This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2814065, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Predicting House Price with a Memristor-based Artificial Neural Network

J. J. Wang[1], S. G. Hu[1], X. T. Zhan[1], Q. Luo[1]; Q. Yu[1], Z. Liu[2], T. P. Chen[3], Y. Yin[4], Sumio Hosaka[4] and Y. Liu[1]

[1]State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu 610054, P. R. China
[2]School of Materials and Energy, Guangdong University of Technology, Guangzhou 510006, China
[3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798
[4]GraduateSchool of Engineering, Gunma University, 1-5-1Tenjin, Kiryu, Gunma 376-8515, Japan

Corresponding author: Y. Liu (e-mail: yliu1975@uestc.edu.cn).

This work is supported by the NSFC under project No.61774028 and 61771097, the Fundamental Research Funds for the Central Universities under project No. ZYGX2016Z007, and the Opening Project of Science and Technology on Reliability Physics and Application Technology of Electronic Component Laboratory under project No. ZHD201602.

ABSTRACT Synaptic memristor has attracted much attention for its potential applications in artificial neural networks (ANNs). However useful applications in real life with such memristor-based networks have seldom been reported. In this work, an artificial neural network based on memristors is designed to learn a multi-variable regression model with back-propagation algorithm. A weight unit circuit based on memristor, which can be programed as an excitatory synapse or inhibitory synapse, is introduced. The weight of the electronic synapse is determined by the conductance of the memristor, and the current of the synapse follows the charge-dependent relationship. The ANN has the ability to learn from labeled samples and make predictions after online training. As an example, the ANN was used to learn a regression model of the house prices of several Boston towns in the US, and the predicted results are found to be close to the target data.

INDEX TERMS house price predicting; neural network; memristor; memristive synapse

I. INTRODUCTION

The idea of building an artificial brain has existed for a long time. Artificial neural network (ANN) is a possible method to realize artificial intelligence. Until now, researchers have made many amazing achievements in the applications like pattern recognition [1-3], face detection [4,5], learning cat concept from cat videos in the internet [6], classifying [7], and playing Go game [8]. However, most of these works were implemented with CPUs and GPUs. Both have separated memories and processors and consume a large amount of energy [8]. Another way to realize artificial intelligence is to design a customized integrated circuit (IC), which consumes much less power and works in a high speed [7, 9]. However, the customized IC has difficulties in online training, and usually, the neural network in the customized IC [10] has to be trained with the help of von-Neumann computers. In 1983, Michalski proposed a machine that can learn from labeled samples [11]. In 1986, Rumelhart introduced a back-propagation (BP) algorithm to train ANN online automatically [12,13]. In 1971, Chua predicted the fourth basic circuit element, namely, the memristor [14], which was later demonstrated in laboratory by Williams *et al*. in 2008[15]. Subsequently, some studies showed that memristors could be used as electronic synapses in ANN [16-18]. For example, ANN consisting of neurons and memristor-based synapses was used to mimic the associate function of human brain [16,19].

In this work, a multi-layer feed-forward neural network has been designed using memristors as electronic synapses to realize automatic online training. Memristor weights of the ANN can be adjusted by the BP algorithm to build up a regression model spontaneously, which is different from the classification model realized with memristor-based network in [1]. The trained ANN has been used to predict the house price of several Boston towns in the US, and the predicted result is close to the target data.

II. SIMULATION AND METHODS

A. The architecture of ANN

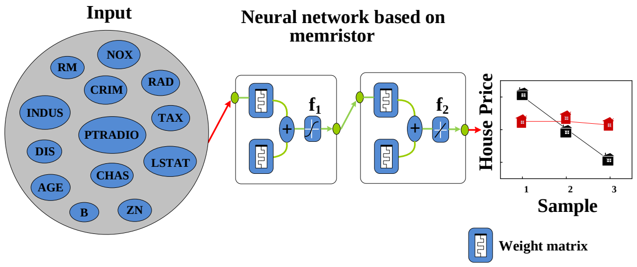

FIGURE 1. **Schematic illustration of the house price predicting system. Thirteen parameters are considered to predict the house price, and the abbreviations of these parameters in the "Input" module are explained in Supplementary Table 1.**

