Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 63 (2005) 229–251
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Asynchronous neurocomputing for optimal control and reinforcement learning with large state spaces

Bruno Scherrer[a,b,*]

[a]Computer Science and Artificial Intelligence Laboratory, NE43-783, 200 Technology Square, Cambridge, MA 02139-4307, USA
[b]LORIA INRIA-Lorraine, CORTEX/MAIA Teams, Campus Scientifique BP 239, 54506 Vandœuvre-lès-Nancy, France

Received 23 October 2003; accepted 23 January 2004
Available online 20 August 2004

Abstract

We consider two machine learning related problems, optimal control and reinforcement learning. We show that, even when their state space is very large (possibly infinite), natural algorithmic solutions can be implemented in an asynchronous neurocomputing way, that is by an assembly of interconnected simple neuron-like units which does not require any synchronization. From a neuroscience perspective, this work might help understanding how an asynchronous assembly of simple units can give rise to efficient control. From a computational point of view, such neurocomputing architectures can exploit their massively parallel structure and be significantly faster than standard sequential approaches. The contributions of this paper are the following: (1) We introduce a theoretically sound methodology for designing a whole class of asynchronous neurocomputing algorithms. (2) We build an original asynchronous neurocomputing architecture for optimal control in a small state space, then we show how to improve this architecture so that also solves the reinforcement learning problem. (3) Finally, we show how to extend this architecture to

Abbreviations: AN: asynchronous neurocomputing; CM: contraction mapping; CMFP: contraction mapping fixed point; MDP: Markov decision process; PADC: parallel asynchronous distributed computation; RL: reinforcement learning

*Corresponding author. Computer Science and Artificial Intelligence Laboratory, NE43-783, 200 Technology Square, Cambridge, MA 02139-4307, USA.
E-mail address: scherrer@ai.mit.edu (B. Scherrer).

doi:10.1016/j.neucom.2004.01.192

address the case where the state space is large (possibly infinite) by using an asynchronous neurocomputing adaptive approximation scheme. We illustrate this approximation scheme on two continuous space control problems.

Keywords: Neurocomputing; Optimal control; Reinforcement learning

0. Introduction

Research in neurocomputing combines two intimately related (and sometimes contradictory) motivations (1) doing good computer science using interesting ideas taken from neuroscience and (2) understanding neuroscience issues better with computer (theoretical or simulation-based) modelling. We think that a way to satisfy both motivations at the same time is to make or strengthen the relation between a certain computational "intellectual" capacity C (memory, generalization, control, etc.) and a specific brain-like process P. To achieve this, one encounters two opposite and complementary trends in the literature: (a) the *bottom-up* neurocomputing researchers copy real neurons, study the resulting process P from a computational point of view, and update the model until it exhibits capacity C; while, (b) the *top-down* neurocomputing researchers formalize the computational "intellectual" capacity C and try to find a process P which respects most of the constraints of the neurocomputing paradigm. There are advantages and drawbacks in both approaches: the former is often closer to neuroscience but tends to be more empirical. The latter is closer to computational science but often lacks biological plausibility. The research we present in this paper belongs to the latter top-down approach: we consider the related capacities of control and reinforcement learning as they are formalized in the machine learning literature, and show that, even when the problem is hard (when the state space is big), they can be addressed by neurocomputing. Rather than proposing new computational methods for solving these machine learning problems, our aim is here to show that their standard mathematically-motivated solutions are naturally compatible with the neurocomputing paradigm.

As there is no general accepted definition of what neurocomputing is, (and to make sure that the reader can quickly understand the assumptions of this work), we now explain what we exactly mean by neurocomputing. In this paper, a neurocomputing algorithm will be an assembly of interconnected units. Each such unit will have a number of internal variables (or internal states) and will receive a number of input variables from other units. A unit will only be able to perform basic computations (*in this paper:* `sum`, `max`, `argmax` and access to a composant of a finite local array) with all these variables and store the result in one of its local variables. Pieces of numerical information will be able to flow from specific units' variables (called units' outputs) to other units through weighted connections; when a numerical value goes from one unit to another through a given connection,