



Hardware acceleration of homogeneous and heterogeneous ensemble classifiers



Vuk S. Vranjković, Rastislav J.R. Struharik*, Ladislav A. Novak

Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, Novi Sad, 21000, Serbia

ARTICLE INFO

Article history:

Available online 21 October 2015

Keywords:

Homogeneous and heterogeneous ensemble classifiers
Decision trees
Support vector machines
Artificial neural networks
Hardware acceleration
Reconfigurable hardware

ABSTRACT

In this paper a universal reconfigurable computing architecture for hardware implementation of homogeneous and heterogeneous ensemble classifiers composed from decision trees (DTs), artificial neural networks (ANNs), and support vector machines (SVMs) is proposed. The following types of ensemble classifiers have been implemented in FPGA using proposed architecture: homogeneous ensemble classifiers composed from two versions of DT (Functional DT and Axis-Parallel DT), two versions of SVM (with polynomial and radial kernel) and two versions of ANN (Multilayer Perceptron ANN and Radial Basis ANN) machine learning predictive models, as well as a number of types of heterogeneous ensemble classifiers composed of a mixtures of DTs, SVMs and ANNs. Comparison of the FPGA implementation of REC architecture with standard WEKA software implementation suggests that proposed hardware architecture offers substantial speed-ups for all types of considered machine learning ensemble classifiers, ranging from 10^2 to 10^5 times.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Machine learning [1,2] as a branch of artificial intelligence can be viewed as a set of procedures/algorithms for construction of systems that can learn from data, using appropriate representations of data instances and functions to be learned. The main feature of machine learning systems is requirement to perform well on previously unseen data instances (generalisation property).

A wide range of machine learning predictive models have been introduced in the open literature, including decision trees (DTs) [3,4], support vector machines (SVMs) [5] and artificial neural networks (ANNs) [6]. In particular, machine learning predictive models have been widely used in data mining (see e.g. [7]), among which DTs, SVMs and ANNs are the most popular (e.g. [8–10]).

To reduce dependence on the peculiarities of a single training set (variance) and enable predictive models to learn more expressive concept class than a single classifier system (that is to reduce bias), ensemble classifier systems [11] have been introduced. The main idea behind ensemble classifier is to create a set of classifiers and combine their predictions into a collective decision. Key step is to create an ensemble of diverse classifiers from a single training set.

In large-scale classification problems reduction of individual instance classification time is one of the key concerns. Reduction of

classification time typically appears in data mining [7], or in classification problems requiring real-time data processing (e.g. machine vision [12,13], bioinformatics [14,15], web mining [16,17], text mining [18,19], etc.). Developing new algorithms or software tools is the predominant approach in addressing these concerns [20–26].

In the course of reduction of individual instance classification time, hardware implementation of machine learning classifier systems is a promising alternative. Most of existing hardware implementations are concerned with acceleration of individual machine learning classifier systems, DTs [27–32], SVMs [33–39] and ANNs [40–45]. Recently, reconfigurable hardware architecture capable of implementing individual DT, SVM or ANN classifiers [46] has been proposed.

Concerning the hardware acceleration of ensemble classifier systems, according to our best knowledge, most proposed solutions are related to the hardware implementation of homogeneous ensemble classifiers [47–51]. Furthermore, as we know, all existing architectures for hardware implementation of homogeneous ensemble classifier systems, available in the open literature, are designed to implement only one type of homogeneous ensemble classifier, e.g., ensemble composed of DTs or SVMs or ANNs.

Bermak et al. [47] described a 3D VLSI chip for the implementation of a homogeneous DT ensemble, where each DT is implemented as a one-layer threshold network. Osman et al. [48] presented a hardware implementation of random forest ensemble classifier, using LNS arithmetic, that was used in the object recognition system. Van Essen et al. [49] compared FPGA, GP-GPU and multi-core CPU implementations for instance classification acceleration of random

* Corresponding author. Tel.: +381 63552364.

E-mail addresses: bykbpa@uns.ac.rs (V.S. Vranjković), rasti@uns.ac.rs (R.J.R. Struharik), ladislav@uns.ac.rs (L.A. Novak).

forest ensemble classifiers generated by compact random forest (CRF) algorithm. Hussain et al. [50] presented a single core K -NN classifier, as well as homogeneous K -NN ensemble classifier, implemented as a multi-core system implemented using FPGA devices. Struharik et al. [51] proposed several architectures for hardware acceleration of axis-parallel, oblique and non-linear DT ensemble classifiers, as well as a number of parallel and sequential architectures for hardware implementation of various combination rules.

The only proposed solution to the hardware implementation of heterogeneous ensemble classifier is the one presented in [52]. Solution, proposed by Shi et al. in [52], is hardware optimised to implement only one specific instance of one specific type of heterogeneous ensemble, called Committee Machine, consisting from following individual classifiers: K Nearest Neighbors, Multilayer Perceptron ANN, Radial Basis Function ANN, and Probabilistic Principal Component Analysis (PPCA). Furthermore, the architecture of every individual classifier is fixed and cannot be changed. Although being capable of implementing a heterogeneous ensemble classifier, system presented by Shi et al. in [52] cannot be considered as universal and configurable, since the size of the ensemble and its composition cannot be changed, as well as the parameters of individual ensemble members.

