Received: 27 April 2017 | Revised: 14 June 2017 | Accepted: 14 June 2017

DOI: 10.1002/cnm.2920

RESEARCH ARTICLE - FUNDAMENTAL

WILEY

A new type of neurons for machine learning

Fenglei Fan | Wenxiang Cong | Ge Wang

Biomedical Imaging Center, BME/CBIS, Rensselaer Polytechnic Institute, Troy, New York, USA

Correspondence
Ge Wang, Biomedical Imaging Center, BME/CBIS, Rensselaer Polytechnic Institute, Troy, New York, USA.
Email: wangg6@rpi.edu

Funding information
Clark & Crossan Endowment Fund at Rensselaer Polytechnic Institute

Abstract

In machine learning, an artificial neural network is the mainstream approach. Such a network consists of many neurons. These neurons are of the same type characterized by the 2 features: (1) an inner product of an input vector and a matching weighting vector of trainable parameters and (2) a nonlinear excitation function. Here, we investigate the possibility of replacing the inner product with a quadratic function of the input vector, thereby upgrading the first-order neuron to the second-order neuron, empowering individual neurons and facilitating the optimization of neural networks. Also, numerical examples are provided to illustrate the feasibility and merits of the second-order neurons. Finally, further topics are discussed.

KEYWORDS

artificial neural network, convolutional neural network, machine learning, second-order neuron

1 | INTRODUCTION

In the field of machine learning, artificial neural networks especially deep neural networks (CNNs) have recently achieved remarkable successes in various applications such as classification, unsupervised learning, prediction, image processing, and analysis.[1-3] Excited by the tremendous potential of machine learning, major efforts are being made to improve machine learning methods.[4] An important aspect of the methodological research is how to optimize the topology of a neural network. For example, generative adversarial networks were proposed to use some deep results of game theory. In particular, the Wasserstein distance was introduced for effective, efficient, and stable performance of the generative adversarial network.[5,6]

To our best knowledge, all artificial neural networks/CNNs are currently constructed with neurons of the same type characterized by the 2 features: (1) an inner product of an input vector and a matching weighting vector of trainable parameters and (2) a nonlinear excitation function.[7-9] Although these neurons can be interconnected to approximate any practical function, the topology of the network is not unique. On another side of this nonuniqueness, *our hypothesis is that the type of neurons suitable for general machine learning is not unique either, which is a new dimension of machine learning research*.

The above hypothesis is well motivated. It is commonly known that the current structure of artificial neurons was inspired by the inner working of biological neurons. Biologically speaking, a core principle is that diversity brings synergy and prosperity at all levels. In biology, multiple types of cells are available to make various organisms.[10] Indeed, many genomic components contribute complementary strengths and compensate for others′ weaknesses.[11] Most of cell types are highly adapted to desirable forms and functions. Synthetic biological research may enhance the biological diversity further. *This observation makes us wonder if we could have multiple types of neurons for machine learning either in general or for specific tasks*.

As the first step along this direction, here we investigate the possibility of replacing the inner product with a quadratic function of the input vector, thereby upgrading the first-order neuron to the second-order neuron, empowering individual neurons and facilitating the optimization of neural networks.

The model of the current single neurons, which are also referred to as perceptrons, has been applied to solve linearly separable problems. For linearly inseparable tasks, multilayers of neurons are needed to perform multiscale nonlinear analysis. In other words, existing neurons can only perform linear classification individually, and the linearly limited cellular function can be only enhanced via cellular interconnection into an artificial organism. *Our curiosity is to produce such an artificial organism with neurons capable of performing quadratic classification*.

In the next section, we describe the second-order neurons. Given n inputs, the new type of neurons could have $3n$ parameters to approximate a general quadratic function or have $n(n+1)/2$ degrees of freedom to admit any quadratic function. Furthermore, we formulate how to train the second-order neuron. In the third section, we present numerical simulation results for fuzzy logic operations. In the last section, we discuss relevant issues and conclude the paper.

2 | SECOND-ORDER MODEL AND OPTIMIZATION

The current structure of an artificial neuron is shown in Figure 1, where we define $w_0 := b$ and $x_0 = 1$. The linear function of the input vector produces the output[8]

$$f(x) = \sum_{i=0}^{n} w_i x_i, \tag{1}$$

and then $f(x)$ will be nonlinearly processed, such as by a sigmoid function. Clearly, the single neuron can separate 2 sets of inputs that are linearly separable. In contrast, for linearly inseparable groups of inputs, the single neuron is subject to classification errors. For example, a single neuron is incapable of simulating the function of the XOR gate.

We introduce a new type of neurons in Figure 2 where the input vector is turned into 2 inner products and 1 norm term for summation before feeding to the nonlinear excitation function. Again, for compact notation, we define that $w_{0r} := b_1$, $w_{0g} := b_2$, and $x_0 = 1$. Then, the output function is expressed as follows:

$$f(x) = \left(\sum_{i=0}^{n} w_{ir} x_i\right)\left(\sum_{i=0}^{n} w_{ig} x_i\right) + \sum_{i=1}^{n} w_{ib} {x_i}^2 + c. \tag{2}$$

In this embodiment, the threshold is chosen by a sigmoid function. Equation 2 is quadric and has the current neuron type as a special case. Because of its added nonlinearity, our proposed neuron is intrinsically superior to the current neuron in terms of representation power such as for classification.

