



# Solution of Brain Contradiction by Extension Theory

Germano Resconi<sup>1(✉)</sup> and Chunyan Yang<sup>2</sup>

<sup>1</sup> Mathematical and Physical Department,  
Catholic University, Via Trieste 17, Brescia, Italy  
resconi42@gmail.com

<sup>2</sup> Research Institute of Extenics and Innovation Methods,  
Guangdong University of Technology,  
Guangzhou 510006, People's Republic of China

**Abstract.** In fuzzy theory any degree to belong to a set can be considered as a positive distance from complementary set. So the distance moves from zero to one when the object belongs to the set. The extension theory considers a negative value of the distance. This is in conflict with the classical definition of the distance is a positive scalar. So we have a classical contradiction. To solve this conflict we define the distance as a vector with two different directions one positive and the other negative. The distances are vectors with positive norm. In this way we have positive norm for the two directions. In extension theory we define the dependent function and suitable transformations in a way to build a nonlinear neuron that can solve a very old conflicting problem in brain linear neural computation.

**Keywords:** Dependent function · Vector distance · Nonlinear neuron  
Solution of conflicts · Solution of Boolean function by neuron

## 1 Introduction

A new approach to implementation of Boolean function by nonlinear neuron is introduced by the implementation of characteristic nonlinear function introduced for the first time by Cai Wen in the extension theory [1–4]. We show that with the new neuron we can easily compress the tradition AND OR NOT Boolean functions network into one step system to have a more efficient system for brain implementation of complex functions.

## 2 From Vector Distance to the Dependent Functions

In Extenics, one of the dependent functions is given by the expression

$$k(x) = \frac{|x - \frac{a+b}{2}| - \frac{b-a}{2}}{(|x - \frac{a+b}{2}| - \frac{b-a}{2}) - (|x - \frac{a_0+b_0}{2}| - \frac{b_0-a_0}{2})} = \frac{\rho(x, a, b)}{\rho(x, a, b) - \rho(x, a_0, b_0)} \quad (1)$$

For continuous function we transform (1) into (2)

$$k'(x) = \frac{(x - \frac{a+b}{2})^2 - (\frac{b-a}{2})^2}{\left( (x - \frac{a+b}{2})^2 - (\frac{b-a}{2})^2 \right) - \left( (x - \frac{a_0+b_0}{2})^2 - (\frac{b_0-a_0}{2})^2 \right)} = \frac{(x-a)(x-b)}{(x-a)(x-b) - (x-a_0)(x-b_0)} \tag{2}$$

So we have the form

$$V = \begin{bmatrix} k(15) \\ k(20) \\ k(50) \\ k(85) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \text{ and } a = 20, b = 50, a_0 = 15, b_0 = 85$$

We have the Fig. 1.

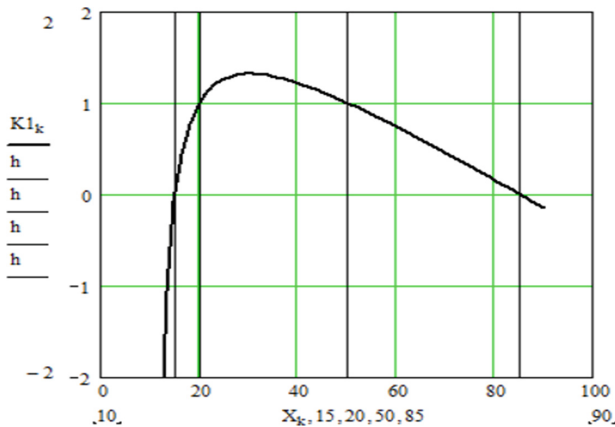


Fig. 1. Graph of the expression (2) with a = 20, b = 50, a<sub>0</sub> = 15, b<sub>0</sub> = 85.