The house price predicting system is illustrated in Figure 1. The ANN used in the house price predicting system is constructed with memristor synapses. The relevant labeled information (i.e. a labeled data includes thirteen parameters and its corresponding target house price) is used to train the ANN online. The ANN learns to predict the house price and then does the prediction with the unlabeled information (i.e., an unlabeled data includes only thirteen parameters). Abbreviations for the relevant parameters in the input module are explained in Supplementary Table 1. f_1 and f_2 are transfer functions. The ANN in this work has a 2-layer feed-forward network, as shown in Figure 1. It consists of an input module, a hidden layer and an output layer as shown in Figure 2(a). Vector $\boldsymbol{X}$ is the input while $\hat{y}$ is the output. As shown in Figures 2(b) and 2(c), there are four weight matrices, $\boldsymbol{V}$, $\boldsymbol{W}$, $\boldsymbol{b_1}$, and $\boldsymbol{b_2}$ for the three layers as given below:

$$\boldsymbol{V} = (v_{ij})_{m\times n} \tag{1}$$

$$\boldsymbol{W} = (w_{ij})_{1\times m} \tag{2}$$

$$\boldsymbol{b_1} = (\gamma_1, \gamma_2, \cdots \gamma_j, \cdots, \gamma_n) \tag{3}$$

$$\boldsymbol{b_2} = (\theta) \tag{4}$$

As shown in Figure 2(b), V_{bpi} is the back propagation voltage applied to the ith line in $\boldsymbol{V}$, while V_{xj} is the input voltage applied to the jth column in $\boldsymbol{V}$. As shown in Figure 2(c), V_{bp} is the back propagation voltage, and V_{hi} is the voltage transferred from the hidden layer applied to the *i*th column in $\boldsymbol{W}$. V_{fix} is a constant voltage applied to $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$. The circuit of the weight unit with one memristor is illustrated in Figure 2(d). All of the circuits shown in Figure 2 are designed and simulated with Cadence and MATLAB, respectively.

As shown in Figure 3, the weight matrix element v_{ij} (i = 1,2,…,m; *and* j = 1,2,3,…,n) is connected to the input vector $x_i(i = 1,2, \ldots n)$ in $\boldsymbol{X}$. The *i*th Neuron of the hidden layer ($N_i^{(Hidden)}$) collects the currents i_{ij} (j = 1,2,3,…,n) from the weight units v_{ij} (j = 1,2,3,…,n), and the currents are integrated in the membrane potential capacitor $C_{MEMi}^{(Hidden)}$ (i = 1,2,…n) connected to the transfer function module f_1. The output h of the neuron is determined by f_1 according to Equations (5) and (6).

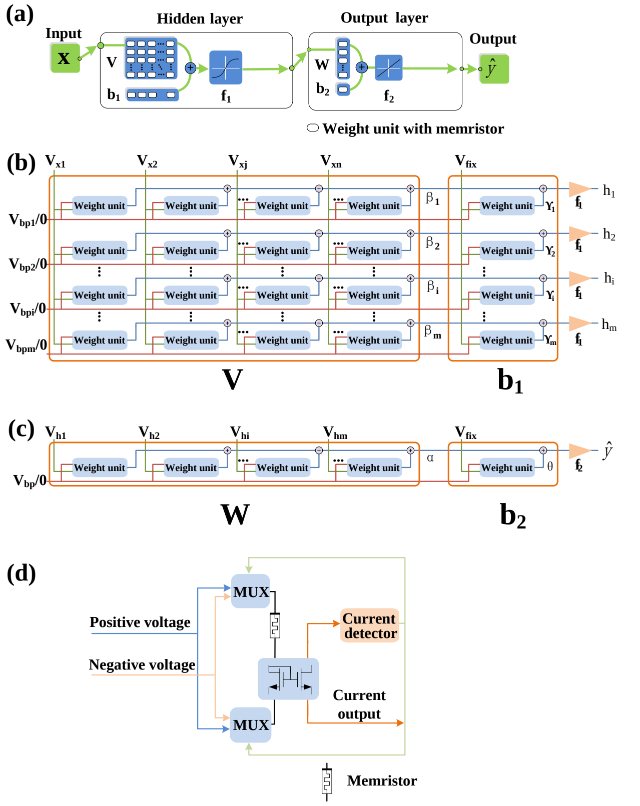

FIGURE 2. **Schematic of the two layer forward neural network (a), weight matrices in hidden layer (b) and output layer (c), and circuit schematic of the weight unit with a memristor (d).**

$$\boldsymbol{h} = f_1(\boldsymbol{V}\cdot\boldsymbol{X} + \boldsymbol{b_1}) \tag{5}$$

where $f_1\ (x) = \frac{1}{1+e^{-x}}$.

