Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

The most robust design for digital logics of multiple variables based on neurons with complex-valued weights

Wei-feng LÜ†[1,2], Mi LIN[2], Ling-ling SUN[1,2]
([1]*Institute of VLSI Design, Zhejiang University, Hangzhou 310027, China*)
([2]*College of Electronics and Information, Hangzhou Dianzi University, Hangzhou 310018, China*)
†E-mail: lvwf@hdu.edu.cn

Received Mar. 30, 2008; Revision accepted Aug. 30, 2008; Crosschecked Dec. 26, 2008

Abstract: Neurons with complex-valued weights have stronger capability because of their multi-valued threshold logic. Neurons with such features may be suitable for solution of different kinds of problems including associative memory, image recognition and digital logical mapping. In this paper, robustness or tolerance is introduced and newly defined for this kind of neuron according to both their mathematical model and the perceptron neuron's definition of robustness. Also, the most robust design for basic digital logics of multiple variables is proposed based on these robust neurons. Our proof procedure shows that, in robust design each weight only takes the value of i or −i, while the value of threshold is with respect to the number of variables. The results demonstrate the validity and simplicity of using robust neurons for realizing arbitrary digital logical functions.

Key words: Complex-valued weights, Multi-valued neurons (MVNs), Digital logic, Robust design
doi:10.1631/jzus.A0820238 **Document code:** A **CLC number:** TP183; TN431

INTRODUCTION

Neurons with complex-valued weights [also called multi-valued neurons (MVNs)], were introduced by Aizenberg and Aizenberg (1994), based on the ideas of multi-valued threshold logic. The application of different kinds of networks based on MVNs to associative memory, image recognition, segmentation, time-series prediction and signal processing confirms their high efficiency (Jankowski *et al*., 1996; Aizenberg N. *et al.*, 1996; Aizenberg I. *et al.*, 2000; Goh and Mandic, 2005; Michel *et al.*, 2006; Kalra *et al.*, 2007). Many papers focused on these applications because solution of these problems using neural networks has been very popular in recent years. Also, the MVNs based neural network with random connections has been proposed as an alternative to the Hopfield network (Muezzinoglu *et al.*, 2003; Nishikawa *et al.*, 2005; 2006; 2007; Michel *et al.*, 2006; Kalra *et al.*, 2007). The validity and efficiency of neurons basically depend on their ability to implement digital logics or Boolean expressions, and many papers have discussed this problem in relation to other neurons (Gray and Miched, 1992; Hundiwal, 1993; Zhang and Xu, 1996). However, most of these papers have not taken this into consideration in detail even though it is of great importance for MVNs. Moreover, robustness or tolerance should also be considered because such a feature is an essential property for a neuron. However, there are no publications proposing a precise definition of robustness in relation to MVNs or a robust design for digital logics. In this paper, we attempt to address these deficiencies.

MODEL OF NEURONS WITH COMPLEX-VALUED WEIGHTS AND THEIR LEARNING

Neurons with complex-valued weights have been described in many recent papers and some important theories have been presented. Here, we summarise some of the key aspects of MVN theory including mathematical models and their learning (Aizenberg and Aizenberg, 1994; Aizenberg N. *et al.*, 1996). The basic idea of a neuron is the representation of the Boolean function or the k-valued function of n variables by n+1 complex numbers $w_0, w_1, \ldots, w_n$:

$$f(x_1, x_2, \ldots, x_n)=P(w_0+w_1x_1+\ldots+w_nx_n), \tag{1}$$

where $x_1, x_2, \ldots, x_n$ are variables; P is the output (active) function of the neuron, which is defined by the following Eq.(2) or Eq.(3).

(1) In the case of MVNs,

$$P(z)=\exp(\mathrm{i}2\pi j/k),\ \text{if } 2\pi(j+1)/k>\arg z>2\pi j/k, \tag{2}$$

where $j\in\{0, 1, \ldots, k-1\}$ are values of k-valued logic; i is an imaginary unit; arg$z$ is the argument of the complex number z and $z=w_0+w_1x_1+\ldots+w_nx_n$ (weighted sum). Here, values of function and variables are coded by the kth power roots of a unit $\varepsilon^j=\exp(\mathrm{i}2\pi j/k)$; i.e., the values of k-valued logic are represented as the kth power roots of a unit: $j\rightarrow\varepsilon^j$. Fig.1 shows such a neuron.

