Automated State Feature Learning for Actor-Critic Reinforcement Learning through NEAT

Yiming Peng, Gang Chen, Scott Holdaway, Yi Mei, Mengjie Zhang
School of Engineering and Computer Science, Victoria University of Wellington
yiming.peng,aaron.chen,scott.holdaway,yi.mei,mengjie.zhang@ecs.vuw.ac.nz

ABSTRACT

Actor-Critic (AC) algorithms are important approaches to solving sophisticated reinforcement learning problems. However, the learning performance of these algorithms rely heavily on good state features that are often designed manually. To address this issue, we propose to adopt an evolutionary approach based on NeuroEvolution of Augmenting Topology (NEAT) to automatically evolve neural networks that directly transform the raw environmental inputs into state features. Following this idea, we have successfully developed a new algorithm called NEAT+AC which combines Regular-gradient Actor-Critic (RAC) with NEAT. It can simultaneously learn suitable state features as well as good policies that are expected to significantly improve the reinforcement learning performance. Preliminary experiments on two benchmark problems confirm that our new algorithm can clearly outperform the baseline algorithm, i.e., NEAT.

CCS CONCEPTS

•Computing methodologies → Neural networks; •Computer systems organization → Embedded systems; *Redundancy;* Robotics;

KEYWORDS

NeuroEvolution, NEAT, Actor-Critic, Reinforcement Learning, Feature Extraction, Feature Learning

ACM Reference format:
Yiming Peng, Gang Chen, Scott Holdaway, Yi Mei, Mengjie Zhang. 2017. Automated State Feature Learning for Actor-Critic Reinforcement Learning through NEAT. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076035

1 INTRODUCTION

Reinforcement Learning (RL) aims to learn an optimal policy for sequential action selection while observing states in an unknown environment [7]. As an important RL algorithm family, Actor-Critic Reinforcement Learning (ACRL) algorithms are designed to directly search effective policies (a.k.a., actor) guided by value functions (a.k.a, critic) [2].

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3076035

Many ACRL algorithms usually assume the availability of suitable state features that are immediately accessible during reinforcement learning. However, for effective RL, these state features must be carefully designed with the support of domain experts via a time-consuming and error-prone procedure [1]. During the procedure, even for experienced domain experts, important state feature information may be overlooked, resulting in serious degradation of learning performance [4, 8].

To address this important issue, state-of-the-art learning algorithms have considered switching across different parametric functions (e.g. Radial Basis Function networks) [3] or optimizing some predefined score functions [5]. These techniques inevitably require substantial domain knowledge. Additionally, when neural networks are chosen as the feature base, its topology also requires to be well designed prior to the activation of any learning algorithms.

These new issues motivate us to consider exploiting an NeuroEvolution based approach towards fully automated state feature learning which can be performed simultaneously with any ACRL algorithm. Specifically, we are interested in a major EC method for NeuroEvolution, i.e, NeuroEvolution of Augmenting Topology (NEAT). This is because of several reasons: 1) Neural Networks (NNs) are well recognized as good feature bases for various learning paradigms including RL [1]. 2) NEAT has a strong capability of evolving both structure and weights simultaneously for effective reinforcement learning. 3) NEAT introduces a unique innovation number to each individual to preserve useful structural innovations for future learning. 4) NEAT adopts a strategy to evolve increasingly complicated NNs starting from the simplest structures. These properties are very important for our feature learning tasks.

Goals: Motivated by this understanding, the overall goal of this research is to develop a new algorithm (NEAT+AC) based on NEAT and Regular-gradient Actor-Critic (RAC) algorithm [2]. Through the seamless integration of NEAT and AC, we can learn good features, in the mean time use the learned features to identify desirable policies.

2 NEAT+AC

As seen in Figure 1, our NEAT+AC algorithm consists of four phases, including *initialization*, *evolution*, *evaluation*, and *termination*.

Initialization: Similar to the standard NEAT, NEAT+AC also starts with a population with a fixed number of randomly generated individuals. Each individual is designed differently from that of the standard NEAT. Since it is composed of three main parts, a NN, an actor and a critic.

Evolution: Aimed at searching good features, we use the standard evolutionary operators defined in [6], including crossover and mutation, to evolve solely the state feature extraction function $\phi(\vec{s})$.

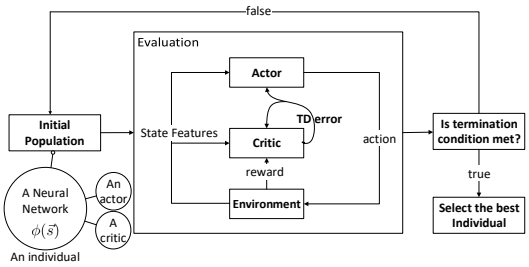

Figure 1: The proposed NEAT+AC algorithm.