$$h_i = f_1(\beta_i + \gamma_i) \tag{6}$$

$\beta_i = \sum_{j=1}^{n}(v_{ij}\cdot x_j)$is the sum of the synaptic currents flowing through all membranes of $N_i^{(Hidden)}$; and γ_i is the possible leak current of the *i*th membrane of $N_i^{(Hidden)}$.

There is only one neuron $N^{(output)}$ in the output layer which collects currents from the weight units w_{1j} (j = 1, 2, … , n), as shown in Figure 4. The linear-function module $f_2(x)$ converts voltage of the membrane potential capacitor $C_{MEM}^{(Output)}$ to the final output $\hat{y}$ according to Eqs. (7) and (8).

$$\hat{y} = f_2(\boldsymbol{W}\cdot\boldsymbol{h} + \boldsymbol{b_2}) \tag{7}$$

where $f_2(x) = cx$ and c is a constant. $\hat{y}$ can also be rewritten as

$$\hat{y} = f_2(\alpha + \theta) \tag{8}$$

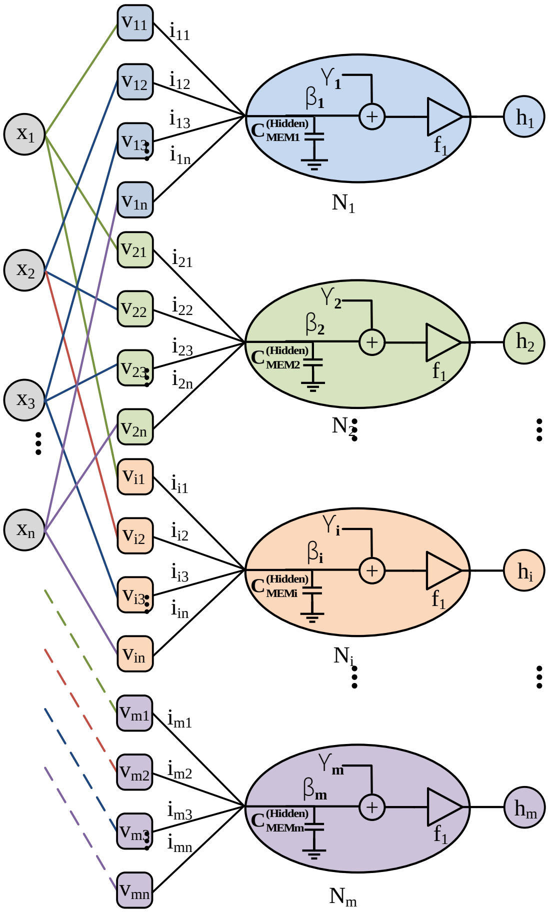

FIGURE 3. Schematic illustration of the network connection in the hidden layer.

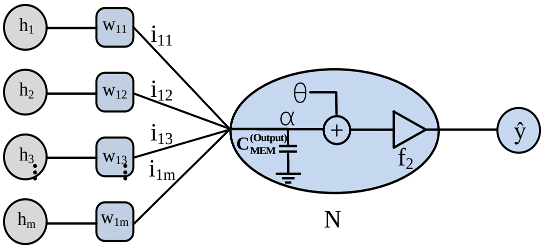

FIGURE 4. Schematic illustration of the network connection in the output layer.

where $\alpha = \sum_{j=1}^{\mathrm{m}} (w_{1j} \cdot h_{\mathrm{j}})$ is the sum of synaptic currents which flow through the membranes of $N^{(Output)}$; and θ is the possible leak current of the membrane of $N^{(Output)}$.