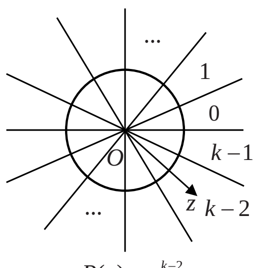

Fig.1 Multi-valued neuron and its active function

(2) In the case of universal binary neurons (UBNs),

$$P(z)=(-1)^j,\ \text{if } 2\pi(j+1)/m>\arg z\geq 2\pi j/m, \tag{3}$$

where m is a positive integer, $0<m\leq 3n$, and j is a non-negative integer, $0\leq j<m$. Variables and functions in this case are Boolean and take values from the set $\{-1, 1\}$ after being coded.

We know from the mathematical model above that the learning of such a neuron devotes the weighted sum of inputs to moving correct section in the complex plane. The learning algorithm was expressed as follows (Aizenberg and Aizenberg, 1994):

$$\boldsymbol{W}_{m+1}=\boldsymbol{W}_m+\omega\varepsilon^q\bar{\boldsymbol{X}}, \tag{4}$$

where $\boldsymbol{W}_m$ and $\boldsymbol{W}_{m+1}$ are the current and the next weight vectors, respectively; ω is the correction coefficient; $\bar{\boldsymbol{X}}$ is a vector of the neuron's input signals with a complex-conjugated component; ε^q are values of the neuron's output, and q is the corresponding section of active function definition ($q=k-2$, Fig.1). For better application to image processing, another learning algorithm was described by Aizenberg I. *et al.*(2000) as follows:

$$\boldsymbol{W}_{m+1}=\boldsymbol{W}_m+\frac{C_m}{n+1}(\varepsilon^q-\varepsilon^s)\bar{\boldsymbol{X}}, \tag{5}$$

where C_m is a coefficient to be determined, and s is the current section.

The definitions of the MVN activation function and its learning algorithm have some wonderful properties that make MVNs much more powerful than traditional artificial neurons (Aizenberg I. *et al.*, 2000; Goh and Mandic, 2005; Michel *et al.*, 2006; Kalra *et al.*, 2007).

ROBUST DESIGN FOR DIGITAL LOGIC BASED ON MVNS

Definition of robustness on MVNs

Zhang and Xu (1996) defined the robustness of perceptron. Suppose the colors red and blue correspond to logic "1" and logic "0", respectively, and red vertex $\boldsymbol{X}^{(1)}=[x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}]$ and blue vertex $\boldsymbol{X}^{(2)}=[x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)}]$ are adjacent in an n-dimensional hypercube. If the classification hyperplane $A_1x_1+A_2x_2+\ldots+A_nx_n+\theta=0$, which classifies the following two vertices:

$$\begin{cases} A_1x_1^{(1)}+A_2x_2^{(1)}+\cdots+A_nx_n^{(1)}+\theta=1/2, \\ A_1x_1^{(2)}+A_2x_2^{(2)}+\cdots+A_nx_n^{(2)}+\theta=-1/2, \end{cases}$$

we call the neuron defined by

$$y=\mathrm{sgn}(A_1x_1+A_2x_2+\cdots+A_nx_n+\theta) \\ =\begin{cases} 1, & A_1x_1+A_2x_2+\cdots+A_nx_n+\theta\geq 0, \\ 0, & A_1x_1+A_2x_2+\cdots+A_nx_n+\theta<0 \end{cases}$$

the most robust logical neuron or simply a robust neuron. A neural network that is composed of only robust neurons is called a robust or tolerant network. Zhang and Xu (1996) described some useful and important results relating to the design of the most robust and sub-robust logical neural network based on perceptrons.

In this work, we study a robust problem similar to the MVN. From Eq.(2), which performs a mapping described by a k-valued function, we have exactly k domains, in which the angle of each is $2\pi/k$. Through the learning rule, MVNs can rectify the weighted sum z from an incorrect sector number to the correct one. For robust design, z must be in the middle of two adjacent classification complex vectors in the complex plane, which means that the angle of z to each adjacent complex vector must be maximal π/k (Fig.2). In the case of the UBN from Eq.(3), the smaller the value of m, the bigger the angle, and the more robust the MVN. If m=2, we have two domains, each of angle π, and the angle of z to each adjacent classification complex vector must be maximal $\pi/2$.