### 3 Neural Solution of Boolean Function by Dependent Function $k'(x)$

For Linear neuron when  $y = 1$  if  $c_1x_1 + c_2x_2 - \theta > 0$  and  $y = 0$  if  $c_1x_1 + c_2x_2 - \theta \leq 0$ . Now given the Boolean function  $(\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) = x_1 XOR x_2 = y$ . We have a contradiction so we cannot obtain the Boolean function XOR by the classical neuron. In fact we have  $c_1 \cdot 0 + c_2 \cdot 0 - \theta \leq 0, c_1 \cdot 0 + c_2 \cdot 1 - \theta > 0, c_1 \cdot 1 + c_2 \cdot 0 - \theta > 0, c_1 \cdot 1 + c_2 \cdot 1 - \theta \leq 0$  and  $-\theta \leq 0$ . True,  $c_2 - \theta > 0$  True if  $c_2 > \theta, c_1 - \theta > 0$  True if  $c_1 > \theta, c_1 + c_2 - \theta \leq 0$  True if  $c_1 + c_2 \leq \theta$ .

Boolean function XOR solved by a nonlinear neuron

$$\begin{bmatrix} X & Y & (\neg X \wedge Y) \vee (X \wedge \neg Y) \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} x & k'(x) \\ 0 & 0 \\ 1 & 1 \\ 2 & 1 \\ 3 & 0 \end{bmatrix}, k'(x) = \frac{(x-0)(x-3)}{(x-0)(x-3) - (x-1)(x-2)} \quad (3)$$

$$\begin{bmatrix} X & Y & X \rightarrow Y \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x & k'(x) \\ 0 & 1 \\ 1 & 0 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}, k'(x) = \frac{(x-1)}{(x-1) - (x-0)(x-2)(x-3)} \quad (4)$$

We remark that for this Boolean function  $k'(x)$  has a singular point. Now because for dependent function, we have the scheme (Figs. 2 and 3). We can change the form of  $k'(x)$  with the same original properties but without the negative value of  $k'(x)$  and also without the singularity. So we have

$$k''(x) = \frac{A(x)^2}{A(x)^2 + B(x)^2} = \frac{(x-2)^2}{(x-2)^2 + [(x-0)(x-1)(x-3)]^2} \quad (5)$$

For the Boolean function and dependent function we can create the scheme for a nonlinear neuron by which we solve in one step and without linear hidden neurons the old problem of the learning process in Brain by AND, OR, NOT logic functions (Fig. 4).

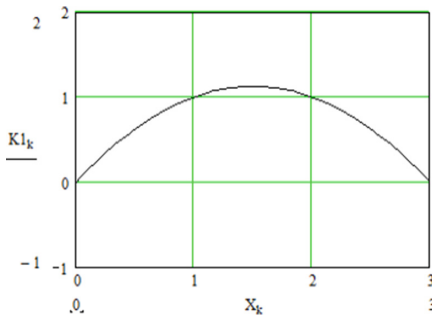


Fig. 2. Graph of the dependent function (3)

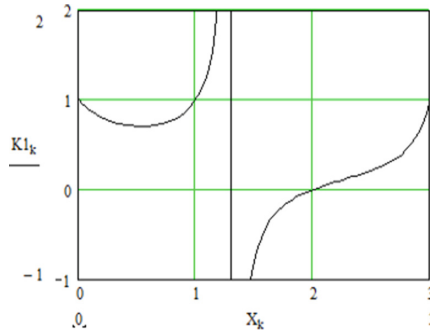


Fig. 3. Graph of the dependent function (4)

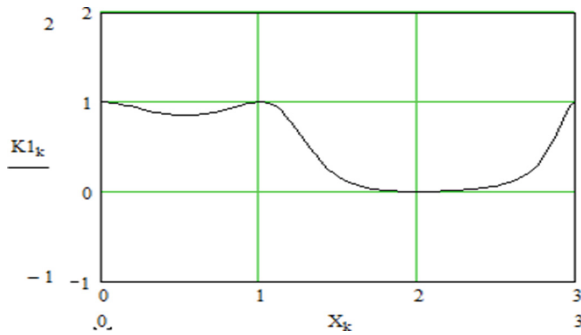


Fig. 4. Graph of the dependent function (5)