B. Weight unit design based on memristor

The memristor was predicted by Chua according to the symmetry of circuit theory [14]. In this work, we adopt the memristor model reported in [18]. Supplementary Notes 1 and Supplementary Figure 1 present the details of electrical characteristics of the memristor model used in this work. Voltage pulses above v_{th} can also be used to adjust the conductance of memristor as well as the weight. In this work, v_{th} is set to 0.5 V.

Weight matrix contains weight units and biasing units as shown in Figure 2(b) and Figure 2(c), respectively. The biasing unit is similar to the weight unit except that its magnitude is fixed at 0.5 V. As shown in Figure 2(d), the memristor-based weight unit consists of two switches, a memristor, a micro current detector and a multi-output current mirror. The memristance (i.e., the resistance of the memristor) can be adjusted to an appropriate magnitude by the voltage across the memristor. The multiplication can be realized by obtaining the current flowing through the memristor which is driven by the voltage pulse. In order to avoid convergence of the ANN to a local minimum point, the memristances of the memristors are set to random values in a uniform distribution during system initialization. When the current sink is connected to the output, a positive weight is obtained; when the current source is connected to the output, a negative weight is obtained.

In the training mode, the back propagation pulse amplitude is fixed at 0.5 V. The switches at the anode and cathode of one memristor can change the polarity of the voltage pulse. When the weight is positive, positive pulses can increase the conductance of the memristor as well as the weight, while negative pulses can reduce the conductance and also the weight. The detailed circuit design for the whole ANN is discussed in Supplementary Note 2 and Supplementary Figure 2.

C. Training and predicting method

In the training mode, the ANN is able to adjust the weight online automatically based on the BP algorithm. The square of the error between the target result y^k and the predicted output $\hat{y}^k$ ($\hat{y}^k$ represents the predicted result for the k^{th} input sample data), $\mathrm{E_k}$, represents the precision of prediction. The steepest descent is adopted to adjust the weight matrixes. E_k, Δv_{ij}, Δw_{1j}, $\Delta \gamma_j$ and Δθ are expressed by Equation series (9).

$$\begin{aligned} E_k &= \frac{1}{2}(y_k - \hat{y}_k)^2 \\ \Delta v_{ij} &= -\eta_1 \frac{\partial \mathrm{E_k}}{\partial v_{ij}} \\ \Delta \gamma_j &= -\eta_1 \frac{\partial E_k}{\partial \gamma_j} \\ \Delta w_{ij} &= -\eta_2 \frac{\partial E_k}{\partial w_{1j}} \\ \Delta \theta &= -\eta_2 \frac{\partial E_k}{\partial \theta} \end{aligned} \tag{9}$$

The partial derivatives to E_k are used to calculate the change of weight (i.e., Δv_{ij}, Δw_{1j}, $\Delta \gamma_j$ and $\Delta\theta$). The derivative of f_1 is expressed by Equation (10), and the derivative of f_2 is equal to the constant c. According to the chain derivative method, the partial derivative can be calculated according to Equation (11).

$$\frac{d}{dx} f_1(x) = f_1(x) \cdot \big(1 - f_1(x)\big) \tag{10}$$

$$\frac{\partial E_k}{\partial w_{1j}} = \frac{\partial E_k}{\partial \hat{y}^k} \cdot \frac{\partial \hat{y}^k}{\partial \alpha} \cdot \frac{\partial \alpha}{\partial w_{1j}} \tag{11}$$

where $\frac{\partial \alpha}{\partial w_{1j}} = h_j$.

By defining $g = \frac{\partial E_k}{\partial \hat{y}^k} \cdot \frac{\partial \hat{y}^k}{\partial \alpha} = c \cdot (y^k - \hat{y}^k)$, the following expression can be obtained:

$$\frac{\partial \mathrm{E_k}}{\partial v_{ij}} = \frac{\partial E_k}{\partial h_i} \cdot \frac{\partial h_i}{\partial v_{ij}} = \left(\frac{\partial E_k}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \alpha} \cdot \frac{\partial \alpha}{\partial h_i}\right) \cdot \left(\frac{\partial h_i}{\partial \beta_i} \cdot \frac{\partial \beta_i}{\partial v_{ij}}\right) = (g \cdot w_{1i}) \cdot h_i(1 - h_i) \cdot x_{\mathrm{j}} \tag{12}$$

