

## Reconfigurable Neural Network Using DAP/DNA

Yu, Yunqing

Department for Informatics Graduate School of Information Science and Electrical Engineering,  
Kyushu University

Murakami, Kazuaki

Department for Informatics Graduate School of Information Science and Electrical Engineering,  
Kyushu University

<https://hdl.handle.net/2324/6794474>

---

出版情報 : The 7th International Conference on High Performance Computing and Grid in Asia  
Pacific Region, pp.432-433, 2004-07

バージョン :

権利関係 :



# Reconfigurable Neural Network Using DAP/DNA

Yunqing YU, Kazuaki MURAKAMI

Department for Informatics

Graduate School of Information Science and Electrical Engineering, Kyushu University

6-1 Kasuga-koen, Kasuga, Fukuoka 816-8580, Japan

E-mail: yu@fleets.jp

**Abstract** In this paper, we want to find ways of realizing a neural network in DAP/DNA. As convenient platform for experiments the DAP/DNA were taken, which allows the change of hardware in one clock per application, flexibly and dynamically. Implementing neural network by using this method, high computational power will be provided, and flexibility and scalability are hold.

## 1. INTRODUCTION

Neural networks have been developed with practical applications of field such as image and speech signal processing. Building neural network in digital hardware is well known. In order to increase high network performance, massively parallel VLSI implementations are used, all connections of a neuron to other neurons are calculated at the same time and a neuron needs all synaptic signals at the same time. So it is inflexible, because a design can only be used for one specific implementation, for a different size of backpropagation network the architecture of a neuron has to be changed. [1]

As a solution to the above problems, reconfigurable neural network using DAP/DNA (Digital Application Processor/Distributed Network Architecture) was examined. The DAP/DNA is a high performance reconfigurable processor with an absolutely new concept which allows the change of hardware in one clock per application, flexibly and dynamically. Implement neural networks to DAP/DNA, a process flow is controlled by DAP and hardware realization of the complicated calculation is carried out by DNA. The higher processing speed will be realizable because the part of calculation implemented by hardware. The size of networks and the process flow of networks training are implemented by software. So for different practical applications, implementing

becomes easy, flexible and scalable.

## 2. NEURAL NETWORK

Different types of neural networks were considered, the error backpropagation network [2] was chosen for hardware implementation in this paper.

An error backpropagation network has an input layer of neurons, which don't perform any computations but serve as data buffers, then one or more hidden layers and an output layer. The neurons of one layer have connections to all neurons of the previous and next layer.

Each neuron contains a number of synapses, which multiply each neuron input by a weight value. The weighted inputs are accumulated and passed through a nonlinear activation function.

The backpropagation algorithm is used for training multi-layer neural network. It iteratively computes the values of weights using a gradient algorithm as following.

1. Network initialization.

All weights are initialized to small random numbers.

2. Forward propagation.

An input vector is presented and propagated layerwise through the network.

3. Output error computation.

4. Backward propagation.

The output error signal is back-propagated through the network. This process allows to assign errors to hidden neurons.

5. Weight update.

Previously computed errors (stages 3 and 4) and neuron activations determine the weight changes.

This training process (step 2 to step 5) is repeated until the output error signal fall below a predetermined threshold.

### 3. IMPLEMENTATION

#### 3.1. Reconfigurable DAP/DNA

DAP/DNA was developed by IPFlex Company. [3] IPFlex has developed "DAP/DNA dynamic reconfigurable processor technology" to achieve "Software to Silicon" that enables a system described in software (C language, MATLAB/Simulink, etc) to be directly implemented on device with a performance equivalent to a custom chip.

The DAP RISC core controls the processor's dynamic reconfiguration, while portions of an application that require high-speed processing are handled by the DNA matrix, which provides both parallel and pipelined processing.

The DNA matrix is an array of 376 Processing Elements (PE) which comprised of computation units, memory, synchronizers, and counters. The DNA matrix circuitry can be reconfigured freely into the structure that is most optimal for meeting the needs of the application in demand.

#### 3.2. DAP/DNA Design Flow

The application performed by DAP/DNA can be divided into two parts: the one operated with DAP RISC core using a performance analyzer, and the other performed by the DNA-Matrix.

According to DAP/DNA design flow, the backpropagation algorithm are described by language C as user application.

Then performance analysis is carried out. Performance analysis can be performed and a bottleneck portion can be specified.

By making DNA process the specified bottleneck portion, an application performance can be raised dramatically.

#### 3.3 Implementation in DNA

In order to increase processing speed, we implement the following calculation to DNA in this paper.

##### 1. Output of a neuron

Typically, sigmoid functions can be employed here. So the output value of a neuron can be calculated by following function.

$$y = f(\alpha) = \frac{1}{1 + e^{-\alpha}} \quad \alpha = \sum_{i=0}^I x_i w_i \quad (1)$$

Here, sigmoid function is expanded by Taylor expansion, and calculated by add and multiplication.

##### 2. Update weight.

$$w_{new} \leftarrow w_{old} + \eta \Delta w \quad (2)$$

For hidden layer

$$\Delta w_{ij} = -y_j (1 - y_j) x_i \sum_k \delta_k w_{jk} \quad (3)$$

$$\delta_k = y_k (1 - y_k) (t_f - y_k)$$

For output layer

$$\Delta w_{jk} = -y_k (1 - y_k) (t_f - y_k) y_j \quad (4)$$

Here,

$w_{ij}, w_{jk}$  : weight parameters,

$\eta$  : coefficient,

$t_f$  : desired output,

$x_i$  : input signal,

$y_j, y_k$  : output signal of hidden layer and output layer, respectively.

So, function (1) (2) (3) (4) can be performed using multiplier and adding machine. It is easy to implementing design by using DNA blockset, because there are adding machines and multipliers block in DNA.

Moreover, although output values of neurons are calculated one by one, the computational power is improved since DNA provides pipelined processing.

### 4. CONCLUSION

In this paper, reconfigurable neural network using DAP/DNA was discussed. According to the feature of DAP/DNA, the network size and backpropagation algorithm process flow are controlled by DAP, the output calculation of a neuron and update weight are implemented by DNA. The advantages of the design method are following.

The first is that hardware design becomes easy.

Then, fast calculation capability, leading to fast update of the neural network. By making DNA process the specified calculation portion, high processing speed will be realizable.

Furthermore, flexibility and scalability are hold. Because different network architectures are possible, furthermore, as the proposed hardware architecture can be used for arbitrary large networks, considered by the number of neurons in one or more layer.

#### References

- [1] Henriette Ossoinig, Erwin Reisinger, Christian Steger, Reinhold Weiss: "Design and FPGA Implementation of a Neural Network"
- [2] Jean-Luc Beuchat, Jacques-Olivier Haenni and Eduardo Sanchez: "Hardware Reconfigurable Neural Networks"
- [3] <http://www.ipflex.com>