The training algorithm can be formulated for the proposed second-order neuron as follows, assuming the sigmoid function[8]

$$\sigma(x) = \frac{1}{1 + \exp(-\beta x)}, \tag{3}$$

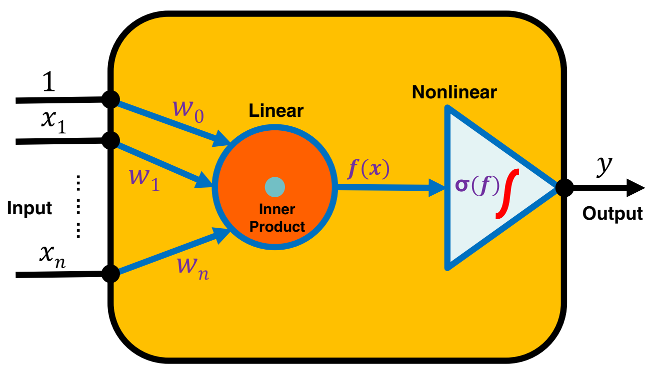

FIGURE 1 Current structure of an artificial neuron consisting of a linear inner product and a nonlinear excitation function

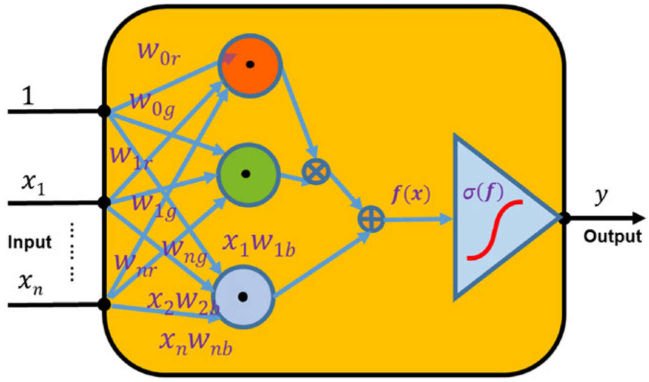

FIGURE 2 Exemplary structure of our proposed artificial neuron consisting of a quadratic form and a nonlinear excitation function

where $\beta = 1$ in this pilot study. Let us denote a training dataset of m samples by $X^k = \left(x_1^k, \ x_2^k, \ \dots, \ x_n^k\right)$, along with the ideal output data y^k, $k = 1, \ 2, \ \dots, \ m$. Then, the output of the second-order neuron can be modeled as

$$\begin{aligned} &h\left(X^k, w_r, w_g, \mathrm{w}_b, b_1, b_2\right) = \sigma(f(x)) \\ &= \sigma\left(\left(\sum_{i=1}^{n} w_{ir} x_i^k + b_1\right)\left(\sum_{i=1}^{n} w_{ig} x_i^k + b_2\right) + \sum_{i=1}^{n} w_{ib}\left(x_i^k\right)^2 + c\right). \end{aligned} \tag{4}$$

Let us define the error function as

$$E\left(\overrightarrow{w_r}, \overrightarrow{w_g}, \overrightarrow{w_b}, b_1, b_2\right) = \frac{1}{2}\sum_{k=1}^{m}\left(h\left(X^k, \overrightarrow{w_r}, \overrightarrow{w_g}, \overrightarrow{w_b}, b_1, b_2\right) - y^k\right)^2. \tag{5}$$

The error function depends on the following structural parameters: b_1, b_2, c, $\overrightarrow{w_r} = (w_{1r}, \ w_{2r}, \ \dots, \ w_{nr})$, $\overrightarrow{w_g} = \left(w_{1g}, \ w_{2g}, \ \dots, \ w_{ng}\right)$, and $\overrightarrow{w_b} = (w_{1b}, \ w_{2b}, \ \dots, \ w_{nb})$.

The goal of machine learning is to find optimal parameters to minimize the objective function.[9] The optimal parameters can be found using the gradient descent method with an appropriate initial guess. Thus, we can iteratively update $\overrightarrow{w_r}$, $\overrightarrow{w_g}$, $\overrightarrow{w_b}$, b_1, b_2, and c in the form of $\alpha = \alpha - \eta \cdot \frac{\partial E}{\partial \alpha}$, where α denotes a generic variable of the objective function and the step size η is typically set between 0 and 1. The gradient of the object function for any sample can be computed as follows:

$$\frac{\partial E}{\partial w_{ir}} = \left(h\left(\overrightarrow{x_i}\right) - y_i\right)\frac{\partial \sigma}{\partial x} x_i\left(\sum_{i=1}^{n} w_{ig} x_i + b_2\right)$$