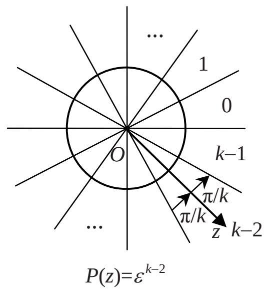

Fig.2 Robust multi-valued neuron and its active function

Now we give our definition of the most robust MVNs for basic digital logic in the case of UBNs. Suppose the colors red and blue correspond to logic "1" and logic "0" respectively, and red vertex $\boldsymbol{X}^{(1)}=[x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}]$ and blue vertex $\boldsymbol{X}^{(2)}=[x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)}]$ are adjacent in an n-dimensional hypercube corresponding to complex vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, respectively. If the classification complex vector satisfies $\arg(B_1x_1+B_2x_2+\ldots+B_nx_n+\theta)=L\pi$, $L\in\{0, 1\}$, which classifies the following two vertices:

$$\begin{cases}\arg(B_1x_1^{(1)}+B_2x_2^{(1)}+\cdots+B_nx_n^{(1)}+\theta)=\pi/2,\\ \arg(B_1x_1^{(2)}+B_2x_2^{(2)}+\cdots+B_nx_n^{(2)}+\theta)=2\pi-\pi/2,\end{cases} \tag{6}$$

we call the neuron defined by

$$y=P(z)=(-1)^j,\ \text{if } (j+1)\pi>\arg z\geq j\pi,\ \ j\in\{0, 1\}$$

the most robust (or tolerant) MVN. Here 2π is used because the defined argument of z should be limited in 0~2π. In this case, the model and active function of the neuron are depicted in Fig.3.

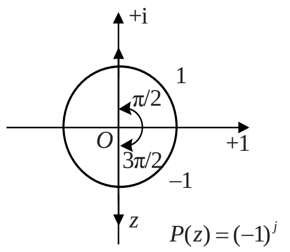

Fig.3 The most robust multi-valued neuron and its active function

Design for digital logics based on the most robust MVN

The basic operations of multiple variables are AND, OR and NOT, which comprise the complete set of digital logic or Boolean function. Let $x_1, x_2, \ldots, x_n$ be n inputs in the set {0, 1}. We transform them as $\{0, 1\}\rightarrow\{1, -1\}$ according to the coded rule.

1. AND operation

For AND operation, if and only if all inputs are −1, the output is −1, otherwise the output is 1. This means that the arguments of output are 3π/2 and π/2 respectively, in accordance with the definition of robustness for MVNs, and their corresponding values are $-\alpha$i and $+\alpha$i (α is a real number), respectively. Now we have the following matrix equation:

$$\begin{bmatrix}1 & 1 & 1 & \cdots & 1\\ 1 & 1 & 1 & \cdots & -1\\ \cdots & \cdots & \cdots & & \cdots\\ 1 & -1 & -1 & \cdots & 1\\ 1 & -1 & -1 & \cdots & -1\end{bmatrix}\begin{bmatrix}w_0\\ w_1\\ \vdots\\ w_{n-1}\\ w_n\end{bmatrix}=\begin{bmatrix}c_0\mathrm{i}\\ c_1\mathrm{i}\\ \vdots\\ c_{2^n-2}\mathrm{i}\\ c_{2^n-1}\mathrm{i}\end{bmatrix}, \tag{7}$$

where $c_0, c_1, \ldots, c_{2^n-2}>0$ and $c_{2^n-1}<0$ are the coefficients to be determined.

It is found that all variables are equivalent and the chances of all n variables being 1 or −1 are all equal too. So the components (except w_0) of the weight vector can all be equal too. We can let all be i since it is the minimum imaginary unit. However, the weight (also called the threshold) w_0 is distinctly different in the weight vector of the above expression. We let it be $(n-1)$i though there are n variables, which means that the output is -1 only if all inputs are -1. Now we obtain the weight vector $\boldsymbol{W}=[(n-1)\mathrm{i}, \mathrm{i}, \ldots, \mathrm{i}]$.

Proof Consider 2^n different states of n variables. We then separate them into two cases: let all n variables be (a) -1; (b) not less than one 1.