By setting $e_h = \frac{\partial \mathrm{E_k}}{\partial \gamma_i} = \frac{\partial E_k}{\partial h_i} \cdot \frac{\partial h_i}{\partial \gamma_i} = (g_i \cdot w_{1i}) \cdot h_i(1 - h_i)$, Equation series (13) can be obtained:

$$\begin{aligned} \Delta \mathrm{v_{ij}} &= -\eta_1 e_h \cdot x_j \\ \Delta \mathrm{\gamma_i} &= -\eta_1 e_h \\ \Delta w_{ij} &= -\eta_2 g \cdot \mathrm{h_j} \\ \Delta \mathrm{\theta_j} &= -\frac{\eta \partial E_k}{\partial \theta_j} = -\eta_2 g \end{aligned} \tag{13}$$

In the training mode, the initial output $\hat{y}^k$ is calculated by the ANN, and the teacher's signals are applied to the output layer. With the BP algorithm, the weight of the output layer is updated, and the weight of the hidden layer is updated according to Equation series (13).

In the circuit design, the difference between the output $\hat{y}^k$ and the teacher's signal y^k can be represented by the BP signal $(\hat{y}^k - y^k)$. Then the BP signal modulates the delay time.

By applying the teacher's signal, the feed-forward signal h_j of the hidden layer is generated again. The implementation of the BP algorithm is based on the linear increase segment of the STDP rule (Supplementary Figure 3(f)) [18,19]. The voltage of the pre-synapse is modulated by the amplitude, while the voltage of the post-synapse with amplitude V_{th} is modulated by the delay time. When the delay time is zero, the voltages on both the anode and cathode are the same such that the voltage across the memristor is zero. If the delay time is longer than 1 ms, it is considered as positive; otherwise it is negative. When the delay time is positive, the voltage during the delay time at the transient edge is beyond the threshold voltage, and the memristance changes following Equation (14). The change of memristance is proportional to both the delay time and amplitude of the post-synapse voltage. Neglecting the high order series of Equation (15), ΔM is proportional to the product of delay time t_d and $V_{pre-syn}$.

$$\Delta M_{ij}(t_d, V_{pre-syn}) = \int_0^{t_d} f(V_{pre-syn} - V_{th}) dt = \int_0^{t_d} I_0 \left(e^{\frac{V_{pre-syn} + V_{th}}{V_0}} - e^{\frac{v_{th}}{V_0}}\right) dt \tag{14}$$

where $t_d = A_m \cdot (\hat{y}^k - y^k)$ and $V_{pre-syn} = h_j \cdot V_{th}$.

Then

$$\Delta M_{ij}(t_d, V_{pre-syn}) = \int_0^{t_d} I_0 \, e^{\frac{v_{th}}{V_0}} \left(e^{\frac{V_{pre-syn}}{V_0}} - 1\right) dt$$

where $V_{pre-syn} \ll V_0$

$$\approx \int_0^{t_d} I_0 e^{\frac{v_{th}}{V_0}} \left(1 + \frac{V_{pre-syn}}{V_0} - 1\right) dt$$

$$= I_0 e^{\frac{v_{th}}{V_0}} \cdot \frac{V_{pre-syn}}{V_0} \cdot t_d$$

$$= \frac{I_0}{V_0} e^{\frac{v_{th}}{V_0}} \cdot (h_j \cdot V_{th}) \cdot A_m(\hat{y}_j^k - y_j^k)$$

$$= A_p \cdot h_j \cdot (\hat{y}^k - y^k) \tag{15}$$

where $A_p = \frac{I_0}{V_0} V_{th} \cdot A_m \cdot e^{\frac{v_{th}}{V_0}}$

The input layer generates the amplitude-modulated signal, while the hidden layer generates the waveform which is modulated by the delay time $\mathrm{t_d}$ based on Equation series (13). t_d can be generated by an analog multiplier which produces the output $t_{ij} = A_m \cdot (g \cdot w_{1i}) \cdot h_i(1 - h_i)$. In this case, the change of M_{ij} follows

$$\Delta M_{ij}(t_d, V_{pre-syn}) = A_p \cdot g \cdot w_{1i} \cdot h_i(1 - h_i) \tag{16}$$

