SCIENTIFIC REPORTS
nature research

OPEN

Binarized Neural Network with Silicon Nanosheet Synaptic Transistors for Supervised Pattern Classification

Received: 22 February 2019
Accepted: 29 July 2019
Published online: 12 August 2019

Sungho Kim[1], Bongsik Choi[2], Jinsu Yoon[2], Yongwoo Lee[2], Hee-Dong Kim[1], Min-Ho Kang[3] & Sung-Jin Choi[2]

In the biological neural network, the learning process is achieved through massively parallel synaptic connections between neurons that can be adjusted in an analog manner. Recent developments in emerging synaptic devices and their networks can emulate the functionality of a biological neural network, which will be the fundamental building block for a neuromorphic computing architecture. However, on-chip implementation of a large-scale artificial neural network is still very challenging due to unreliable analog weight modulation in current synaptic device technology. Here, we demonstrate a binarized neural network (BNN) based on a gate-all-around silicon nanosheet synaptic transistor, where reliable digital-type weight modulation can contribute to improve the sustainability of the entire network. BNN is applied to three proof-of-concept examples: (1) handwritten digit classification (MNIST dataset), (2) face image classification (Yale dataset), and (3) experimental 3 × 3 binary pattern classifications using an integrated synaptic transistor network (total 9 × 9 × 2 = 162 cells) through a supervised online training procedure. The results consolidate the feasibility of binarized neural networks and pave the way toward building a reliable and large-scale artificial neural network by using more advanced conventional digital device technologies.

Although relatively little is known about the principle of information processing in the brain, it is certain that the information flows from neuron to neuron through synapses which have adjustable connection strengths (*i.e.*, synaptic weights). The learning process in the brain is consequently the reconfiguration of the synaptic weights in the neural network, where the weights are updated in an analog manner. Based on this fact, several learning rules regulating the evolution of the synaptic weights have been proposed (such as spike-timing-dependent plasticity[1]), and recently, intensive efforts have been made to implement an electronic synaptic device that can emulate the functionality of synapses. The final goal of this research, which has been named neuromorphic engineering, is the realization of innovative computing architecture (neuromorphic system) based on an artificial neural network to overcome the energy inefficiency of conventional von Neumann architecture, by mimicking both the functional and structural characteristics of the biological systems[2,3].

To date, the most promising candidates for a synaptic device are two-terminal resistive switching devices, *i.e.*, memristors[4]. With memristors, analog conductance states can be modulated by using only a minuscule amount of energy consumption and can be maintained over the long term, which indicates the promising feasibility of emulating biological synapses[5–9]. Furthermore, by applying such memristors, primitive levels of artificial neural networks (*i.e.*, synaptic device arrays) have been demonstrated experimentally for the application of pattern classification[8], analog-to-digital conversion[10], principal component analysis[11], sparse coding calculations[12], reservoir computing[13], *K*-means data clustering[14], and differential equation solver[15]. However, the on-chip implementation of neuromorphic systems with emerging synaptic devices is still very challenging due to the instability of analog weight modulation in a synaptic device, which has been identified in recent simulation studies[16,17]: although the neuromorphic systems are capable of tolerating the device-to-device variation or noise to a certain degree[18–20],

[1]Department of Electrical Engineering, Sejong University, Seoul, 05006, Korea. [2]School of Electrical Engineering, Kookmin University, Seoul, 02707, Korea. [3]Department of Nano-process, National Nanofab Center (NNFC), Daejeon, 34141, Korea. Correspondence and requests for materials should be addressed to S.-J.C. (email: sjchoiee@kookmin.ac.kr)

