Improving Reinforcement Learning Function Approximators via Neuroevolution

Shimon Whiteson
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712 USA
shimon@cs.utexas.edu

ABSTRACT

Reinforcement learning problems are commonly tackled with *temporal difference methods*, which estimate the long-term value of taking each action in each state. In most problems of real-world interest, learning this value function requires a *function approximator*. However, the feasibility of using function approximators depends on the ability of the human designer to select an appropriate representation for the value function. My thesis presents a new approach to function approximation that automates some of these difficult design choices by coupling temporal difference methods with policy search methods such as evolutionary computation. It also presents a particular implementation which combines NEAT, a neuroevolutionary policy search method, and Q-learning, a popular temporal difference method, to yield a new method called NEAT+Q that automatically learns effective representations for neural network function approximators. Empirical results in a server job scheduling task demonstrate that NEAT+Q can outperform both NEAT and Q-learning with manually designed neural networks.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Connectionism and neural nets*

General Terms

Experimentation

Keywords

reinforcement learning, genetic algorithms, neural networks

1. THESIS OVERVIEW

In reinforcement learning, an agent must learn a *policy* for selecting actions based on its *state* without ever seeing examples of correct behavior. The most common approach to reinforcement learning relies on *temporal difference methods*[2], which use dynamic programming and statistical sampling to estimate the long-term value of taking each possible action in each possible state. For small problems, the value function can be represented in a table but for most problems of real-world interest, temporal difference methods must be coupled with a *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. In practice, the feasibility of using function approximators depends largely on the ability of the human designer to select an appropriate representation (e.g. the topology and initial weights of the neural network). Unfortunate design choices can result in estimates that diverge wildly from the optimal value function and agents that perform extremely poorly.

This thesis presents a new approach to doing function approximation that automates some of these difficult design choices by coupling temporal difference methods with policy search methods such as evolutionary computation. In particular, I use NeuroEvolution of Augmenting Topologies (NEAT) [1], a method that uses evolutionary computation to learn both the topology and weights of neural networks, in conjunction with Q-learning, a popular temporal difference method. The resulting method, called NEAT+Q, uses NEAT to learn the topology and initial weights of networks which are then updated, via backpropagation, towards the value estimates provided by Q-learning.

NEAT+Q also makes it possible to take advantage of the Baldwin Effect, a phenomenon whereby populations whose individuals learn during their lifetime adapt more quickly than populations whose individuals remain static. In the Baldwin Effect, which has been demonstrated in evolutionary computation, evolution proceeds more quickly because an individual does not have to be exactly right at birth; it need only be in the right neighborhood and learning will adjust it accordingly. By combining learning *across* fitness evaluations with learning *within* fitness evaluations, NEAT+Q has the potential to reap the Baldwin Effect.

This thesis presents empirical results from the domain of server job scheduling, a challenging reinforcement learning task from the burgeoning field of *autonomic computing*. My experiments demonstrate that NEAT+Q, by automatically discovering appropriate topologies and initial weights, can dramatically outperform a Q-learning approach that uses manually designed neural networks. These experiments also demonstrate that NEAT alone does not perform as well as NEAT+Q, which harnesses the power of temporal difference methods.

2. REFERENCES

[1] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[2] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.
Copyright 2005 ACM 1-59593-094-9/05/0007 ...$5.00.