In case (a),

$$z=(n-1)\mathrm{i}+\mathrm{i}(-1)+\mathrm{i}(-1)+\ldots+\mathrm{i}(-1)=(n-1)\mathrm{i}+(-n)\mathrm{i}=-\mathrm{i},$$
$$P(z)=(-1)^1=-1.$$

In case (b),

$$z/\mathrm{i}=(n-1)+(-1)+\ldots+1+\ldots+(-1)\geq 1,$$
$$P(z)=(-1)^0=1.$$

For determining the value of c_N $(0\leq N\leq 2^n-1)$, the operation is as follows: first, we code the number N in the decimal system into a number in the binary system; then, transform all binary bits as $\{0, 1\}\rightarrow\{1, -1\}$; finally, we add $n-1$ after getting the sum of all bits of N in the set $\{1, -1\}$. For instance, the coefficient c_{20} is calculated to be 5 and c_{31} to be -1, assuming there are 5 variables.

2. OR operation

For OR operation, the output will be 1 if and only if all inputs are 1, otherwise the output is –1. So we have

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & -1 & -1 & \cdots & 1 \\ 1 & -1 & -1 & \cdots & -1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \\ w_n \end{bmatrix} = \begin{bmatrix} d_0\mathrm{i} \\ d_1\mathrm{i} \\ \vdots \\ d_{2^n-2}\mathrm{i} \\ d_{2^n-1}\mathrm{i} \end{bmatrix}, \quad (8)$$

where $d_0>0$ and $d_1, \ldots, d_{2^n-2}, d_{2^n-1}<0$ are coefficients to be determined. Similarly, the weight components (except w_0) can all be equal too and we also let them have the imaginary unit i. The weight w_0 is different and we let it be $-(n-1)$i though there are n variables, which means that the output is 1 only if all inputs are 1. So we obtain the weight vector $\boldsymbol{W}=[-(n-1)\mathrm{i}, \mathrm{i}, \ldots, \mathrm{i}]$.

Proof There are only two cases: let all n variables be (a) 1; (b) not less than one -1.

In case (a),

$$z=-(n-1)\mathrm{i}+\mathrm{i}+\mathrm{i}+\ldots+\mathrm{i}=-(n-1)\mathrm{i}+n\mathrm{i}=\mathrm{i},$$
$$P(z)=(-1)^0=1.$$

In case (b),

$$z/\mathrm{i}=-(n-1)+1+\ldots+-1+\ldots+1\leq -1,$$
$$P(z)=(-1)^1=-1.$$

For determining the value of d_N $(0\leq N\leq 2^n-1)$, the steps are the same as in the above AND operation except that 'add' is substituted for 'subtract' at the last step. For example, the coefficient d_{20} is -5 and d_0 is 1 assuming there are 5 variables.

3. NOT operation

For NOT operation, the equation is easily found as

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} e_0\mathrm{i} \\ e_1\mathrm{i} \end{bmatrix}, \quad (9)$$

where $e_0<0$ and $e_1>0$ are coefficients to be determined. Also, we find the weight vector $\boldsymbol{W}=[0, -\mathrm{i}]$, and $e_0=-1$, $e_1=1$. Based on the conclusion of the NOT operation above, the weight vector is easily obtained if there is a NOT variable in an AND or OR Boolean expression by letting only the weight of the corresponding variable be $-$i. For example, the weight vectors for most robust design of digital logics $x_1x_2\overline{x_3}x_4$ and $x_1+x_2+\overline{x_3}$ based on MVNs are [3i, i, i, $-$i, i] and [$-$2i, i, i, $-$i], respectively.

Now we integrate all the above results in Table 1. Since the basic operations AND, OR and NOT can be implemented by a single MVN, the arbitrary digital logic function that can be expressed by the above basic operations can certainly be implemented by a network consisting of MVNs. A special advantage of using MVNs rather than other neurons such as perceptron, is that only one neuron is needed for realizing the XOR function (Aizenberg and Aizenberg, 1994) and the MVNs retain good robustness. Our design is also suitable for solution of problems such as the drug therapy interaction warning system (Gray and Miched, 1992).

