A Cellular Network Architecture With Polynomial Weight Functions

Abstract:

Emulations of cellular nonlinear networks on digital reconfigurable hardware are renowned for an efficient computation of massive data, exceeding the accuracy and flexibility of full-custom designs. In this contribution, a digital implementation with polynomial coupling weight functions is proposed for the first time, establishing novel fields of application, e.g., in the medical signal processing and in the solution of partial differential equations. We present an architecture that is capable of processing large-scale networks with a high degree of parallelism, implemented on state-of-the-art field-programmable gate arrays. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

Enhancement of the project:

Existing System:

The precision of analog implementations is usually not sufficient for numerically sophisticated applications. Moreover, the design of these application specific integrated circuits (ASICs) is generally fixed and parameters like network size or data precision cannot be adjusted. Thus, the emulation of CNNs on reconfigurable digital devices, especially on field-programmable gate arrays (FPGAs), becomes attractive for prototyping and applications where flexibility and/or higher precision is required.

We present an advancement of the recently introduced NERO architecture to the TITUS architecture for the digital emulation of PTCNNs with an arbitrary polynomial order.

PTCNN MODEL

A CNN is a regular system of processing elements (PEs) (cells) that are coupled to its neighbors in lateral and diagonal directions. In a common 1-neighborhood each cell is hence coupled to eight neighbors and to itself (also called 3 × 3 neighborhood). In a PTCNN, the couplings
between the neighboring cells are represented by polynomial weight functions. Since a standard model has not yet been defined, we exclusively refer to feedback and feed-forward terms of polynomial order $P \in \mathbb{N}^*$. 

Using the forward Euler integration method, we obtain the discretetime state equation of a PTCNN

$$
x_{ij}(n+1) = \mathcal{N}\left(x_{ij}(n) + \sum_{|k| \leq 1} \sum_{p=1}^{P} \hat{a}_{kl}^{(p)} x_{i+k,j+l}^{p}(n) + \hat{w}_{ij}\right)
$$

(1)

Disadvantages:

- System efficiency is less

Proposed System:

NERO Architecture

A direct mapping of the cellular network structure to digital hardware is feasible and efficient, yet strongly limited by the available resources. For that reason, we designed the NERO architecture by mapping large-scale networks to medium- or small-scale processor arrays, yet retaining the local couplings of the CNN paradigm and minimizing the number and length of interconnections between the PEs.
Fig. 1. Array architecture with a row of locally coupled PEs and additional PE dummies (D).

In this architecture, the network is divided into vertical segments that are treated line-by-line and segment-by-segment by a row of npe PEs. Each PE is connected to its left and right neighbors, or to a dummy PE at either ends of the row, respectively. Fig. 1 shows the structure of the 1-D processor array with the main control and data paths. For an optimal hardware utilization (Section IV-A), two PEs share a common memory unit (mem) and a PE control unit. Both the PEs and the mems are coupled only locally to moderate the routing complexity. The left and right PE dummies (D) have no calculation core and are required only to provide data from the borders to the neighboring segments, or to ensure the network boundary conditions, respectively.

**TITUS—Extension for PTCNN**

The modifications to NERO to process the PTCNN state equation (1) comprise the calculation core of the PE, the local controller unit, and the template FIFO. The novel architecture of the calculation core will be explained subsequently.

The structure of the extended calculation core is shown in Fig. 2. It comprises three DSPs for the left, central, and right column, each processing the contribution of three neighboring cells to the state equation. Each DSP (shown in Fig. 3) calculates three elements of the Horner scheme with $m_0 = 0$. 

Copyright © 2016 LeMenizInfotech. All rights reserved
The Master of IEEE Projects

\[ m_{k+1} = x \cdot (a + m_k) \]

(2)

The number of iterations of (2) depends on the selected polynomial order. The results are accumulated, added to \( \hat{w}_{ij} \), and finally constrained to the range \([-1; 1]\) with the operator \( N \).

Advantages:

- System efficiency is less

Software implementation:

- Modelsim
- Xilinx ISE