In the predicting mode, the input module converts the input data vector ($\boldsymbol{x^k} = (x_1^k, x_2^k, \dots, x_j^k \dots, x_n^k), k \in (1,2,3, \dots N)$ to individual voltage pulses whose levels are proportional to the input data x_j. The voltage pulses are applied to the anodes of the first weight matrix unit (the $\mathrm{V_{pb}}$ is set to 0 V), and the voltage pulses are multiplied with the corresponding weights to output current signals. The currents are collected by $C_{MEMi}^{(Hidden)}(i = 1,2, \dots n)$ of the neuron in the hidden layer. In the output layer, $C_{MEM}^{(Output)}$is used in a way similar to that of $C_{MEMi}^{(Hidden)}$ to collect the currents, and the output is finally obtained by the conversion from the current to voltage signal by the linear function module f_2.

III. RESULTS

The house price can be predicted by inputting relevant parameters, such as crime rate, property-tax rate, pupil-teacher ratio, etc. As an example, the data in suburbs of Boston obtained from [20] is used. The relevant data is normalized to its maximum value of each parameter. The data is divided into two sets: one is for training and the other is for prediction. Before training, the weights are randomly set in the uniform distribution, and the error histogram is showed in Figure 5(a). As observed in the figure, the errors are large and randomly distributed.

Then, the ANN is switched to the training mode, and the relevant parameters are input to the ANN; at the same time, the target house price is fed to the output layer as the supervision signal. The weights are adjusted according to the BP algorithm to make the outputs close to the target prices during the training process.

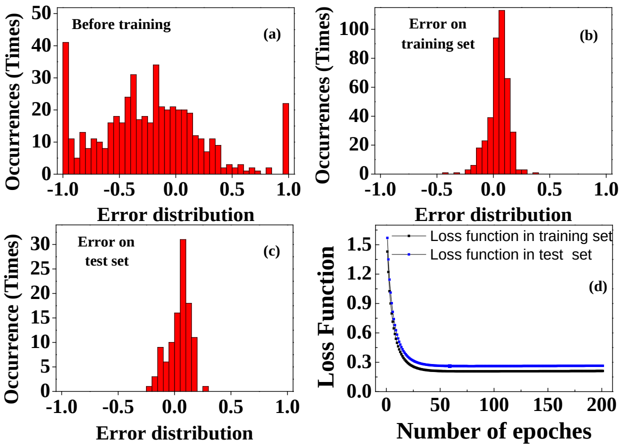

FIGURE 5. **Occurrence rate of errors before training (a), after training with the parameters from the training set (b) and test set (c), and loss function (MSE) vs epoch(d).**

The mean square error (MSE) loss function ($Loss\ function = \sum_{k=1}^{\mathrm{N}}(y_k - \hat{y}_k)^2$) is used to estimate the performance of the network. The loss function decreases gradually with the increase of training epoch, and reaches a saturation value of ~0.24 after ~50 epochs, indicating successfully learning of house price prediction, as shown in figure 5(d). The error histogram is roughly concentrated to zero after 200 training epochs, as shown in Figure 5(b). Finally, the parameters of the test set, which are not used to train the model, are fed to the trained ANN, and the error histogram is showed in Figure 5(c). The errors between the predicted prices and the target prices are small and distribute in a very narrow region (about [-0.25, 0.25]), which again proves that the trained ANN is able to precisely predict the house prices.

Figure 6(a) shows the target house prices, the prices predicted with the parameters from both the training and test set after training. Figure 6(b) shows the relation between the target prices and the predicted prices. As illustrated in the figure, the departures from the linear regression line $y = x$ (dash line in Figure 6(b)) is small, indicating that the prediction is reliable.

IV. CONCULSIONS

In this paper, a 2-layer feed-forward neural network is designed using memristors as synapses. The weights of synapses can be adjusted online by the pulse voltage with BP algorithm. The ANN has the ability to learn to predict the house price under training mode, and then can successfully predict the house price in the predicting mode. The ANN was trained with the house price samples of several Boston towns in the US to make predictions, and the predicted results are found to be close to the target data.

