



A novel recursive algorithm used to model hardware programmable neighborhood mechanism of self-organizing neural networks

Marta Kolasa^a, Tomasz Talaska^a, Rafał Długosz^{a,b,*}

^a UTP University of Science and Technology, Faculty of Telecommunications, Computer Science and Electrical Engineering, ul. Kaliskiego 7, 85-796 Bydgoszcz, Poland

^b Institute of Microtechnology of Swiss Federal Institute of Technology in Lausanne, Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland

ARTICLE INFO

Keywords:

Unsupervised learning algorithms
Recursive neighborhood mechanism
Topology of neural network
Software and hardware implementations

ABSTRACT

In this paper we propose a novel recursive algorithm that models the neighborhood mechanism, which is commonly used in self-organizing neural networks (NNs). The neighborhood can be viewed as a map of connections between particular neurons in the NN. Its relevance relies on a strong reduction of the number of neurons that remain inactive during the learning process. Thus it substantially reduces the quantization error that occurs during the learning process. This mechanism is usually difficult to implement, especially if the NN is realized as a specialized chip or in Field Programmable Gate Arrays (FPGAs). The main challenge in this case is how to realize a proper, collision-free, multi-path data flow of activations signals, especially if the neighborhood range is large. The proposed recursive algorithm allows for a very efficient realization of such mechanism. One of major advantages is that different learning algorithms and topologies of the NN are easily realized in one simple function. An additional feature is that the proposed solution accurately models hardware implementations of the neighborhood mechanism.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Artificial neural networks (ANNs) can be seen as universal expert systems, that are more and more frequently used in various areas including engineering and medical ones [1,2]. Usually they are considered from the point of view of a development and/or an optimization of learning algorithms specific for particular architectures of the NNs. A substantially different problem is how to translate a mathematical description of a given learning algorithm, sometimes very complex, into its efficient realization. Looking from the point of view of the second criterion, the most popular are software realizations, as they are very flexible. In practice any algorithm can be easily implemented in software. Unfortunately, pure software systems are not suitable for some kinds of applications that include portable miniaturized devices that have to consume ultra low power. An interesting example of such systems are Wireless Body Area Networks (WBANs), recently more and more frequently used in medical diagnostics [3–6].

Hardware realization that are an alternative, usually mean Field Programmable Gate Arrays (FPGAs) [7–10] but also, more rarely, Application Specific Integrated Circuits (ASICs) realized in the ‘full custom’ style [8,11–14]. A challenge in this case

* Corresponding author at: UTP University of Science and Technology, Faculty of Telecommunications, Computer Science and Electrical Engineering, ul. Kaliskiego 7, 85-796 Bydgoszcz, Poland.

E-mail addresses: markol@utp.edu.pl (M. Kolasa), talaska@utp.edu.pl (T. Talaska), rafal.dlugosz@gmail.com (R. Długosz).

relies on the necessity of solving some specific problems not known in software engineering. In digital ASICs the main problems we have to face with is the minimization of the number of transistors used to implement particular functions and modules, as this factor has a direct influence on the energy consumption, the computation power and area occupied by the overall system. Another important problem is a proper synchronization of particular building blocks to avoid glitches in the signals, which raise up the power dissipation. In analog realizations we have to additionally face with various phenomena of physical nature, such as charge injection effect, leakage in analog memory cells, parasitic capacitors and the transistor mismatch. These phenomena are the source of many problems which significantly disturb the outcomes of the learning process.

The disadvantages of hardware realization are diminished by several paramount features. One of them is a strong ability of parallel data processing, which allows for reaching computation power related to consumed energy not available in software systems [11,15]. However, to make realization of such systems possible, a careful design is required, which in practice means a proper modeling of the system. In the literature one can find many examples of modeling of ANNs. In fact, each software implementation to some extent can be seen as a model. Our new investigations in this area results from the fact that the existent models are insufficient when we cope with ANNs realized at the transistor level, which is our objective. The problem is not trivial. We have to deal not only with development and optimization of specific algorithms, which means determination of the architecture of the NN, but also on such an optimization of this algorithm to reduce the power dissipation and the chip area of the resultant chip.

