Proceedings of the 14th
KACC, Octorber 1999

Functional memories constructed of neural network

Hanxi ZHU, Tomoo AOYAMA, and Ikuo YOSHIHARA
The Faculty of Engineering, Miyazaki University
Gakuen Kibanadai-nishi 1-1, Miyazaki 889-2192,Japan
Aoyama@esl.miyazaki-u.ac.jp

Abstract:

Anyone observes that information processing in animal brains is depended on neural networks. On the other hand, engineering models for the neural networks are well known now, and they have been studied, and learning facility is found in the model. We are sure there is a potential in order to create a non Neuman-machine in the engineering models. We studied iteration forms including the engineering neural network models, taking a first step for the creation.

1. Introduction:

It is well known that some kinds of neural networks have studied and they have functions of information processing. It is impressed that complex information processing is developed in simple network structures and primitive neuron functions. The neural networks were engineering models in order to simulate animal brains. However nowadays, we believe that a root of the processing in the brain is discovered in investigations of the neural networks, and the root is related with memory functions. So, we study many kinds of storage actions for neural networks in this manuscript.

We are sure that there are following types of storage functions in information processing.

(1) addressed memory

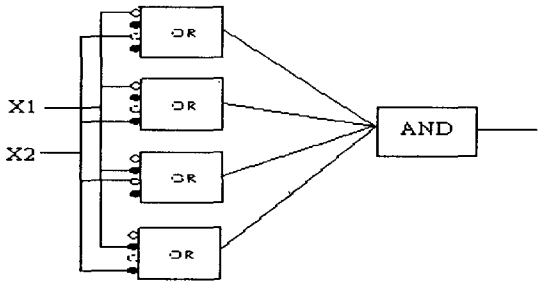
(2) pipeline memory(ex. first-in first-out, first-in last-out memory, or vector register)

(3) associated memory

They are difference in each classification and in functions, but they should be constructed of using neurons and networks. If they can't be done, it does not come to conclusions that investigations of the neural networks can approach to the root.

2. Iteration including the neural network

There is a neural network model constructed of multi layer neurons [1], which is called as multi layer or perception type neural network, and which has a function that translates vectors into different ones. Numbers of elements between the former and latter vectors are different. The translation is plural former vectors to one latter vector correspondingly, and is never one former to plural latter. The former vectors are called as input vectors or data, and the latter are teaching vectors or data. Their genetic name is learning data. Behavior of the multi layer neural network is forward propagation orderly for information represented by vectors. The property is important, for example, any binary logic functions (F) are represented as disjunctive normal forms: F=A1|A2|A3|....|An, where Ai(i=1,2,..) is maximum product term, and "|" is the OR-operator. The product term is constructed of logical variables and their not-operations. Therefore, following a network can simulate any binary logic.



Where " ° " is NOT operator. NOT, OR and AND functions are simulated by neural networks. A differential form (∂eaching data/ ∂nput data) for the neural networks is known [2]. Therefore, the above binding coefficients between the neural networks are also calculated by an iteration similar to the back-propagation algorithm. We believe that the disjunctive concept is extended to multi-valued logic.

The network has not any function to memorize, but if a feedback loop is added to the network, a memory function is shown. Information propagation along the loop is just as if iterations for dynamical systems. The iterations are represented as following,

$$Y=NN(X),$$

$$X=\{Zn,id,D\},\quad Y=\{Zn+1,Q\} \qquad (1)$$

where X is the input data for the neural network, Y is the teaching data, and Q is output value of the neural network. Y includes Zn+1 that is input data for new iteration.

X includes Zn to be found on the feedback loop. D is input data for NN function, and id is description operators for actions of NN. Zn is one kind of status.

The actions are written as,

(1)id="keep": Zn is kept.

(2)id="set": D is set on Zn+1.

Actions of the above iteration (1) are same that of D-type flip-flop. A true value table

derived from eq. (1) is listed in Table 1, which can be learned on forward neural networks.

TABLE1

INPUT			OUTPUT	
Zn	D	id	Q	Zn+1
a	b	0	a	a
a	b	1	b	b

Where Roman letters {a,b} are genetic symbols of values {0,1}. Moreover, a feedback loop is added the network, Zn+1 is put into Zn on input side at next iteration cycle, therefore a memorized function same as D-type flip-flop is realized. That is 1 bit memory can be composed of one neural network with one feedback loop.

We consider that above memory generations may be occurred to a large scaled neural system. If there are many such memory neural networks, they should be distinguished each other and be labeled. The labeling will be called as "addressing". We believe in there are some addressing memories in a neural network system.

3. First-in first-out memory

A first-in first-out memory (FIFO) for 2 bits is represented as TABLE2,

TABLE2

INPUT				OUTPUT			
Zn{0}	{1}	D	id	Q	Zn+1{0}	{1}	id-next
a	b	C	0	b	a	b	0
a	b	c	1	b	a	b	1

where there are two feedback loops {0} and {1}. The id-next signal is used to make 4 bits FIFO out of 2 bits FIFO. The generation of double bits FIFO is recurred again.

4. Vector register

Vector registers have been adopted in some kinds of super computer. By using the registers, data processing are executed as pipeline streams, then as a result, high speed processing is realized. We believe that such a pipeline stream is observed in animal brains, and if the assumption is true, vector register will be also constructed of neural networks. For a confirmation of the construction, we study iterations of truth value tables. The iterations represent all actions of the vector registers, which include up down-counter and first-in-first-out memory and suppress-request-logic. There is no inevitability of the construction, of course, anyone can construct a vector register as sets of many above components. We are also sure that the latter method is easy, but although there is one neural network, we desire that data processing is enabled such as a vector computing.