$$\frac{\partial E}{\partial w_{ig}} = \left(h\left(\overrightarrow{x_i}\right) - y_i\right)\frac{\partial \sigma}{\partial x} x_i\left(\sum_{i=1}^{n} w_{ir} x_i + b_1\right)$$

$$\frac{\partial E}{\partial w_{ib}} = 2\left(h\left(\overrightarrow{x_i}\right) - y_i\right)\frac{\partial \sigma}{\partial x} w_{ib} x_i$$

$$\frac{\partial E}{\partial b_1} = \left(h\left(\overrightarrow{x_i}\right) - y_i\right)\frac{\partial \sigma}{\partial x}\left(\sum_{i=1}^{n} w_{ig} x_i + b_2\right)$$

$$\frac{\partial E}{\partial b_2} = \left(h\left(\overrightarrow{x_i}\right) - y_i\right)\frac{\partial \sigma}{\partial x}\left(\sum_{i=1}^{n} w_{ir} x_i + b_1\right)$$

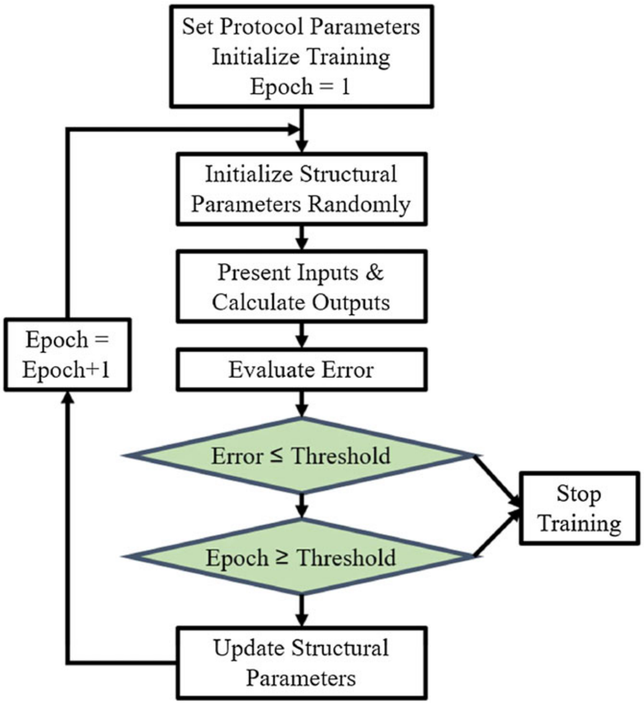

FIGURE 3 Flowchart for training the second-order neuron in terms of its structural parameters (weights and offsets). Note that we plan to improve this iterative process for higher efficiency

$$\frac{\partial E}{\partial c} = \left(h\left(\overrightarrow{x_i}\right) - y_i\right)\frac{\partial \sigma}{\partial x}.$$

A general optimization flowchart is in Figure 3.

3 | NUMERICAL RESULTS

In our study, the proposed second-order neuron of 2 input variables was individually applied in fuzzy logic experiments. Extending classic Boolean logic, fuzzy logic was extensively studied and applied over the past decades, which manipulates vague or imprecise logic statements. Compared with Boolean logic, the value of a fuzzy logic variable is on the interval [0, 1], since a practical judgement may not be purely black (false or 0) or white (true or 1). The fuzziness of a logic variable can be easily reflected in a continuous variable by its closeness to either 0 or 1.

For visualization, the color map "cool" in MATLAB was used to represent the functional value at every point. We used "o" for 0 and "+" for 1. With the proposed second-order neuron, the training process kept refining a quadratic contour to separate labeled points for the highest classification accuracy. By the nature of the second-order neuron, the contour can be 2 lines or curves including parabolic and elliptical boundaries

3.1 | XOR gate

First, the training dataset was simply the XOR logic table, which is not separable by a single neuron of the conventional type. To implement the XOR-gate operation with our proposed second-order neuron, the initial parameters could be randomly selected in the framework of evolutionary computation. For example, an initial seed was set to $w_r = [-0.4, -0.4]$, $w_g = [0.2, 1]$, $w_b = [0, 0]$, $b_1 = -0.9095$, $b_2 = -0.6426$, $c = 0$. The trained logic map is shown in Figure 4, yielding a perfect result.

Then, we generated an XOR-like pattern. The initial parameters were set to $w_r = [0.07994, -0.2119]$, $w_g = [0.06049, -0.144]$, $w_b = [0,0]$, $b_1 = -0.9095$, $b_2 = -0.6426$, $c = 0$. As shown in Figure 5, the deformed XOR pattern was perfectly segmented by our proposed second-order neuron after training.

