A neuron computer model embedded Lukasiewicz' implication

Kenji KOBATA*, Hanxi ZHU, Tomoo AOYAMA**, and Ikuo YOSHIHARA

The Faculty of Engineering, Miyazaki University
Gakuen Kibanadai-nishi 1-1, Miyazaki 889-2192, Japan
*E-mail: ttb236u@student.miyazaki-u.ac.jp
**E-mail: t0b217u@cc.miyazaki-u.ac.jp

Abstract

Many researchers have studied architectures for non-Neumann's computers because of escaping its bottleneck. To avoid the bottleneck, a neuron-based computer has been developed. The computer has only neurons and their connections, which are constructed of the learning. But still it has information processing facilities, and at the same time, it is like as a simplified brain to make inference; it is called "neuron-computer".

No instructions are considered in any neural network usually; however, to complete complex processing on restricted computing resources, the processing must be reduced to primitive actions. Therefore, we introduce the instructions to the neuron-computer, in which the most important function is implications. There is an implication represented by binary-operators, but general implications for multi-value or fuzzy logics can't be done. Therefore, we need to use Lukasiewicz' operator at least. We investigated a neuron-computer having instructions for general implications. If we use the computer, the effective inferences base on multi-value logic is executed rapidly in a small logical unit.

Keywords: *neural network, multi-valued logic, Lukasiewicz' implication, neuron-computer, neuron-CPU*

1.Introduction

Many researchers have studied architectures for non-Neumann's computers because of escaping its bottleneck. We believe that a neuron system has a possibility to realize the architecture. We suppose that the system includes neural networks and connections among the networks, and has a control part to grip over them. Since the system seems to be a kind of the computer, it is often called "neuron-computer". Many variations for the neuron computers have been considered [1]; however, since we are sure that the design must be simplified, wish to realize it by using multi-layer neural networks and their connections. In order to minimize hardware, complex processing must be reduced to primitive ones; then, specifying the order of the primitive processing is necessary. We introduce instructions for the requirement. Thus, the most important point is to design a central processing unit (CPU) driven by the instructions, which is constructed of neural network only.

There are many studies that the neural network is constructed of CPUs [2,3], however, its reverse problem, a CPU is constructed of neural networks, is hardly found The neural networks have many useful characters that are directly implemented by CPUs executed on the binary logic [4], and moreover, the neural networks have an association-memory function [5]. The memory has consistency and compensation ability for partial lack in the memory. The analogue and multi-valued logic variables are used in the neural networks. These are plus factors inherited from original neuron systems.

2. Inference Machines

2.1 Operations

The multi-valued logic extends to the fuzzy logic directly. The final target is the fuzzy inference machine based on the neural networks; it is desirable that a processing system such as the brain is designed. However, even in the field of finite multi-valued logic, there are many logic formulas that cannot be represented by the binary logic. In the evaluation of a plain fuzzy inference, modus-ponens (A→B) & A'→ B', the calculation for implication "→" is required [6]. Where A and B are logical variables or fuzzy membership functions. Therefore, the implication operator would be primitive arithmetic for the inference machine.

Many kinds of the implication operators can be defined, however, the effective operators in the practical usage are Kleene's, Lukasiewicz', Zadeh's, and Goedel's operators [7].

Although Kleene's implication operator can be represented by AND, OR, and NOT operators on multi-value logic, as well as binary logic, the inference is not tautology. Hereafter, we call AND, OR, and NOT operators as binary operators. Therefore, we need to use Lukasiewicz's implication at least; however, the operator is not symmetric and can't be represented by the binary operators. The Lukasiewicz' implication operator is represented by a truth table or expression [6]:

$A \rightarrow B = \min(1, 1-A+B)$.

They can be calculated by quoting tables; however, there is no cortex corresponded to the table in the brain; moreover, the human being acts as the reflex movement. The inference would not be the redundant action such as quoting tables. To design the direct inference, we considered a neural network that has instructions for the implications.

No instructions are considered in any neural network usually; however, reducing complex processing to primitive operations, we introduce input signals for the operations in the neural network, which is similar to instructions in the Neumann's computer. This is a production of CPU by using neural networks and