We got a following truth value table as two words vector register.

TABLE3

INPUT								OUTPUT				
M0	M1	D	w	r	rs	wc	rc	Q	wc	rc	M0	M1	action
a	b	c	d	e	1	f	g	*	0	0	*	*	* initial reset
*	*	a	0	0	0	0	0	*	0	0	*	*	* initial hold
*	*	a	1	0	0	0	0	*	1	0	a	*	*write1
a	*	b	0	0	0	0	0	*	1	0	a	*	*write1 hold
a	*	b	1	0	0	1	0	*	2	0	b	*	awrite2
b	a	c	0	0	0	2	0	*	2	0	b	*	awrite2 hold
a	*	b	0	1	0	1	0	a	1	1	a	*	*write1-read
a	*	b	0	0	0	1	1	*	1	1	a	*	*hold
a	*	b	0	0	2	1	1	*	1	0	a	*	*rc-reset
a	*	b	1	0	0	1	1	*	2	1	b	*	awrite1 read
b	a	c	0	1	0	2	0	a	2	1	b	*	awrite2 read
b	a	c	0	0	0	2	1	*	2	1	b	*	ahold
b	a	c	0	1	0	2	1	b	2	2	b	*	awrite2 read
b	a	c	0	0	0	2	2	*	2	2	b	*	ahold
b	a	c	0	0	2	2	2	*	2	0	b	*	arc-reset

Where INPUT and OUTPUT mean input- and output-sides of a neural network, respectively. D, w, r, rs, and Q are input-, write-request, read-request, reset-, and output signals respectively. M0 and M1 are memory parts for 1 bit. Symbols {0,1,2} are numerical, and Roman letters {a,b,c,...} are genetic symbols from numerical ones, therefore, "a" means {0,1}. Letter "*" is used for meaningless value. In practical calculations, we make "*" to be 0.5.

In the table 3, there are some multi-valued truth values, for example, signal "rs" has three different values {0,1,2}. The values are scaled into {0,0.5,1} in a neural network. We are sure that existence of the multi-valued signals is a superior ability of the neural network. In case of using multi-valued signals, maximum permissible doses of simulation-errors become smaller. We consider the limitations of multi-valued to be 256, which is got from experience of many simulations. The digitized logic must be used in order to cancel the errors at reasonable intervals. As we took precautions against the errors, the digitizing is executed after one neural computation, at every time.

The above table represents one bit actions in two words vector register. We calculated one multi-layered neural network by using back-propagation method from the table. And on use of an analogue-digital-converter and the reverse one, we tested the actions of the neural vector register for an analogue data stream. The simulation list is as following.

(1)initial reset

D=0.987654, w=0, r=0, rs=0.500000, Q=0.996094

M0: 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50

M1: 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50

(2)write1 test

D=0.888889, w=1, r=0, rs=0.000000, Q=0.996094

M0: 1.00 1.00 1.00 0.00 0.00 1.00 0.00 0.00

M1: 0.50 0.50 0.50 0.50 0.50 0.50 0.50 0.50

(3)write2 test

D=0.777778, w=1, r=0, rs=0.000000, Q=0.996094

M0: 1.00 1.00 0.00 0.00 0.00 1.00 1.00 1.00

M1: 1.00 1.00 1.00 0.00 0.00 1.00 0.00 0.00

(4)read1 test

D=0.555556, w=0, r=1, rs=0.000000, Q=0.890625

M0: 1.00 1.00 0.00 0.00 0.00 1.00 1.00 1.00

M1: 1.00 1.00 1.00 0.00 0.00 1.00 0.00 0.00

(5)read2 test

D=0.444444, w=0, r=1, rs=0.000000, Q=0.777344

(6)read counter reset

D=0.333333, w=0, r=0, rs=1.000000, Q=0.996094

(7)read1 test

D=0.000000, w=0, r=1, rs=0.000000, Q=0.890625

(8)read2 test

D=0.000000, w=0, r=1, rs=0.000000, Q=0.777344

Where D, w, r, rs, and Q are input-, write-request, read-request, reset-, and output signals respectively. D is an analogue signal, but Q is a pseudo-analogue one by means of analogue-digital-conversion and its reverse. M0 and M1 are memory parts that have 8 banks. The

list means that:

(1)M0 and M1 are initialized by actions of initial reset.

(2)Data(0.888889) on D are converted and stored in M0 by write1.

(3)Data on M0 are moved to M1 and Data(0.777778) on D are converted and stored in M0 by write2.

(4)By read1 test, Data stored on M1 are quoted and set on Q. By the conversion, value 0.888889 is changed to 0.89094. Slightly digitized error is found, and input data on D is canceled.

(5)By read2 test, Data stored on M0 are read.

(6)Read counter is reset. The action makes reading for M-part possible.

(7)Reading for M0.

(8)Reading for M1.

These responses show that actions of a vector register are reasonable, and The vector register can be constructed by a neural network.

5. Conclusion

We have discussed some memory functions represented by neural networks. The functions are equivalent to that of binary logic units. Since there are many kinds of memory in network system like as the brain, we are sure that the discussions are important. Moreover, neural networks can express any multi-value logic functions which are not done by the binary sufficiently. We believe that neural network representations of disjunctive normal form for the multi-value logic are well worth consideration.

References

[1]D.E.Rumelhart, J.L.McCleland and the PDP Research Group, "Parallel distributed processing", MIT Press(1986).

[2]T.Aoyama and H.Ichikawa,"Backpropagation Algorithm Restricted with some conditions", p.59, IPSJ SIG Notes 91-NA-37(1991).

[3]T.Aoyama, "Large scale multi layer neural networks", p.25, IPSJ SIG Notes 99-HPC-76(1999).