play against stockfish and learn anything since the moves it made did not make a lot of sense. Making sense of the moves is something that alpha zero did better than stockfish.


40 days – AlphaGo Zero surpasses all previous versions, becomes the best Go player in the world

36 hours – AlphaGo Zero reaches level of Alpha Go Lee, which beat world champion Lee Sedol in 2016

72 hours – AlphaGo Zero beats Alpha Go Lee, 100:0

A graph from **‘Mastering the Game of Go without Human Knowledge’**

Connect 4

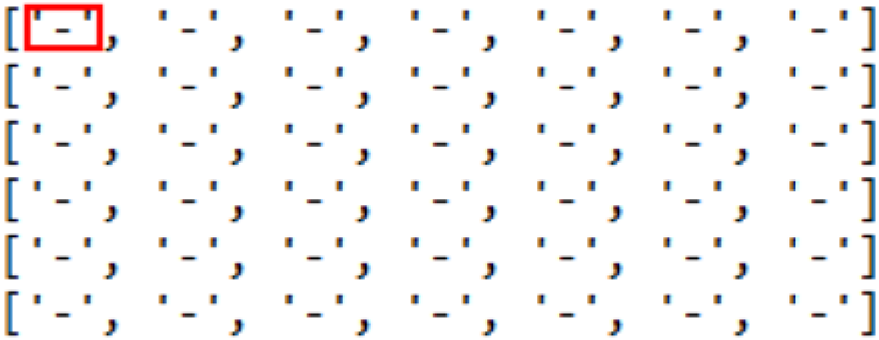
It would be impossible to run chess since it has a possible *"7728772977965919677164873487685453137329736522, or about 7.7∗10457.7∗1045, based on a complicated program by John Tromp"*

This would be impossible to run without a supercomputer. The game of connect4 seems like a better fit for this since there are a total of 4.531.219.092 positions. This number is at least comprehensible for humans. And training with such numbers is more of a feasible task.

For Connect4 an environment had to be built where the game could be played and rules had to be created so the network knew when it had won/lost and where to place the move.

What does the connect 4 environment look like?

In the figure below you can see a visualization of the environment. The environment consists of a grid containing cells, each cell having a value. This value is used to indicate where the move is going to be placed (Grid numbers 1-41).



```
['-', '-', '-', '-', '-', '-', '-']
['-', '-', '-', '-', '-', '-', '-']
['-', '-', '-', '-', '-', '-', '-']
['-', '-', '-', '-', '-', '-', '-']
['-', '-', '-', '-', '-', '-', '-']
['-', '-', '-', '-', '-', '-', '-']
```

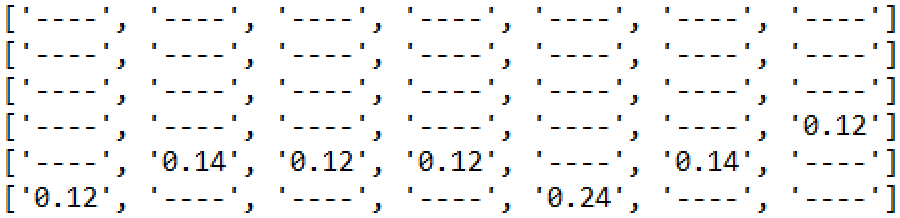
The game consists of two players player 1 and player 2 (X&O). And the goal is to connect 4 in a row.

If you want a small video with a demo on what it looks like to play the machine click the link below. There is a small video image 😊.

https://gyazo.com/0efc36bf7dacbd9af1f62d2c85d175c3

How does the network decide to play it's next move?

It looks at percentage on where it could best play a move to have the highest chance of winning and places it's move there.



```
['----', '----', '----', '----', '----', '----', '----']
['----', '----', '----', '----', '----', '----', '----']
['----', '----', '----', '----', '----', '----', '----']
['----', '----', '----', '----', '----', '----', '0.12']
['----', '0.14', '0.12', '0.12', '----', '0.14', '----']
['0.12', '----', '----', '----', '0.24', '----', '----']
```

Results

After training the algorithm for half a day, we were able to get a network that you were able to play against the network, these were the results

Rik played 2 games against the network and won both.

Max played 5 games against the network and lost once.

A third party also played against this network (Tobias Leenders) he played 3 games against the robot, lost 2 and won once by exploiting a bug in the system.

Conclusion

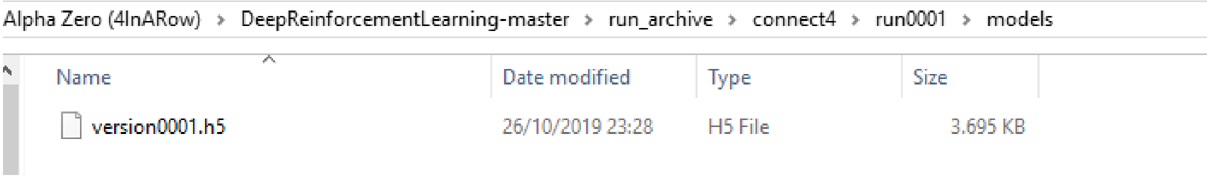
In conclusion, the network performed much better than expected. It played well against humans with only half a day of training. The best part about this exercise was to see how we were able to create and train a network you could play against and see how well it performed and how it made the decisions of what moves to make.

[OUR GIT]

File for weights download, if you want to play against the AI download the weights and refer it to the run file.

https://drive.google.com/file/d/1LXTTxAfwjCyTqSRiqqQPRh7cWVTN3MwH/view?usp=sharing

Special directory is needed for this,



Alpha Zero (4InARow) › DeepReinforcementLearning-master › run_archive › connect4 › run0001 › models

Name	Date modified	Type	Size
version0001.h5	26/10/2019 23:28	H5 File	3.695 KB