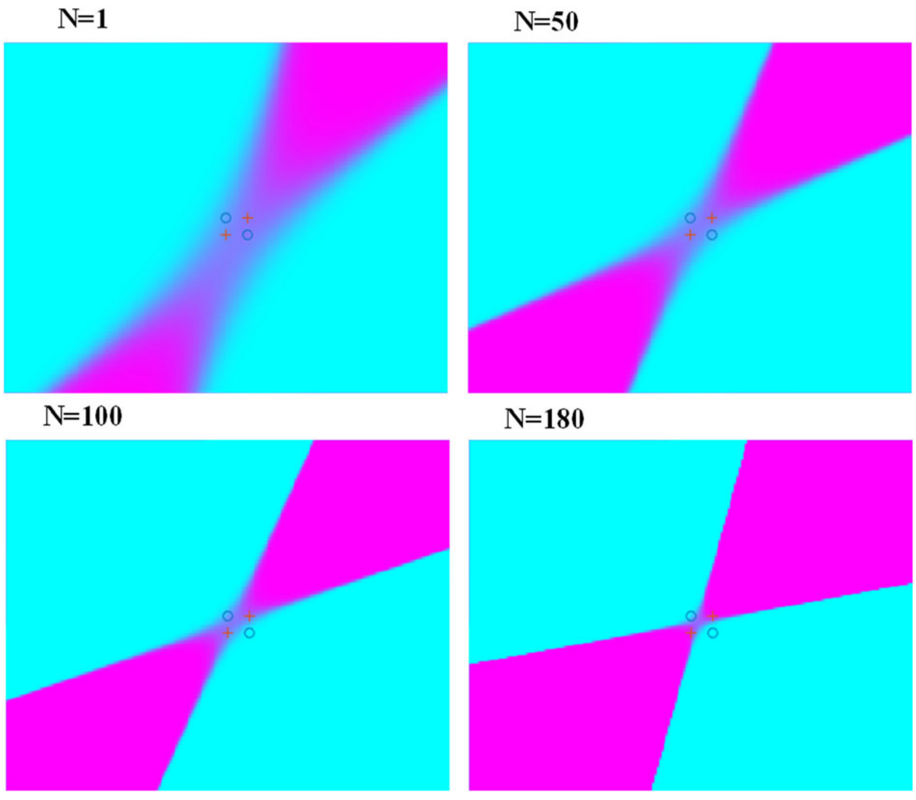

FIGURE 4 XOR logic implemented by the proposed second-order neuron after 180 iterations. After the training, the outputs at [0,0], [0,1], [1,0], and [1,1] are 0.4509, 0.5595, 0.5346, and 0.3111, meaning 0, 1, 1, and 0 respectively relative to the threshold 0.5

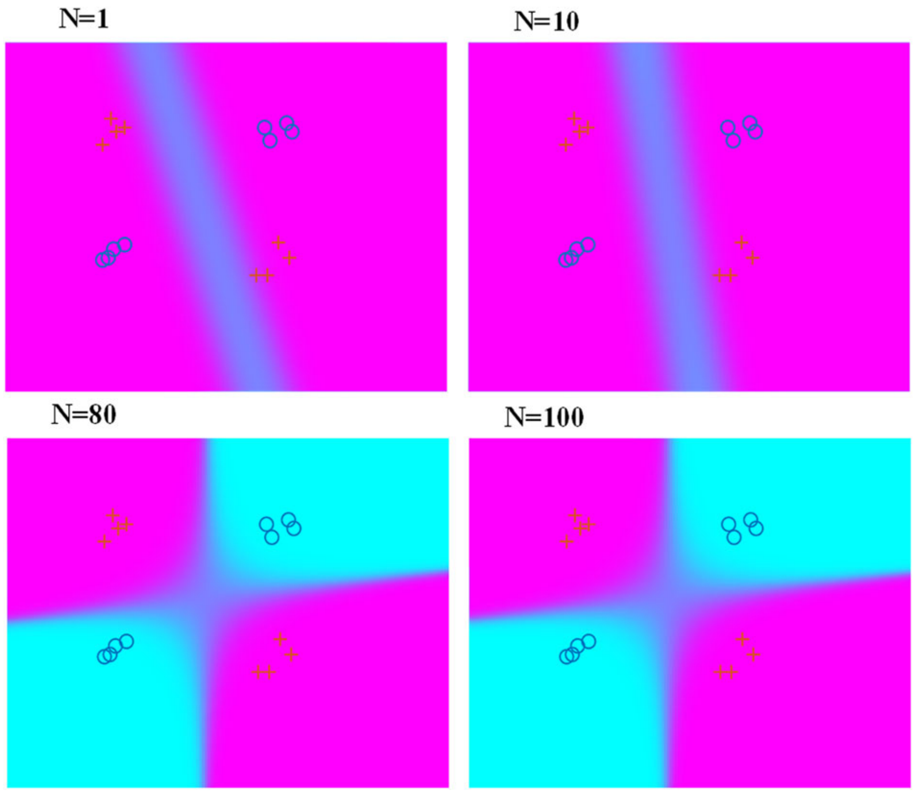

FIGURE 5 XOR-like function computed by the proposed second-order neuron after 100 iterations

3.2 | NAND and NOR gates

While the XOR gate is exemplary, it is not a universal logic gate. To further show the power of the proposed second-order neuron, additional datasets were generated that respectively demand fuzzy NAND and NOR operations for correct classification. Then, training steps similar to those in Section 3.1 were repeated to simulate NAND and NOR operations respectively. In this test, the initial parameters were set to $w_r = [0.4, -0.1]$, $w_g = [0.3, 0.1]$, $w_b = [0, 0]$, $b_1 = 0$, $b_2 = 0$, $c = 1.3$ and $w_r = [-1,1]$, $w_g = [1,-2]$, $w_b = [0,0]$, $b_1 = -0.5$, $b_2 = 1$, $c = 0$, for NAND and NOR tasks, respectively. The results are shown in Figures 6 and 7.

