Project Proposal

MA2823

Benoît Sepe
Centrale Paris
benoit.sepe@student.ecp.fr

Alexandre Huet
Centrale Paris
alexandre.huet@student.ecp.fr

Badr Youbi Idrissi
Centrale Paris
badr.youbi-idrissi@student.ecp.fr

PROBLEM TO SOLVE

Given a simple 2 platformer game, **Use** Deep reinforcement learning (Q- learning) **To** maximize the score.

MOTIVATION AND PROBLEM DEFINITION

Classical game AIs focus on implementing multiple strategies to win a game. The environment is "explained" to the AI through multiple developer defined rules. Therefore, it only aims to perfect already known strategies and make them faster and more efficient. It usually takes the form of "IF … HAPPENS, THEN DO …". This works fine for simple environments but can be very challenging and even impossible in other cases where it is too unpredictable and the rules too complex. With Deep Reinforcement learning we delegate the discovery of the environment and efficient strategies to the AI. We only present it with an input that represents the state of the current environment and the set of possible actions. With a system of "rewards", it learns to take the appropriate actions at the appropriate time.

This kind of AI can be used in real environments too. For example, we can use it in a robot to learn how to still stable in diverse conditions – like robots of Boston Dynamics. Another use is learning an optimal sequence of decisions in an industrial environment.

METHODOLOGY

Firstly, we need to acquire a good understanding of deep reinforcement learning. We've identified an interesting implementation of it called Deep Q-Learning. After nailing down the details of its implementation we will clone a simple, open source game in python and modify it to have multiple instances running efficiently for the algorithm to train. This will require parallel processing. Tensor Flow along with Keras is implemented on the GPU and allows for efficient training. The basic model we have in mind currently uses a Convolutional Neural Network to analyze the game frames and decide which action to take (This is close to a classification CNN in structure). The objective function is the Q-Value which tries to consider immediate rewards, distant rewards and delayed rewards. If we achieve a performant AI (and still have time) we wish to use Feature Visualization techniques on the CNN to see what It really learned. Another possibility is to see how we could modify the objective function to have a more human-like player (Penalize spazzy behavior and too fast reaction times).

This goes in the continuity of all the recent efforts (and successes!) of DeepMind. Indeed, they implemented multiple Deep Reinforcement AIs such as AlphaGo and the different Atari game AIs.

EVALUATION

Evaluation in this context is easier since it is how we train the model in the first place. That said, the goal is to have a performant AI for the whole game and not just a specific level of the game. So wherever possible we will try to introduce randomness as to make it more challenging and avoid “overfitting” a specific map of the game by just remembering the actions.

REFERENCES

[1] "A Beginner's Guide to Deep Reinforcement Learning", Skymind, 2018. [Online]. Available: https://skymind.ai/wiki/deep-reinforcement-learning. [Accessed: 14- Oct- 2018].

[2] "A Painless Q-Learning Tutorial", Mnemstudio.org, 2018. [Online]. Available: http://mnemstudio.org/path-finding-q-learning-tutorial.htm. [Accessed: 14- Oct- 2018].

[3] "Case of application with AlphaGo", Biostat.wisc.edu, 2018. [Online]. Available: https://www.biostat.wisc.edu/~craven/cs760/lectures/AlphaZero.pdf. [Accessed: 14- Oct- 2018].

[4] A. Mkrtchian, J. Aylward, P. Dayan, J. Roiser and O. Robinson, "Modeling Avoidance in Mood and Anxiety Disorders Using Reinforcement Learning", Biological Psychiatry, vol. 82, no. 7, pp. 532-539, 2017.

[5] "What are some good tutorials on reinforcement learning? - Quora", Quora.com, 2018. [Online]. Available: https://www.quora.com/What-are-some-good-tutorials-on-reinforcement-learning. [Accessed: 14- Oct- 2018].

[6] "Course of Toronto", Cs.toronto.edu, 2018. [Online]. Available: http://www.cs.toronto.edu/~zemel/documents/411/rltutorial.pdf. [Accessed: 14- Oct- 2018].

[7] M. Wiering and M. Otterlo, Reinforcement Learning. Berlin: Springer Berlin, 2014.