Evaluation: In this phase, we compute the fitness value with respect to each individual in hope of discovering good features, and in the meantime gradually improve the performance of the corresponding policies. The fitness is defined as the average cumulative rewards obtainable through simulation over all training episodes, i.e., $N.fitness \leftarrow \frac{\tilde{R}}{e_g}$. The policy search is conducted by following RAC.

Termination: Since NEAT+AC consists of two components (NEAT and RAC), each of them has a termination condition. The feature learning process terminates either when the predefined maximum number of generations is reached, or when the highest fitness value cannot be further improved over 50 consecutive generations. Meanwhile, the RAC learning process terminates when the maximum number of training episodes is reached.

3 EXPERIMENTAL RESULTS

To verify any significant performance difference, we conduct 30 independent runs for both learning algorithms on each benchmark problem. In these runs, the population size and the number of generations are both set to 100. Also, while evaluating any individual in one single generation, 5000 training episodes need to be performed. Each episode contains 200 steps. After every generation, 50 independent tests will be conducted to verify the learning effectiveness of the evolved NN with the highest fitness.

Promising experimental results have been collected on the Mountain Car problem [7] and the Cart Pole problem [7]. Figure 2 shows that NEAT+AC performs significantly better than NEAT on the Cart Pole problem. This is supported by as a statistical test with a p-value of 1.18366×10^{-11}. The Mountain Car problem, on the other hand, is much harder for NEAT+AC than for NEAT, as NEAT considers only three optimal actions, while NEAT+AC must learn to select suitable actions from a continuous range. However, Figure 3 shows that NEAT+AC performs comparable with NEAT. Both algorithms reach the optimal solution very quickly (within 10 generations).

4 CONCLUSIONS

In this paper, we have successfully achieved the research goal of evolving useful NNs as feature extrators which accept raw state information as their input and subsequently produce a vector of state features to be subsequently utilized by RAC to learn desirable policies. It is clearly evidenced in the experiment results that NEAT+AC is an effective algorithm for reinforcement learning. Meanwhile, NEAT+AC is purposefully designed to ensure that every newly evolved NN will always be trained for the same number of episodes, starting from identical initial settings. In view of this

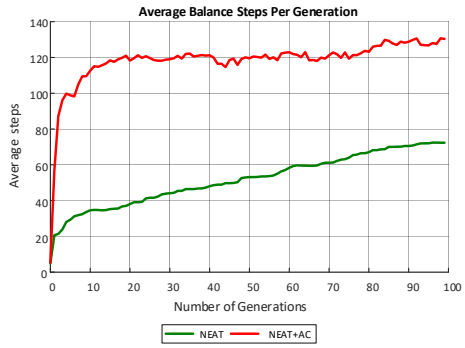

Figure 2: Average Balancing Steps Per Generation on the Cart Pole problem

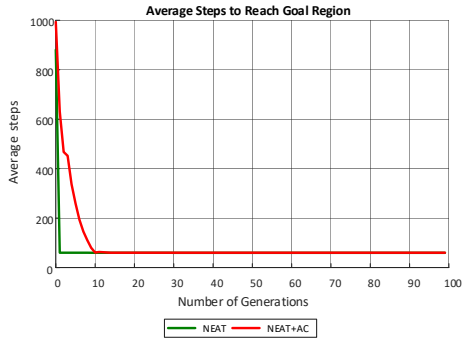

Figure 3: Average Steps Per Generation on the Mountain Car problem

fact, the steady improvement of learning performance during the evolutionary process, as witnessed in our experiments, confirms that NEAT+AC is capable of learning useful state features embodied in NNs.

There is a big room for future research. We plan to conduct more comprehensive experiments involving a wide range of benchmark problems to truly understand the real efficacy of NEAT+AC. We will also study the possibility of exploiting other cutting-edge reinforcement learning algorithms under the same NEAT-based learning framework.

REFERENCES

[1] Yoshua Bengio, Aaron Courville, and Pierre Vincent. 2013. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, 8 (2013), 1798–1828.
[2] Shalabh Bhatnagar, Richard S. Sutton, Mohammad Ghavamzadeh, and Mark Lee. 2009. Natural actor-critic algorithms. *Automatica* 45, 11 (2009), 2471–2482.
[3] George Konidaris, Sarah Osentoski, and Philip Thomas. 2011. Value Function Approximation in Reinforcement Learning using the Fourier Basis. *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence* (2011), 380–385.
[4] Ishai Menache, Shie Mannor, and Nahum Shimkin. 2005. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research* 134, 1 (2005), 215–238.
[5] Ronald Parr, Christopher Painter-Wakefield, and Lihong Li. 2007. Analyzing feature generation for value-function approximation. *Proceedings of the 24th International Conference on Machine Learning (ICML)* (2007), 737–744.
[6] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving neural network through augmenting topologies. *Evolutionary computation* 10, 2 (2002), 99–127.
[7] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement Learning : An Introduction.*
[8] M Wiering and M van Otterlo. 2012. *Reinforcement Learning: State-of-the-Art.* Springer Berlin Heidelberg.