3.3 | Concentric rings

Yet another type of natural patterns linearly inseparable is concentric rings. As an example, we generated 2 concentric rings, which were respectively assigned to 2 classes. With the initial parameters $w_r = [0.04,0.01]$, $w_g = [0.03,-0.01]$,

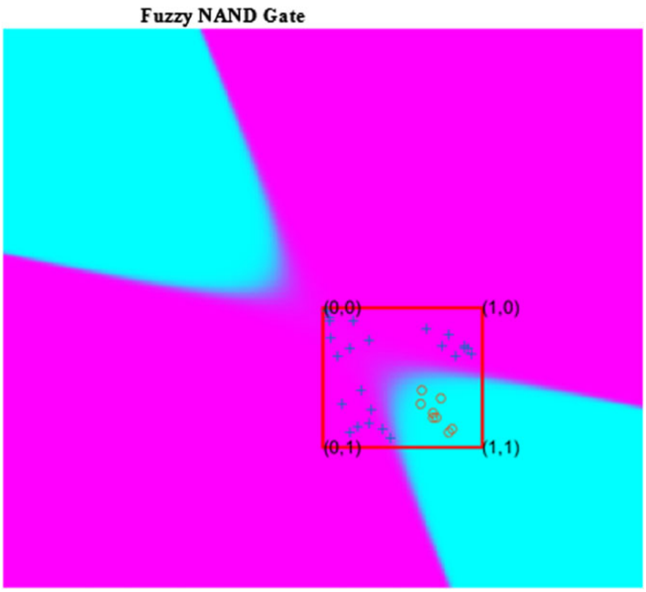

FIGURE 6 Fuzzy NAND function implemented by the second-order neuron after 300 iterations