The presented work is an important step of a bigger project that aims at development of an ultra low power parallel neuroprocessor that will be used in new generation sensing nodes in WBANs used in the analysis of the electrocardiography signals (ECG). A typical WBAN consists of a set of small devices, each equipped with a biosensor or a set of biosensors, allocated or implanted in the human body close to the place of signal acquisition that are capable to communicate wirelessly with a base station (Master Processing Unit – MPU). In a typical WBAN sensors (nodes) usually perform only basic tasks, such as: data collection, analog-to-digital conversion, simply data preprocessing and conditioning (filtering). Finally, the processed data throughout the radio frequency (RF) communication block are being transmitted to the MPU for a further detailed processing and analysis. One of the main problems encountered in such systems today is very large amount of energy lost (even 95% of total energy) during the RF transmission of data that significantly reduces the battery lifespan [16]. This becomes one of the barriers in a development a truly wearable systems. Development of a miniaturized and low power ANNs, which is our objective, will enable the realization of new intelligent sensors that itself will be able to preform data analysis and classification and thus will reduce the energy consumed during data transmission. The resultant WBAN composed of such sensors will be much more convenient in use and cheaper.

One of main challenges both in hardware and in software approaches is an efficient realization of the neighborhood mechanism [11,14]. This mechanism is closely related to the NN topology and determines the map of connections between particular neurons in the NN. The most popular topologies are the rectangular one with four or eight neighboring neurons (Rect4 and Rect8) and the hexagonal one with six neighbors (Hex) [17]. They are schematically shown in Fig. 1. Our former investigations showed that particular topologies are optimal in different scenarios [11,15], so it is well if a NN realized either in software or in hardware leaves the user a choice of which topology to choose. The structure of the neighborhood mechanism depends also on the type of the learning algorithm. All these issues are discussed in details in next Section.

In our investigations we focus mostly on self-organizing learning algorithms, as they offer relatively simple structures with usually only simple arithmetic operations such as additions, subtractions, multiplications, shifting the bits and the abs() function. Nevertheless, such NNs has proven to be suitable for the analysis of various biomedical signals, including ECG signals [2,18]. In next Section we briefly discuss example algorithms of this type.

In contrary to software realizations in which any part of the system can be relatively easily reprogrammed, in case of the transistor level implementation once the system is fabricated any further changes are not possible. For this reason, we developed a software system that enables massive computations for many parameters usually not seen in other models. The system has been written in ANSI C++ language under the Linux OS. It does not need a graphical shell, so it can be compiled and run under most Unix-type systems. The realized model is strongly oriented on hardware realizations, so the outcomes of the simulations can easily be ported to any low level hardware, including full-custom ASIC designs.

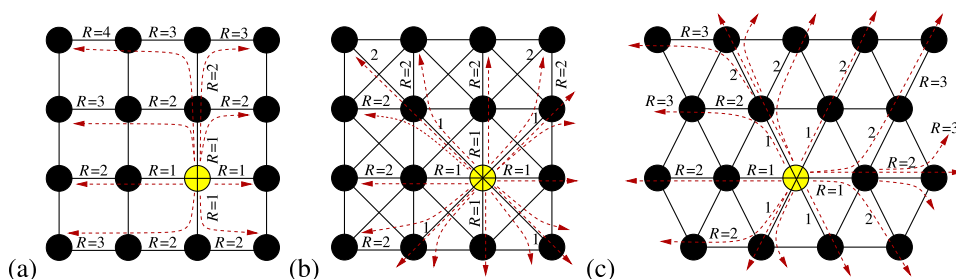


Fig. 1. Typical SOM topologies: (a) and (b) rectangular with 4 (Rect4) and 8 (Rect8) neighbors, (c) hexagonal (Hex).

Download English Version:

<https://daneshyari.com/en/article/6420279>

Download Persian Version:

<https://daneshyari.com/article/6420279>

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