Project proposal  
04-05-2017  
Natural Computing

*The Screeching Owls*

Jordi Beernink, Roel Bouman and Jeffrey Luppes

# Introduction

In recent years, deep learning has become famous in various domains, especially when it comes to mastering human level control policies[[1]](#footnote-1)[[2]](#footnote-2). Such as object recognition with visual attention, high-dimensional robot control and solving physics-based control problems[[3]](#footnote-3). These kind of situations are mostly handled by a variation of deep learning, called reinforcement learning[[4]](#footnote-4). Which is a combination of deep learning with Q-learning, with the most important example to let AI play Atari games and the first victory of AlphaGo over an human champion on Go. In this experiment this method will be explored in a 3D environment[[5]](#footnote-5).

This 3D environment has been made available by Visual Doom (ViZDoom). One of the AI competitions created a platform based on Doom, a first person shooting game[[6]](#footnote-6). The AI players on this platform can only obtain visual input and some variables. This opens new ways to explore AI methods in machine visual learning and deep reinforcement learning.

# Hypothesis

# Methods

In this experiment the following methods will be applied:

* Neural Network
  + Recurrent
  + Convolutional
* Reinforcement/Q learning
* Deep Q Learning

The encompassing method in this experiment will be applying variations of the neural network algorithm.

Neural network is a system of hardware and/or software patterned after the operation of neurons in the human brain. These neurons are connected in layers, and signals travel from the first (input) layer to the last (output) layer[[7]](#footnote-7).

Convolutional neural network is a feed-forward neural network in which the connectivity pattern between its neurons is inspired by the organization of a visual cortex. These 'visualization' neurons respond to stimuli in the receptive field, which is a set of neurons partially overlapping with each other so that they can tile the visual field. The response of an individual neuro to stimuli within its receptive field can be approximated mathematically by a convolution operation[[8]](#footnote-8).

Recurrent neural network is a variation of the neural network algorithm, where the connections between units form a directed cycle. This creates an internal state of network which allows it to exhibit dynamic temporal behavior[[9]](#footnote-9). Unlike feedforward neural networks, recurrent networks use their internal memory to process arbitrary sequences of inputs.

Q-learning, also known as reinforcement learning, is a model-free reinforcement learning technique, which is a method to teach software agents to take actions in an environment so as to maximize some notion of cumulative reward[[10]](#footnote-10). Q-learning can be used to find an optimal action-selection policy for any given (finite) decision process. It works by learning an action-value function that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy thereafter. Where a policy is a rule that the agent follows in selecting actions given the state it is in. When such a function is learned, the optimal policy can be constructed by selecting the action with the highest value in each state[[11]](#footnote-11).

A variation on this is called deep reinforcement learning, which uses the multiple layers of representations of deep learning with reinforcement learning. This combinations allows the agent to control a system based only on visual inputs, using the neural network to extract relevant features from images[[12]](#footnote-12).

# Reasoning

Due to the structure of the experiment. Different aspects of the algorithm, neural network, are necessary. Firstly the different images shown in the Doom platform need to be analyzed and the different objects must be able to be recognized by the agent. So that right action can be picked to get the maximum value of reward.

Because of this the convolutional neural network will be used, to analyze the different objects in the frames of the platform. But due to the fact that the different visual situations need to be remembered by the agent. This is achieved with the recurrent neural network. With this the agent is able to perceive the different visual situations and be able to remember these perceptive fields.

The last of the methods is deep reinforced learning. Which helps the agent with making its decisions and performing his actions in the 3D environment. Other alternatives to this would be policy gradients. This is a variation of the reinforcement learning technique that relies upon optimizing parametrized policies with respect to the expected return by gradient sufficient. The reason why this method will not be used as because it does not follow the traditional framework from reinforcement learning[[13]](#footnote-13). Due to this, the deep reinforced learning will be used. Because of the lack of experience in this field.

# Alternatives

# Bibliography

Bai, S., & Chen, C. (2016). *Deep Reinforcement Learning on the Doom Platform.* Carnegie Mellon University, Department of ECE. Pittsburgh: Carnegie Mellon University. Retrieved 3 May, 2017, from https://dl.dropboxusercontent.com/content\_link/VG1b5DUYmVHJzJJRtxpwy9pV3vIU9gx4PtBChVXsgd2CgWnabIv4Qcz6QxAsrkIp/file

Berstekas, D. P. (1960). *Dynamic Programming and Optimal Control* (425-430 ed., Vol. 2). Massachusetts: Massachusetts Institute of Technology. Retrieved May 3, 2017, from https://en.wikipedia.org/wiki/Markov\_decision\_process

*Convolutional neural network*. (2010). Retrieved May 3, 2017, from Deep Learning: http://deeplearning.net/tutorial/lenet.html

*Introduction VizDoom*. (2016). Retrieved 3 May, 2017, from VizDoom: http://vizdoom.cs.put.edu.pl/

Lample, G., & Chaplot, D. S. (2016). *Playing FPS Games with Deep Reinforcement Learning.* Carnegie Mellon University, School of Computer Science. Pittsburgh: Carnegie Mellon University. Retrieved 3 May, 2017, from https://dl.dropboxusercontent.com/content\_link/SV6CUvSg7bblDhdMVm4cd01dI53FZLPNlaDDCfgykzst0FnQz2j5M5gIYI7Z5NkY/file

Park, H., & Kim, K.-J. (2016). *Deep Q-Learning using Redundant Outputs in Visual Doom.* Sejong University, Department of Computer Science and Engineering . Seoul: Sejong University. Retrieved May 3, 2017, from https://dl.dropboxusercontent.com/content\_link/hEIcMtu1J6EpxNjFPVYeBZyT5sc17uYECR1sclShBuehbFFfx6sZgLxFkghYqkOr/file

Peters, J. (2010). *Policy gradient methods*. Retrieved May 3, 2017, from Scholarpedia: http://www.scholarpedia.org/article/Policy\_gradient\_methods

*Q-Learning*. (n.d.). Retrieved May 3, 2017, from Wikipedia: https://en.wikipedia.org/wiki/Q-learning

*Recurrent Neural Network*. (2010). Retrieved May 3, 2016, from Deep Learning: http://deeplearning.net/tutorial/rnnslu.html

Rouse, M. (2016, July 16). *Definition: Neural Network*. Retrieved May 3, 2017, from Search Networking: http://searchnetworking.techtarget.com/definition/neural-network

Vrancx, P. (n.d.). *Deep reinforcement learning*. Retrieved May 3, 2017, from Vub Artifical Intelligence Lab: https://ai.vub.ac.be/thesis/deeprl

1. (Park & Kim, 2016) [↑](#footnote-ref-1)
2. (Bai & Chen, 2016) [↑](#footnote-ref-2)
3. (Bai & Chen, 2016) [↑](#footnote-ref-3)
4. (Lample & Chaplot, 2016) [↑](#footnote-ref-4)
5. (Park & Kim, 2016) [↑](#footnote-ref-5)
6. (Introduction VizDoom, 2016) [↑](#footnote-ref-6)
7. (Rouse, 2016) [↑](#footnote-ref-7)
8. (Convolutional neural network, 2010) [↑](#footnote-ref-8)
9. (Recurrent Neural Network, 2010) [↑](#footnote-ref-9)
10. (Berstekas, 1960) [↑](#footnote-ref-10)
11. (Q-Learning, sd) [↑](#footnote-ref-11)
12. (Vrancx, sd) [↑](#footnote-ref-12)
13. (Peters, 2010) [↑](#footnote-ref-13)