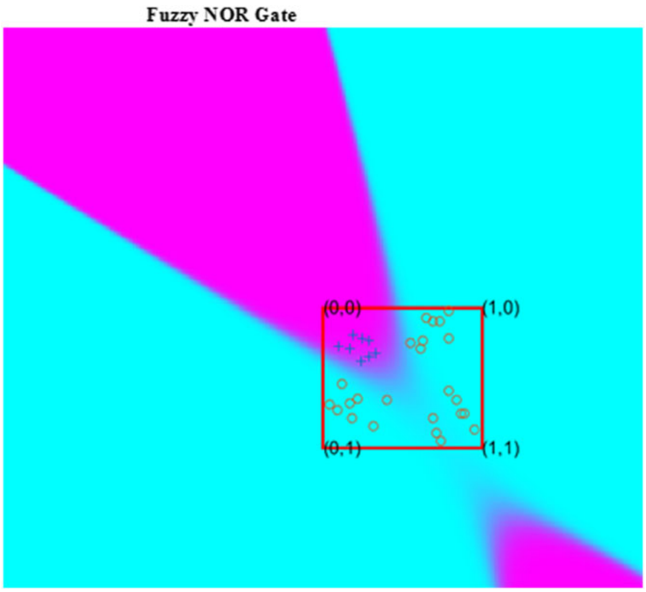

FIGURE 7 Fuzzy NOR function of the second-order neuron after 100 iterations

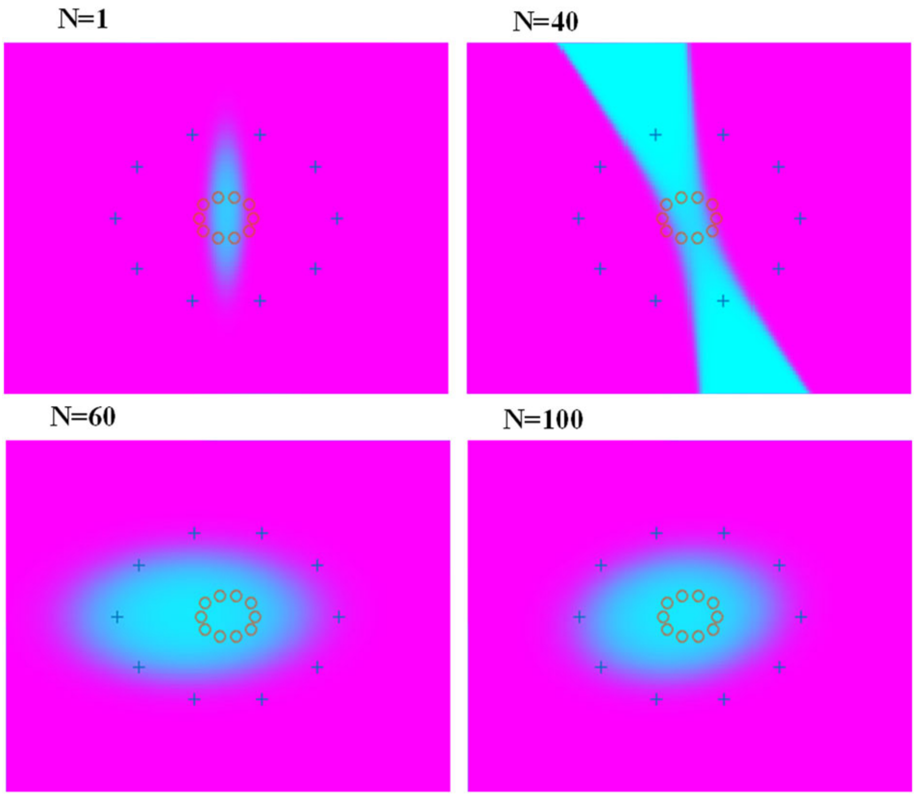

FIGURE 8 Perfect classification of concentric rings with the proposed second-order neuron

$w_b = [0,0.4]$, $b_1 = 0.1$, and $b_2 = 0.2$, $c = 1.3$, our second-order neuron was trained, producing an ideal outcome as shown in Figure 8, which completely separates the inner ring from the outer ring.

3.4 | Biomedical engineering application

In Leite et al,[12] it was shown that fuzzy logic, combined with expert knowledge, allowed to extract warning features of the physiological signals from an ICU patient. Vital signs, including mean blood pressure (MBP) and partial oxygen saturation (POS_2), were used as inputs to the fuzzy system.

To show the feasibility and utility of our proposed second-order neuron in this type of applications, we only selected MBP and POS_2 as inputs to a single second-order neuron. Then, we used the definitions of the relevant fuzzy functions as described in Leite et al.[12] As a result, the fuzzy judgement for MBP is that Low MBP, when the value of MBP <80, Normal MBP when 80 to 130, and High MBP when >130. Also, the fuzzy judgement for POS_2 is that Low POS_2 when the

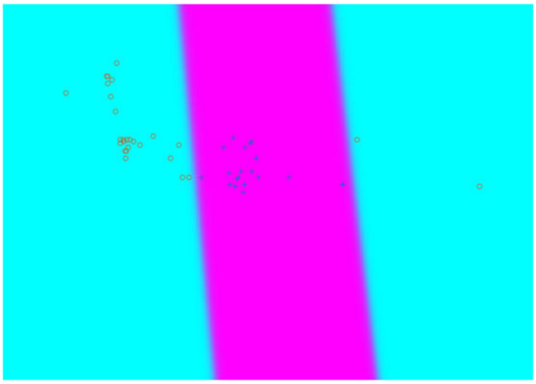

FIGURE 9 Perfect detection of critical signs by a single second-order neuron from mean blood pressure and partial oxygen saturation data

value <94 and Normal POS_2 when 94 to 100. To combine the fuzzy judgements, 2 rules for our single second-order neuron are as follows:

Rule 1. If both MBP and POS_2 are normal, then the monitoring signal is Normal;
Rule 2. If there is either high/low MBP or low POS_2, then the monitoring signal is Abnormal.

We download the data recorded from [09:00:00 22/09/2017] to [09:45:00 22/09/2017] of Record a40487n in the MIMIC II Waveform Database.[13] We used 0 for "Normal" and 1 for "Abnormal." The inputs were normalized before training. With the random initial parameters $w_r = [1,-1]$, $w_g = [-0.8,1]$, $w_b = [0.1,0.2]$, $b_1 = -1$, and $b_2 = -0.8$,

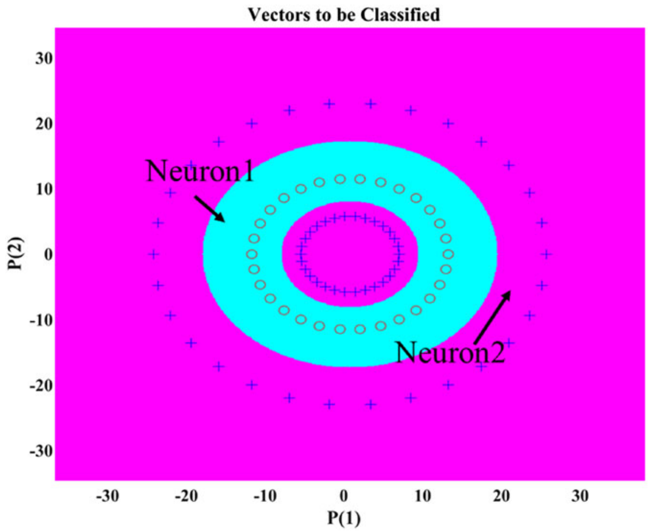

FIGURE 10 Separation of a donut-shaped region from its background with a network of 2-2-1 second-order neurons

