

Report for Research Project

Evolutionary Function Approximation for Reinforcement Learning

Yunian Pan

Abstract—This Report presents the substances of the paper "Evolutionary Function Approximation for Reinforcement Learning", in the context that temporal difference methods are theoretically grounded and empirically effective methods for addressing reinforcement learning problems, and as we move towards an era of deep learning that in most real-world reinforcement learning tasks, TD methods require a function approximator to represent the value function. However, using function approximators requires manually making crucial representational decisions. The related papers investigate evolutionary function approximation, a novel approach to automatically selecting function approximator representations that enable efficient individual learning. The method, which naturally arises from genetic algorithm schemes, evolves individuals that are better able to learn.

A fully implemented instantiation of evolutionary function approximation which combines NEAT, a neuroevolutionary optimization technique, with Q-learning, a popular TD method was presented in paper [?]. The resulting NEAT+Q algorithm automatically discovers effective representations for neural network function approximators. This paper also presents on-line evolutionary computation, which improves the on-line performance of evolutionary computation by borrowing selection mechanisms used in TD methods to choose individual actions and using them in evolutionary computation to select policies for evaluation.

Also some online searching scheme has been proposed in this paper based on epsilon greedy method or Boltzmann distribution, which improves the on-line performance of evolutionary computation by borrowing selection mechanisms used in TD methods to choose individual actions and using them in evolutionary computation to select policies for evaluation. Two empirical experiments were evaluated and discussed in this paper, it also presents additional tests that offer insight into what factors can make neural network function approximation difficult in practice.

Index Terms—reinforcement learning, temporal difference methods, NEAT, Q-learning, Online evolutionary computation

I. Introduction

REINFORCEMENT problems are the subset of machine learning tasks in which the agent never sees

examples of correct behavior. Instead, it receives only positive and negative rewards for the actions it tries. Since many practical, real world problems (such as robot control, game playing, and system optimization) fall in this category, developing effective reinforcement learning algorithms is critical to the progress of artificial intelligence.

The most common approach to reinforcement learning relies on the concept of value functions, which indicate, for a particular policy, the long-term value of a given state or state-action pair. Temporal difference methods (TD(λ)) (Sutton, 1988), which combine principles of dynamic programming with statistical sampling, use the immediate rewards received by the agent to incrementally improve both the agent's policy and the estimated value function for that policy. Hence, TD methods enable an agent to learn during its lifetime i.e. from its individual experience interacting with the environment.

When the problem scale is not large, the value function can be represented as a table of state-action pairs, but for large scale systems or for problems that have infinite dimension of state space or action space, it remains problematic to only use a "table", instead the function approximator which represents the mapping from state-action pairs to values via a more concise, parameterized function and uses supervised learning methods to set its parameters, plays an important role where we can omit the step of enumerating over the state-action space. Many different methods of function approximation have been used successfully, including CMACs, radial basis functions, and neural networks.

However, using function approximators requires making crucial representational decisions (e.g. the number of hidden units and initial weights of a neural network. Poor design choices can result in estimates that diverge from the optimal value function and agents that perform poorly. Even for reinforcement learning algorithms with guaranteed convergence, achieving high performance in practice requires finding an appropriate representation for the function approximator.

As Lagoudakis and Parr observe,^[?] The crucial factor for a successful approximate algorithm is the choice of the parametric approximation architecture(s)

and the choice of the projection (parameter adjustment) method. (Lagoudakis and Parr, 2003, p. 1111) Nonetheless, representational choices are typically made manually, based only on the designers intuition.

Therefore, in order to automate the search for effective representations by employing sophisticated optimization techniques, we focus on using evolutionary methods because of their demonstrated ability to discover effective representations, so that our proposition can cover the 2 things: 1) an evolutionary algorithm capable of optimizing representations from a class of functions; 2) a TD method that uses elements of that class for function approximation.

Neural network is a good choice as they have great experimental value, Nonlinear function approximators are often the most challenging to use, hence, success for evolutionary function approximation with neural networks is good reason to hope for success with linear methods too; and they have great potential in that they can represent value functions that the linear method cannot(given the same basis functions). Therefore NeuroEvolution of Augmenting Topologies(NEAT)^[?] has been chosen as the main framework. The resulting algorithm, called NEAT+Q, uses NEAT to evolve topologies and initial weights of neural networks that are better able to learn, via backpropagation, to represent the value estimates provided by Q-learning.

Section 2 will present the background including Q-learning and deep Q-learning, genetic algorithms and NEAT framework and NEAT algorithm for reinforcement learning and some basic related experiments; Section 3 will present the new emerging method that incorporates the previous algorithm together, along with the experiments and discussion. Section 4 will have some discussion over the results obtained from the experiments.

II. Background

This section mainly contains 2 part, one is for the development of Q-learning, Deep Q learning, and some variance of them; The another will discuss the machinery and framework and using NEAT to do reinforcement learning.

A. From Q-learning to Deep Q-learning

1) *Q-learning*: As a model-free algorithm, Q-learning trains the agent what action to take under what circumstances, is a well-established, canonical method that has also enjoyed empirical success, particularly when combined with neural network function approximators,

B. Q-Learning with NN

Algorithm 1: Q-Learning

Input: S : set of all states; A : set of all actions; σ : standard deviation of initial weights; c : output scale; α : learning rate; γ : discount factor; λ : eligibility decay rate; ϵ_{td} : exploration rate; e : total number of episodes;

Initialize: $N \leftarrow \text{INIT} - \text{NET}(S, A, \sigma)$;

for $i \leftarrow 1$ **to** e **do**

$s, s' \leftarrow \text{null}, \text{INIT-STATE}(S)$;

while $\text{Terminal-state}(s)$ **do**

$Q[] \leftarrow c \times \text{EVAL} - \text{NET}(N, s')$;

With-prob (ϵ_{td}) $a' \leftarrow \text{RANDOM}(A)$;

else: $a' \leftarrow \arg \max Q[j]$;

if $s \neq \text{null}$ **then**

$\text{BACKPROP}(N, s, a, (r + \gamma \max_j Q[j])/c, \alpha, \gamma, \lambda)$;

else

$s, a \leftarrow s', a'$;

$r, s' \leftarrow \text{TAKE} - \text{ACTION}(a')$

end

end

end

C. From genetic algorithm to NEAT

1) :

D. Experiments

E. Final conclusion

References

- [1] Whiteson, Shimon, and Peter Stone. "Evolutionary function approximation for reinforcement learning." *Journal of Machine Learning Research* 7.May (2006): 877-917.
- [2] Stanley, Kenneth O., and Risto Miikkulainen. "Efficient reinforcement learning through evolving neural network topologies." *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2002.