Table 1 Design rule for basic logics based on MVNs

Basic logic		Weight to input x	Threshold w_0
AND	x	i	$(n-1)$i
	$\overline{x}$	−i	$(n-1)$i
OR	x	i	$-(n-1)$i
	$\overline{x}$	−i	$-(n-1)$i
NOT		−i	0

CONCLUSION

Neurons with complex-valued weights have stronger capability to implement digital logical mapping. We propose and define robustness for this kind of neurons according to their mathematical model and binary perceptron neuron's definition of robustness or tolerance. The most robust design for basic digital logic operations of multiple variables is presented based on our most robust neurons by formation and solution matrix equations. The validity and simplicity of robust neurons consisting of basic digital logics for realizing arbitrary digital logical functions is confirmed. The results can conveniently be applied to similar actual problems. Our ideas can be widely applied to any other new artificial neurons.

References

Aizenberg, I., Aizenberg, N., Butakov, C., Farberov, E., 2000. Image Recognition on the Neural Network Based on Multi-valued Neurons. Proc. 15th Int. Conf. on Pattern Recognition, p.989-992.

Aizenberg, N., Aizenberg, I., 1994. CNN-like Networks Based on Multi-valued and Universal Binary Neurons: Learning and Application to Image Processing. Proc. 3rd IEEE Int. Workshop on Cellular Neural Networks and Their Applications, p.153-158.

Aizenberg, N., Aizenberg, I., Krivosheev, G., 1996. Multi-valued and Universal Binary Neurons: Mathematical Model, Learning, Networks, Application to Image Processing and Pattern Recognition. Proc. 13th Int. Conf. on Pattern Recognition, p.185-189. [doi:10.1109/ICPR.1996.547258]

Goh, S.L., Mandic, D.P., 2005. Nonlinear adaptive prediction of complex-valued signals by complex-valued PRNN. *IEEE Trans. Signal Processing*, **53**(5):1827-1836. [doi:10.1109/TSP.2005.845462]

Gray, D.L., Miched, A., 1992. A training algorithm for binary feedforward neural networks. *IEEE Trans. Neural Networks*, **3**(2):176-194. [doi:10.1109/72.125859]

Hundiwal, A.K., 1993. An Improved Algorithm for Learning Representations in Boolean Neural Networks. Proc. 36th Midwest Symp. on Circuits and Systems, **1**:592-595. [doi:10.1109/MWSCAS.1993.342976]

Jankowski, S., Lozowski, A., Zurada, M., 1996. Complex-valued multi-state neural associative memory. *IEEE Trans. Neural Networks*, **7**(6):1491-1496. [doi:10.1109/72.548176]

Kalra, P.K., Mishra, D., Tyagi, K., 2007. A Novel Complex-valued Counterpropagation Network. IEEE Symp. on Computational Intelligence and Data Mining, p.81-87. [doi:10.1109/CIDM.2007.368856]

Michel, H.E., Awwal, A.A.S., Rancour, D., 2006. Artificial Neural Networks Using Complex Numbers and Phase Encoded Weights—Electronic and Optical Implementations. Proc. Int. Joint Conf. on Neural Networks, p.486-491. [doi:10.1109/IJCNN.2006.1716132]

Muezzinoglu, M.K., Guzelis, C., Zurada, J.M., 2003. A new design method for the complex-valued multistate Hopfield associative memory. *IEEE Trans. Neural Networks*, **14**(4):891-899. [doi:10.1109/TNN.2003.813844]

Nishikawa, I., Sakakibara, K., Iritani, T., Kuroe, Y., 2005. 2 Types of Complex-valued Hopfield Networks and the Application to a Traffic Signal Control. Proc. Int. Joint Conf. on Neural Networks, p.782-787. [doi:10.1109/IJCNN.2005.1555951]

Nishikawa, I., Iritani, T., Sakakibara, K., 2006. Improvements of the Traffic Signal Control by Complex-valued Hopfield Networks. Proc. Int. Joint Conf. on Neural Networks, p.459-464. [doi:10.1109/IJCNN.2006.246717]

Nishikawa, I., Hayashi, K., Sakakibara, K., 2007. Complex-valued Neuron to Describe the Dynamics after Hopf Bifurcation: an Example of CPG Model for a Biped Locomotion. Proc. Int. Joint Conf. on Neural Networks, p.327-332. [doi:10.1109/IJCNN.2007.4370977]

Zhang, J.Y., Xu, J., 1996. Analysis and Design of Most Tolerant Logic Neural Networks. Proc. 3rd Int. Conf. on Signal Processing, **2**:1425-1428. [doi:10.1109/ICSIGP.1996.571124]