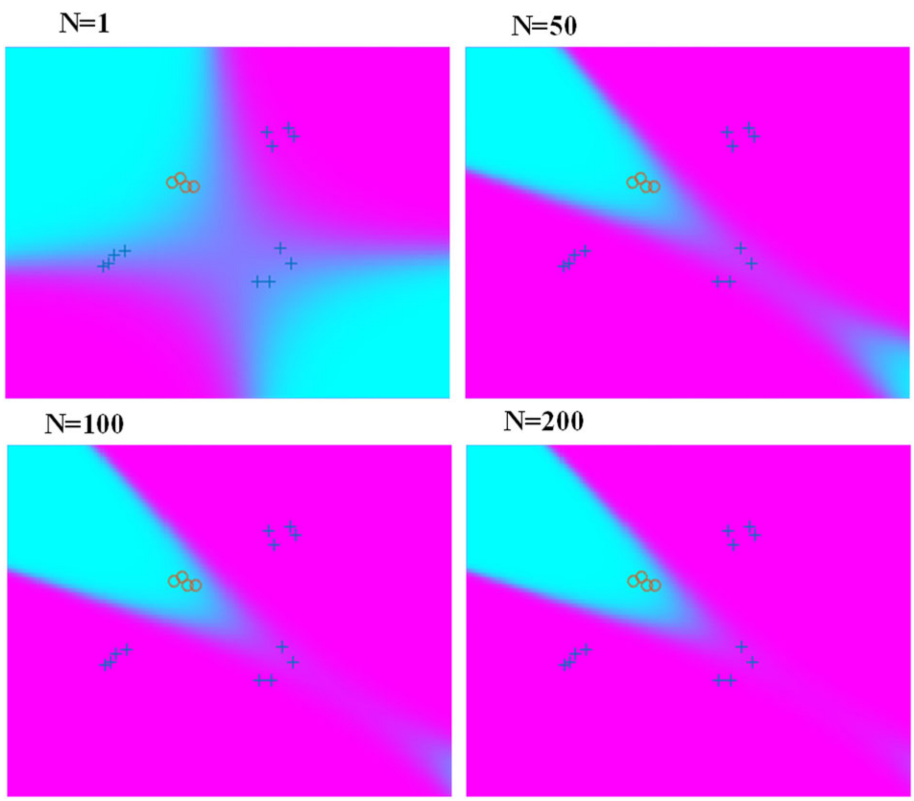

FIGURE 11 OR-like pattern classified with the general second-order neuron

$c = 1.3$, our second-order neuron was successfully trained to produce a perfect prediction as shown in Figure 9. Note that this example only serves to give pilot data, and an optimized neural network for this purpose may need more than 2 inputs and more than a single neuron.

4 | DISCUSSIONS AND CONCLUSION

As demonstrated in this paper, the proposed second-order neuron works well in solving basic fuzzy logic problems and qualified as a different building block for machine learning. The sufficiency of the proposed neuron type can be trivially argued. Clearly, the traditional neuron type is a special case of the proposed second-order neuron. Since the traditional neuron is sufficient for a general machine learning task, the proposed neuron type should be sufficient as well. Furthermore, the superiority of the proposed neuron type can be argued in analogy to the functional approximation. The traditional neuron linearly synthesizes inputs into a single number, which is the first-order Taylor approximation. What we have proposed is the second-order Taylor approximation. This flexibility should simplify the overall architecture of a neural network in certain types of applications including but not limited to fuzzy logic tasks.

From the perspective of digital/fuzzy logic, the proposed second-order neurons can directly implement all typical logic gates with or without fuzziness, especially the so-called universal gates such as NAND and NOR. This comment is a hint that the second-order neurons may be appropriate modules for fuzzy logic processing or soft computing when logic values are not binary, which means the correct logic judgements cannot be directly made with the common neuron that is the first-order but can be efficiently made with the proposed second-order neuron.

Since deep learning performs the best in many applications, it is interesting to see how second neurons perform in an architecture. As an example, we designed a 2-2-1 network to classify 3 regions defined by concentric rings whose radii are 5, 10, and 20, respectively. The donut-shaped zone delimited by the rings of radii 5 and 20 represents one category, while the 2 separate inner and outer spaces are for the other category. For neurons in the hidden layer, the connection parameters are $w_r = [0,0]$, $w_g = [0\ 0]$, $w_b = [1\ 1]$, $b_1 = 0$, $b_2 = 0$, $c = -49$ and $w_r = [0,0]$, $w_g = [0\ 0]$, $w_b = [1\ 1]$, $b_1 = 0$, $b_2 = 0$, $c = -175$. For neuron in the output layer, the connection parameters are $w_r = [-2,0]$, $w_g = [0\ 0]$, $w_b = [1\ 1]$, $b_1 = 1$, $b_2 = 0$, $c = 0.7$. With only three second-order neurons, the nonlinear separation task is accomplished in a straightforward fashion, as shown in Figure 10.

The limited performance of classic one-hidden-layer neural networks on the approximation of nonlinear functions was noticed decades ago. Recurrent neural networks consisting of first-order neurons were shown to work for nonlinear function approximations such as XOR. While an RNN can implement temporal XOR, a single second-neuron can directly implement spatial XOR,[14] and RNNs with second-order neurons should be much more powerful than that with first-order neurons.

