

EXPLOITING TRAINING EXAMPLE PARALLELISM WITH A BATCH VARIANT OF THE ART 2 CLASSIFICATION ALGORITHM

Philipp Ciechanowicz, Stephan Dlugosz, Herbert Kuchen, Ulrich Müller-Funk
Department of Information Systems
University of Münster
Leonardo-Campus 3
D-48159 Münster, Germany
{ciechanowicz | dlugosz | kuchen | mueller-funk}@ercis.de

ABSTRACT

In this article we develop a batch variant of the ART 2 classification algorithm invented by Carpenter and Grossberg. Our algorithm exploits training example parallelism while leaving the overall design of the ART 2 network unchanged such that a significant reduction of the execution time can be achieved on a multiprocessor system. We present a parallel implementation strategy and analyze it w.r.t. execution time and speedup. As our algorithm naturally benefits from data parallelism, the implementation uses data parallel skeletons of the Muenster skeleton library *Muesli*. We show that skeletons are an efficient way to write parallel applications compared to a manual MPI implementation.

KEY WORDS

ART 2, batch, skeletons, training example parallelism

1 Introduction

Since the beginnings in the early 1950's, artificial neural networks (ANN) have become essential instruments in a wide area of application. Especially when it comes to pattern recognition or classification, neural networks are almost indispensable and clearly outperform alternative approaches. The adaptive resonance theory (ART) developed by Carpenter and Grossberg describes a whole family of neural networks particularly applicable for classification tasks. Of peculiar interest are ART 2 networks [1, 2], as they are able to process real-valued (analog) input data.

In order to use an ART 2 network it first has to be trained. This is a time consuming issue, since typically a lot of training examples have to be processed, each requiring to solve a differential equation. Although this process was speed up by the development of the ART 2-A algorithm [3], a lot of research has been conducted to parallelize algorithms and/or architectures for neural networks [4, 5]. Most of these approaches aim at exploiting *node parallelism* [6], which is the most natural form of parallelizing a neural network. This approach is based on the fact that the activation of a neuron at an arbitrary layer is normally independent of the activation of any other neuron at the same layer. Thus, the activation of each node can be computed in parallel. In contrast, *training exam-*

ple parallelism divides the whole training set into multiple chunks and trains multiple neural networks with these chunks in parallel. Although researchers were successful at parallelizing backpropagation networks [7, 8], none of the approaches known to us implemented training example parallelism with an ART 2 network. This paper presents a modified ART 2 learning algorithm capable of exploiting training example parallelism while training the network in batch mode, i.e. any adjustments to the neural network are only executed after looking at *all* training examples. For this reason the algorithm is referred to as *BART*.

This paper is structured as follows: Section 2 briefly summarizes the main concepts of ART 2, establishing a basis for our BART algorithm presented in Section 3. While Section 4 describes some pre-implementation considerations, Section 5 presents the parallel implementation and discusses benefits and drawbacks. In order to show the effectiveness of our approach we tested our implementation on a workstation cluster. The results are presented in Section 6. Finally, Section 7 summarizes the main conclusions of this paper and gives an outlook to future work.

2 ART 2 Fundamentals

ART 2 networks [1, 2] are neural networks capable of classifying real-valued input vectors in an unsupervised way. This means that they are able to autonomously adapt their internal state without the feedback of an external teacher. Furthermore, ART 2 networks autonomously decide how many classes to form, i.e. the network is not forced to use all possible classes. These characteristics make ART 2 networks a perfect choice for a lot of practical classification tasks. Figure 1 shows a cross-section of an ART 2 network. The network can be divided into three layers, each consisting of multiple neurons. Neurons at layers F_0 and F_1 process n -dimensional vectors, while neurons at layer F_2 process m -dimensional vectors, with $n, m \in \mathbb{N}^+$. Here, n denotes the dimension of the input vectors and m denotes the maximal number of classes to learn. The mapping between these two dimensions is achieved by the two weight matrices BU (bottom up) and TD (top down), where the former is of dimension $n \times m$, the latter of dimension $m \times n$. As BU merely is a transposition of TD , we will refer to both