Project proposal
04-05-2017
Natural Computing

Tackling AI and control theory problems using Deep Reinforcement Learning

-Eastern Screech Owl

Jordi Beernink, Roel Bouman and Jeffrey Luppes

INTRODUCTION AND HYPOTHESIS

In recent years, deep learning has successfully become a staple method in various domains, such as object recognition with visual attention, Computer Aided Diagnosis in medical imaging, and many other classification problems. More recently, deep learning has been combined with reinforcement learning into the field of Deep Reinforcement Learning[1,2]. The newly developed method combines Q-learning[3], a Reinforcement Learning technique, with Deep Neural Networks, and was trained to play classic Atari 2600 games. The trained Deep Q-network, as the new Reinforcement Learning agent is called, was able to surpass human performance across several of 49 different games. More recent applications involve high-dimensional robot control, solving physics-based control problems, and the playing of the traditional Go game.[4]

On the 27th of April 2016, a large scale platform for Reinforcement Learning called OpenAI Gym[5] was launched. OpenAI gym provides a Python environment for the fair comparison of Reinforcement Learning techniques. The platform supports many classical problems from Reinforcement Learning theory, such as balancing a pole atop a minecart, swinging a pendulum upwards from a random starting position, and driving a mine cart atop a hill from a local minimum using momentum. More advanced problems are also presented, such as several of the classic Atari 2600 games, several games from the PyGame platform, and the classic shooter DOOM.

Deep Q-Learning methods, and other Deep Reinforcement Learning techniques, have already proven to be highly applicable to several of the problems represented on the OpenAI Gym platform. It is proposed that several staple Deep Reinforcement Learning algorithms be used for training on several of the listed problems presented on the OpenAI Gym platform. This will be done in a bottom up approach, as to efficiently study the effectiveness of each of the different algorithms, and to promote an efficient learning curve. This should yield competitive results for the staple problems. It is hypothesized that decent results on several of the DOOM challenges should be achievable; although a general DOOM playing AI seems to be out of reach. Note that in every case, a generalist approach will be used, with the rawest possible input for the training. This ensures an unsupervised approach, which is preferred over hand-crafted features often used in AI development.

METHODS AND MATERIALS

Several methods are applicable to these problems, and most will be incorporated in the Deep Reinforcement Pipeline. Most methods revolve around the neural network architecture, which can differ greatly between problems. The Google subsidiary DeepMind has achieved great performance with their Q-networks[1,2], although some elaboration on their structure is needed. This structure should form an adequate foundation for later exploration for improvements on network structure for more specific problems. To illustrate, the network architecture DeepMind used can be found in table 1.

Layer	Input	Filter size	Stride	Num filters	Activation	Output
conv1	84x84x4	8x8	4	32	ReLU	20x20x32
conv2	20x20x32	4x4	2	64	ReLU	9x9x64
conv3	9x9x64	3x3	1	64	ReLU	7x7x64
fc4	7x7x64			512	ReLU	512
fc5	512			18	Linear	18

Table 1: The Q-network architecture used by DeepMind on the Atari 2600 games.

It can be seen that the network has several convolutional layers, each of which takes an input of 84x84x4, which consists of four 84x84 grayscale images. These four images are the last four observed images from the game. Using the last four images allows for the abstraction of movement as features. As such, using multiple frames will be thoroughly experimented with in this project.

Of course, parameters such as the filter size, the stride, the number of filters, and the activation function can and should be examined for different problems. In addition, the number of layers might be changed, as adding layers might improve results. It should be noted that although larger networks might improve the performance, the network complexity is severely limited by the available hardware.

In cases like DOOM, the general structure of the network will be evaluated across different challenges, as to assess the general applicability of the trained Deep Q-Network.

The most notable of the Deep Reinforcement techniques to be used will be the well-known Deep Q-Network.[1,2] Deep Q-Networks try to estimate the state and action dependent Q-function using a Neural Network. Using a well-designed network, the long and short terms rewards for a given action sequence can be estimated, which facilitates the quick search for a decent solution, something which is not possible in large scale problems such as Go, Atari 2600 games, and DOOM, or non-deterministic problems. The Deep Q-Network presented by DeepMind also uses experience replay with random mini-batches from replay memory to prevent overfitting and convergence towards a local maximum reward. Another important addition is the exploration algorithm used. Their ε-greedy exploration chooses a random action with probability ε, instead of choosing the optimal action according to the highest Q-value. By decreasing ε from 1 to a lower value, exploration is initially encouraged, but used less and less as more of the training space is explored. Depending on the problem, the value and change rate of ε should be optimized.

In recent years several improvements have been made upon the staple Deep Q-Network technique. One of these is the Double Deep Q-Network, which combines Deep Reinforcement Learning with Double Q-learning[6]. In the Double Q-learning variant, the action selection is decoupled from the evaluation, which leads to less overoptimism due to estimation errors. Double Deep Q Networks will be experimented with, and it remains to be seen whether they significantly increase performs.

The architecture proposed in Double Deep Q-Networks was further extended into an architecture where the state value function and the state-dependent action advantage are completely decoupled.[7] This procedure leads to the development of a more robust agent, which in turn leads to a better policy evaluation when many actions are similarly valued, which was shown to be a weakness of Deep Q Networks. This new architecture has proven to be effective, and improved upon the Atari 2600 state-of-the-art, and should yield decent results when applied to the OpenAI Gym challenges. It must be noted that the required computation time should be extensively evaluated for such a different architecture, and whether possible increases are worth the performance.

For the development of the architecture, the well-developed Keras library[8] for the Python language can be used. Keras is a library developed for rapid prototyping of Neural Network applications, and it offers an extra layer of abstraction over either the Theano or TensorFlow libraries. Many different Neural Network structures can be tested using the Keras library, without the need to build the architecture from the ground up.

The inclusion of Deep Reinforcement Learning into the process is done using the Keras-rl library[9], which is an extension upon Keras focusing on Reinforcement Learning. It has several Reinforcement Learning techniques available, and can easily be extended. In addition, the Keras-rl library has built-in support for the OpenAI Gym platform, which was elaborated on earlier, and can thus easily be used on various challenges and easy implementation of a reward system.

References

1. Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *Nature* 518.7540 (2015): 529-533.
2. Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." *arXiv preprint arXiv:1312.5602* (2013).
3. Watkins, Christopher JCH, and Peter Dayan. "Q-learning." *Machine learning* 8.3-4 (1992): 279-292.
4. Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." *Nature* 529.7587 (2016): 484-489.
5. Brockman, Greg, et al. "OpenAI gym." *arXiv preprint arXiv:1606.01540* (2016).
6. Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning." *AAAI*. 2016.
7. Wang, Ziyu, et al. "Dueling network architectures for deep reinforcement learning." *arXiv preprint arXiv:1511.06581* (2015).
8. Chollet, François. "Keras." (2015).
9. Plappert, Matthias. "keras-rl" (2016).