While the above-proposed second-order neuron has important new representation capabilities, the general second-order neuron should use more parameters:

$$f(\vec{x}) = \sum_{i,j=1,i\geq j}^{n} a_{ij}x_i x_j + \sum_{k=1}^{n} b_k x_k + c.$$

From a training set $\{\vec{x}_p\}$ and $\{y_p\}$, the parameters $\{a_{ij}\}$, $\{b_k\}$, and c can be updated using the gradient descent method in the evolutionary computing framework, where the gradient can be computed as follows:

$$\frac{\partial E}{\partial a_{ij}} = \left(h(\vec{x_p}) - y_p\right)\frac{\partial \sigma}{\partial x} x_i x_j,$$

$$\frac{\partial E}{\partial b_k} = \left(h(\vec{x_p}) - y_p\right)\frac{\partial \sigma}{\partial x} x_k,$$

$$\frac{\partial E}{\partial c} = \left(h(\vec{x_p}) - y_p\right)\frac{\partial \sigma}{\partial x}.$$

This general second-order neuron can be similarly trained. For example, this general neuron was adapted for an OR-like operation with the initial parameters $a_{11} = a_{22} = a_{21} = 0.1$, $b_1 = b_2 = 1$, $c = 0.1$, yielding the result in Figure 11. Despite the fact that the number of parameters is now in the order of n^2, this should not present a major computational challenge when these second-order neurons are used in CNNs.

It is underlined that with new types of neurons, we have broadened the space for the network optimization to adjust not only neural interconnections but also neural inner makings. The second-order neuron is only an example. Other types of neurons can be imagined as well. Preferred neuron structures may depend on specific applications. To have a greater impact of the second-order neurons, we would need to connect them into multilayers and compare the power and efficiency of competing networks made with first- and second-order neurons, respectively. To train a network consisting of second-order neurons, we will formulate a generalized back propagation algorithm and optimize its performance. Importantly, we need to find specific niche applications of second-order neurons. A complete theory remains missing for the neural network consisting of first-order neurons and needless to say about theoretical results on the network with second-order neurons. However, this challenge is also our opportunities to perform further research on solution optimality, robustness, etc.

In conclusion, a new type of artificial neurons, which we call the second-order neurons, has been proposed for an enhanced expressing capability relative to what the current neuron has. Our pilot results show some encouraging results, but much efforts are needed to demonstrate a real-world application and establish a rigorous theory.

ACKNOWLEDGEMENT

This work is partially supported by the Clark & Crossan Endowment Fund at Rensselaer Polytechnic Institute, Troy, New York, USA.

ORCID

Fenglei Fan http://orcid.org/0000-0003-3691-5141
Ge Wang http://orcid.org/0000-0002-2656-7705

REFERENCES

1. Pereira S, Pinto A, Alves V, Silva CA. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Trans Med Imaging*. 2016;35:1240-1251.
2. Shin HC, Roth HR, Gao M, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans Med Imaging*. 2016;35(5):1285-1298.
3. Anthimopoulos M, Christodoulidis S, Ebner L, Christe A, Mougiakakou S. Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE Trans Med Imaging*. 2016;35(5):1207-1216.
4. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. 1943. *Bull Math Biol*. 1990;52(1-2):99-115; discussion 73-97.
5. Su Z, Zeng W, Wang Y, Lu ZL, Gu X. Shape classification using Wasserstein distance for brain morphometry analysis. *Inf Process Med Imaging*. 2015;24:411-423.
6. Arjovsky M, Chintala S, Bottou L. Wasserstein GAN, arXiv: 1701.07875, 2017.
7. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw*. Jan, 2015;61:85-117.
8. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. Cambridge: MIT Press; 2016.
9. Shen D, Wu G, Zhang D, Suzuki K, Wang F, Yan P. Machine learning in medical imaging. *Comput Med Imaging Graph*. Apr, 2015;41:1-2.
10. Medler DA, Dawson MR. Training redundant artificial neural networks: imposing biology on technology. *Psychol Res*. 1994;57(1):54-62.
11. Cartwright HM. Artificial neural networks in biology and chemistry: the evolution of a new analytical tool. *Methods Mol Biol*. 2008;458:1-13.
12. Leite CR, Sizilio GR, Neto AD, Valentim RA, Guerreiro AM. A fuzzy model for processing and monitoring vital signs in ICU patients. *Biomed Eng Online*. 2011;10:68.
13. Goldberger AL, Amaral LAN, Glass L, et al. Physiobank, physiotoolkit, and physionet. *Circulation*. 2000;101:215-220.
14. Lipton ZC, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning, arXiv preprint arXiv:1506.00019, 2015.

How to cite this article: Fan F, Cong W, Wang G. A new type of neurons for machine learning. *Int J Numer Meth Biomed Engng*. 2018;34:e2920. https://doi.org/10.1002/cnm.2920