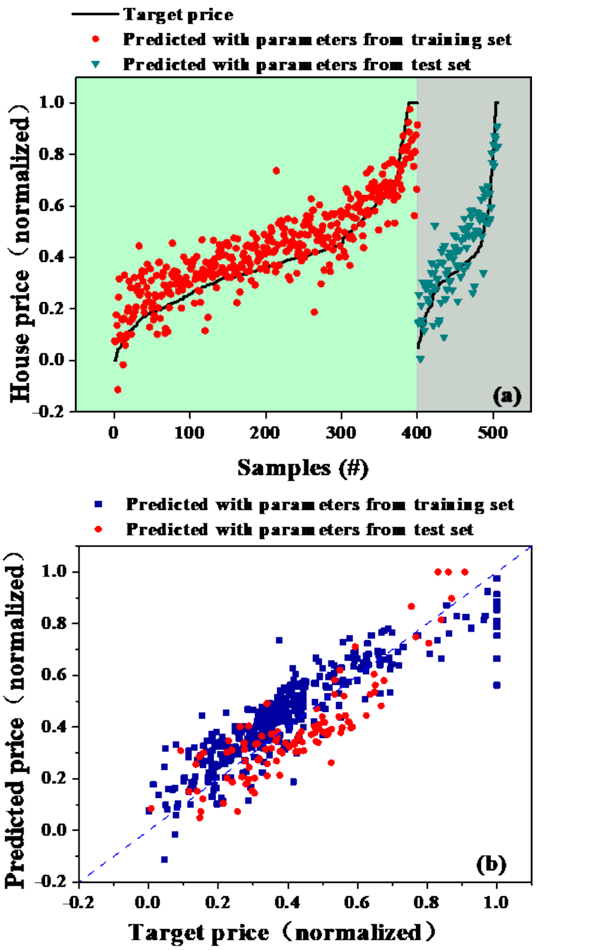

FIGURE 6. **(a) Predicted house price with parameters from the training set and test set, respectively; (b) comparison between the predicted price and the target price. The values are normalized to the highest house price. The target prices are obtained from [20].**

ACKNOWLEDGMENT

This work is supported by the NSFC under project No.61774028 and 61771097, Fundamental Research Funds for the Central Universities under project No. ZYGX2016Z007, and the Opening Project of Science and Technology on Reliability Physics and Application Technology of Electronic Component Laboratory under project No. ZHD201602.

REFERENCES

[1] M. Chu, B. Kim, S. Park, et al. Neuromorphic hardware system for visual pattern recognition with memristor array and CMOS neuron. IEEE Transactions on Industrial Electronics, 2015,62:2410–2419
[2] S. Park, M. Chu, J. Kim, et al. Electronic system with memristive synapses for pattern recognition. Scientific Reports, 2015,5:1–9
[3] X. Wu, V. Saxena, K. Zhu. Homogeneous Spiking Neuromorphic System for Real-World Pattern Recognition. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2015,5:254–266
[4] K. Curran, X. Li, N. McCaughley. Neural network face detection. The Imaging Science Journal, 2013,53:105–115

[5] S. S. Farfade, M. J. Saberian, L. J. Li. Multi-view face detection using deep convolutional neural networks. Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015:643–650

[6] Q. V. Le. Building high-level features using large scale unsupervised learning. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013:8595–8598

[7] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 2014,345:668–673

[8] D. Silver, A. Huang, C. J. Maddison, et al. Mastering the game of Go with deep neural network sand tree search. Nature, 2016,529:484–489

[9] M. N. Bojnordi, E. Ipek. Memristive Boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning. 2016 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2016:1–13

[10] P. Merolla, J. Arthur, F. Akopyan, et al. A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. 2011 IEEE Custom Integrated Circuits Conference (CICC), 2011:1–4

[11] J. G. Carbonell, R. S. Michalski, T. M. Mitchell. Machine learning: A historical and methodological analysis. AI Magazine, 1983,4:69

[12] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning internal representations by error propagation. DTIC Document

[13] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning representations by back-propagating errors. Cognitive Modeling, 1988,5:1

[14] L. Chua. Memristor-the missing circuit element. IEEE Transactions on Circuit Theory, 1971,18:507–519

[15] D. B. Strukov, G. S. Snider, D. R. Stewart, et al. The missing memristor found. Nature, 2008,453:80–83