### 4 Machine and Systems by Nonlinear Neuron

Given the machine with  $x$  the input,  $q$  the states and  $y$  the output the system can be represented by the Boolean equations (Figs. 5 and 6). Given the machine with  $x$  the input,  $q$  the states and  $y$  the output

$$\begin{bmatrix} q \backslash x & 0 & 1 & y \\ 1 & 3 & 6 & 0 \\ 2 & 3 & 4 & 1 \\ 3 & 2 & 5 & 0 \\ 4 & 5 & 2 & 0 \\ 5 & 6 & 3 & 0 \\ 6 & 5 & 1 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} q \backslash x & 0 & 1 & y \\ 000 & 100 & 110 & 0 \\ 010 & 100 & 001 & 1 \\ 100 & 010 & 111 & 0 \\ 001 & 111 & 010 & 0 \\ 111 & 110 & 100 & 0 \\ 110 & 111 & 000 & 0 \end{bmatrix}, \text{ by code } \begin{bmatrix} 1 \rightarrow 000 \\ 2 \rightarrow 010 \\ 3 \rightarrow 100 \\ 4 \rightarrow 001 \\ 5 \rightarrow 111 \\ 6 \rightarrow 110 \end{bmatrix},$$



$$\begin{aligned}
 q_1(t+1) &= k_1''(x) = \frac{[(x-1)(x-10)(x-12)(x-11)]^2}{[(x-1)(x-10)(x-12)(x-11)]^2 + [(x-0)(x-2)(x-4)(x-7)(x-3)(x-8)(x-9)(x-15)]^2} \\
 q_2(t+1) &= k_2''(x) = \frac{[(x-0)(x-2)(x-10)(x-15)(x-11)]^2}{[(x-0)(x-2)(x-10)(x-15)(x-11)]^2 + [(x-1)(x-4)(x-7)(x-3)(x-8)(x-15)(x-11)]^2} \\
 q_3(t+1) &= k_3''(x) = \frac{[(x-0)(x-2)(x-1)(x-7)(x-8)(x-12)(x-15)(x-11)]^2}{[(x-0)(x-2)(x-1)(x-7)(x-8)(x-12)(x-15)(x-11)]^2 + [(x-4)(x-3)(x-10)(x-9)]^2} \\
 Y &= k_4''(x) = \frac{[(x-0)(x-1)(x-4)(x-7)(x-3)(x-8)(x-9)(x-12)(x-15)(x-11)]^2}{[(x-0)(x-1)(x-4)(x-7)(x-3)(x-8)(x-9)(x-12)(x-15)(x-11)]^2 + [(x-2)(x-10)]^2}
 \end{aligned}
 \tag{7}$$

## 5 Conclusions

In this paper, we show how is possible by extension theory [1] to solve a very important problem in realization of Boolean function by one step nonlinear neurons.

**Acknowledgments.** This paper is sponsored by National Natural Science Foundation Project (61503085) and Science and Technology Planning Project of Guangdong Province (2016A040404015).

## References

1. Cai, W., Yang, C.Y., Lin, W.: Extension Engineering Methods. Science Press, Beijing (2003)
2. Yang, C.Y., Cai, W.: Extenics: Theory, Method and Application. Science Press & The Educational Publisher, Beijing (2013)
3. Cai, W.: Extension theory and its application. Chin. Sci. Bull. **38**(8), 1538–1548 (1999)
4. Li, W.H., Yang, C.Y.: Extension information-knowledge-strategy system for semantic interoperability. J. Comput. **3**(8), 32–39 (2008)
5. Resconi, G., Xu, X., Xu, G.: Introduction to Morphogenetic Computing. SCI, vol. 703. Springer, Cham (2017). <https://doi.org/10.1007/978-3-319-57615-2>