In this paper we propose universal reconfigurable computing architecture, called Reconfigurable Ensemble Classifier (REC), for hardware acceleration of homogeneous and heterogeneous ensemble classifiers composed from DTs, ANNs, and SVMs. As far as we know, this is the very first appearance of a universal architecture capable to implement homogeneous ensembles of DTs, SVMs and ANNs, as well as heterogeneous ensembles involving mixture of DTs, SVMs and ANNs. Proposed architecture is fully configurable, allowing user to specify the size of the ensemble, its composition, as well as the characteristics of every individual ensemble member. Due to the reconfigurability of proposed architecture, all these features can be modified during the system operation. Moreover, comparison with the standard software solution (WEKA) suggests that proposed architecture offers substantial speed-ups, ranging from 2 to 5 orders of magnitude for all types of considered machine learning ensemble classifiers.

REC architecture, proposed in this paper, presents a further evolution of the architecture proposed in [46], with following important improvements:

- Architecture proposed in [46] can be used to implement only individual machine learning classifiers. It cannot be used to implement neither homogeneous nor heterogeneous ensemble classifier systems. To achieve this, architecture from [46] had to be thoroughly redesigned, including several new modes of operation for each reconfigurable block. This redesign also resulted in improved classification efficiency of the REC architecture when compared with the architecture from [46].
- In the architecture presented in [46] all reconfigurable blocks in one column are configured using the same configuration information at the same time. To allow ensemble classifier implementation this had to be changed, so every reconfigurable block can be configured independently.
- To enable easy SoC integration, REC architecture uses standard AXI-Stream interface, which is not the case for the architecture proposed in [46].
- In the setup of experiments, used to estimate classification performance of REC architecture, WEKA software is used, because it has excellent support for ensemble classifiers systems. Experiments presented in [46] have been based on the R Project software, whose support of ensemble classifiers is very limited.
- Several very large machine learning benchmark datasets have been used in the experiments, to better estimate classification performance in more realistic applications.

Section 2 highlights features of ensemble classifiers systems that are relevant for the proposed reconfigurable hardware architecture. Details of the coarse-grained reconfigurable hardware architecture are presented in the Section 3. Section 4 provides comparison of the classification speed between the FPGA implementation of proposed reconfigurable hardware architecture and the software implementation, based on the WEKA software platform.

2. Ensemble classifiers

In this section we provide a brief account on ensemble classifiers based on DTs, SVMs and ANNs. Ensemble classifiers have been in the focus of artificial intelligence and machine learning community in the last few decades and have proved to be an efficient tool in solving a number of different machine learning tasks, such as feature selection, confidence estimation, incremental learning, error correction, learning concept drift from non-stationary distributions, etc.

The main idea behind ensemble classifier is using an appropriate combination rule to combine predictions from the set of diverse individual classifiers (two classifiers are considered as diverse if they make different errors on new instances), provided that the accuracy of each individual classifier is greater than 50%. Under these assumptions, an ensemble classifier could achieve arbitrary classification accuracy using the power of making collective decision.

Generally speaking, ensemble could consist of classifiers with different learning paradigms (so called heterogeneous ensembles). Ensemble consisting of classifiers with the same learning paradigm is also called homogeneous. In our case ensemble involves classifiers with three types of learning paradigms, that is DTs, SVMs and ANNs. In the open literature there are a variety of algorithms for creation of ensemble members, such as *Bagging*, *Boosting*, *AdaBoost* [53], *Stacked generalization* [54], *Mixture of experts* [55] and variety of combination rules, including *majority voting* (*unanimous voting*, *simple majority voting*, *plurality voting* and *weighted majority voting*) [11], *Behavioural Knowledge Space* [56] and *Borda Count*.

In this paper we are not concerned with the hardware acceleration of ensemble classifier building algorithms. The main result of this paper is related to the hardware acceleration of already created ensemble classifiers. In what follows, we provide a brief description of DTs, SVMs and ANNs.

Decision trees [3] are graph-like structures in which internal node represents test on some attributes, each branch represents one possible outcome of the test and each leaf node represents a class label. DT classifies an instance by performing tests associated with visited nodes, while traversing a path from the root node, through intermediary nodes, until a leaf node is reached. Test function, which is applied in every DT node, in general, has the following form:

$$f(\mathbf{a}) = \mathbf{w} \cdot \mathbf{a} + b \quad (1)$$

where \mathbf{w} denotes normal vector of the separating hyperplane in the attribute space, \mathbf{a} is an input vector, b is scalar and \cdot is dot product.

Support vector machines [5] classify data by constructing a hyperplane in a finite or infinite dimensional attribute space. Hyperplane can be expressed as the linear combination of so-called kernel-mapped support vectors, derived from the training set instances during the learning process. Previously unseen instance is classified by the SVM by checking the sign of the following expression

$$\hat{V}(\mathbf{a}) = \sum_{j=1}^m \alpha_j \cdot \text{class}_j \cdot k(\mathbf{s}_j, \mathbf{a}) + b \quad (2)$$

where α_j are linear coefficients, $k()$ is the kernel function, \mathbf{s}_j are support vectors and \mathbf{a} is the input vector.

A variety of functions can serve as the kernel function. The most commonly used kernels are the radial kernel (also known as Gaussian

Download English Version:

<https://daneshyari.com/en/article/461306>

Download Persian Version:

<https://daneshyari.com/article/461306>

[Daneshyari.com](https://daneshyari.com)