[16] S.G. Hu, Y. Liu, Z. Liu, et al. Associative memory realized by a reconfigurable memristive Hopfield neural network. Nature Communications, 2015, 6:7522

[17] S. H. Jo, T. Chang, I. Ebong, et al. Nanoscale memristor device as synapse in neuromorphic systems. Nano Letters, 2010,10:1297–1301

[18] T. Serrano-Gotarredona, T. Prodromakis, B. Linares-Barranco. A proposal for hybrid memristor-CMOS spiking neuromorphic learning systems. IEEE Circuits and Systems Magazine, 2013,13:74–88

[19] S. Song, K. D. Miller, L. F. Abbott. Competitive Hebbian learning through spike-timing dependent synaptic plasticity. Nature Neuroscience, 2000,3:919–926

[20] Murphy, P. M., & Aha, D. W. (1992). UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets/Housing

J.J. Wang received the B.S. degree in microelectronics from the University of Electronic Science and Technology of China, Chengdu, China, where he is currently pursuing the Ph.D. degree. His current research interests include thin-film transistor, nonvolatile memory devices, and their applications in artificial intelligence.

S. G. Hu received the Ph.D. degree in microelectronics from the University of Electronic Science and Technology of China, Chengdu. His current research interests include neuromorphic devices, chips and systems. He has been an Associate Professor with the University of Electronic Science and Technology of China, Chengdu, China, since 2016.

X. T. Zhan received the B.S. degree in microelectronics from the University of Electronic Science and Technology of China, Chengdu, China, where he is currently pursuing the M.S. degree. His current research interests include thin-film transistor, nonvolatile memory devices, and their applications in artificial intelligence.

Q. Luo received the Ph.D. degree in microelectronics and solid-state electronics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2007. He is currently an Associate Professor with the School of Microelectronics and Solid-State Electronics, UESTC. His current research interests include the fabrication of GaN-based HEMT and Si-based MOSFET devices.

Q. Yu received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2010. He is currently a Professor and the Vice Dean with the School of Microelectronics and Solid-State electronics, UESTC.

Zhen Liu was born in Changsha, China, in 1983. He received the Ph.D. degree in 2011 from the School of Electrical and Electronic Engineering, NTU. He is currently an Associate Professor with the School of Materials and Energy, Guangdong University of Technology. His research interestsinclude the electrical and optoelectronic properties of metal-dielectric nanocomposites and their device applications.

T. P. Chen received the Ph.D. degree from The University of Hong Kong, Hong Kong, in 1994. He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

Y. Yin received the Ph.D. degree in Microelectronics and Solid-state Electronics from Shanghai Jiao Tong University, China. His current research interests include Micro/nano-electronic devices for future's IoT & AI and renewable energy and Nano-fabrication. He has been an Assistant Professor with the Gunma University, Japan, since 2008.

Sumio Hosaka was born in 1948. He received the Ph.D. degree in engineering from Waseda University, Tokyo, Japan, in 1983. He joined the Hitachi Central Research Laboratory (HCRL) Company Ltd., Kokubunji, Japan, in 1971, where he worked on semiconductor lithography using electron, ion, and light beams, nanometrology using electron beam, and scanning probe microscopy (SPM) in HCRL and terabyte-huge storage using SPM in the Hitachi Advanced Research Laboratory (HARL). After joining the Hitachi Kenki Finetech Company, Tsuchiura, Japan, in 1999, where he worked on the development of in-line atomic force microscope products, he then joined the Department of Electronic Engineering, Gunma University, Kiryu, Japan, in 2000. Since then, he has been a Professor and is currently working on nanodevices, nanofabrication, and nanometrology with the Department of Nano-Material Systems, Graduate School of Engineering.

Y. Liu received the B.Sc. degree in microelectronics from Jilin University, China, in 1998, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2005. In 2006, he was awarded the prestigious Singapore Millennium Foundation Fellowship. In 2008, he joined the School of Microelectronics, University of Electronic Science and Technology, China, as a full professor. He is the author or coauthor of over 130 peer-reviewed journal papers and more than 100 conference papers. His current research includes memristor neural network system, and neuromorphic ICs.