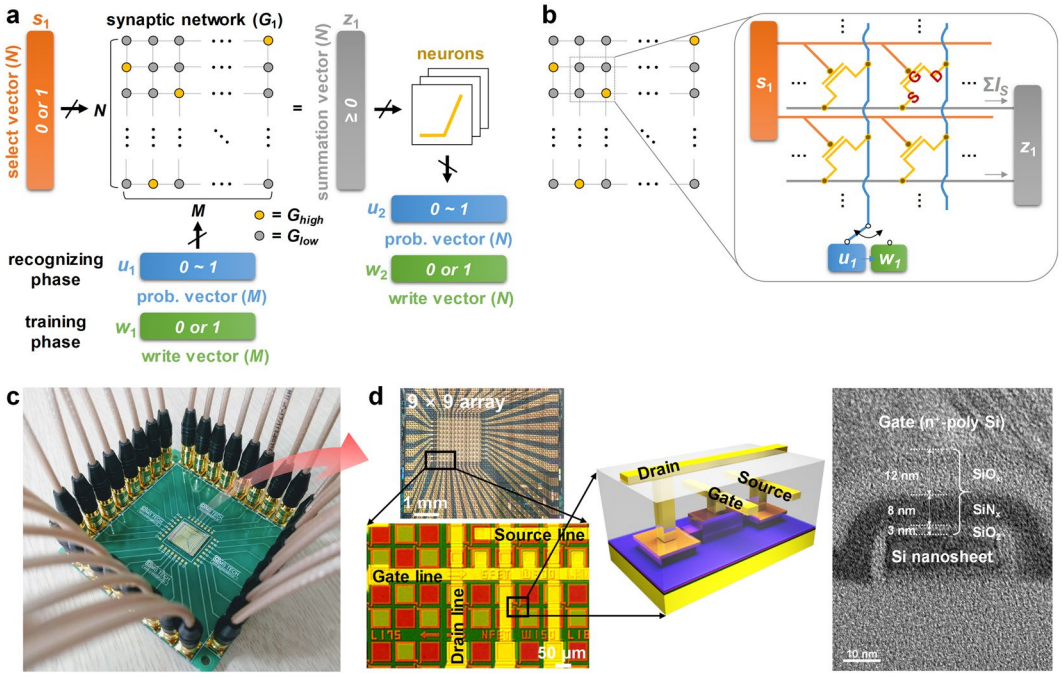

Figure 1. (**a**) The architecture of the binary neural network with M inputs and N outputs. The input pattern information corresponds to the $u_1(i)$ and $w_1(i)$, the $s_1(i)$ enables supervised training by selecting a specific row, and the $z_1(i)$ is the output of the network. (**b**) The schematic of synaptic transistor array, where $s_1(i)$ involves V_G, and either $u_1(i)$ or $w_1(i)$ involves V_D. Integrated I_S in a row direction corresponds to $z_1(i)$. (**c**) The photo of the test board with an integrated synaptic transistor array. (**d**) The optical and transmission electron microscope images of the synaptic transistor. The SiN charge trap layer embedded in the gate dielectric enables the digital-type channel conductance switching with high reliability.

intrinsic nonlinearity and uncontrollability of analog conductance switching behavior critically degrades the performance of the system[16,17,20,21]. Unfortunately, this issue is common to almost all memristors and could not be solved by further optimizing the fabrication process or materials because the physical mechanism of the analog conductance modulation is typically an atomic-level random process based on electro/thermodynamics[22–24]. Although several methods for precise adjustment of the analog weight have been proposed[25–27], these methods require a specially designed pulse waveform and impractical complex peripheral circuitry. In addition, recent memristors exhibit improved reliability[28–30], but the fabrication process of the device is complex or the materials used are incompatible with conventional silicon processes, is a critical obstacle to the design of peripheral circuits.

Alternatively, the sustainability and reliability of digitally switching devices have been guaranteed over the past 20 years[31]. For example, in the case of the present NAND flash technology, stable multiple memory states with 3-dimensional stackability have already been applied to a product. Particularly, the density of the NAND flash already exceeds 2×10^9 bits/mm^2 [32], close to the density of synapses in the human frontal cortex (1.1×10^9 synapses/mm^3)[33]. Therefore, if the well-qualified conventional digital devices can contribute to a synaptic device, the goal of achieving on-chip implementation of a neuromorphic system can be realized sooner. Here, we demonstrate a binarized neural network (BNN) where the synaptic device is a more advanced digital-type switchable device, that is, a gate-all-around (GAA) silicon nanosheet transistor. A developed training/recognition algorithm of BNN enables the task of pattern classification with a supervised online training scheme. In this study, BNN is applied to three proof-of-concept examples: (1) handwritten digit classification (MNIST dataset[34]) verified by the simulation, (2) face image classification (Yale dataset[35]) verified by the simulation, and (3) 3×3 binary pattern classifications by using an integrated two 9×9 synaptic transistor arrays. The simulation and experimental results consolidate the feasibility of BNN and pave the way toward building a reliable, large-scale, and practical neuromorphic system from advanced conventional digital device technologies.

Results and Discussion

Figure 1a depicts the architecture of BNN[36] with M inputs and N outputs. Synaptic weights in the network $G_1(i, j)$ are given within one binary value: $G_1(i, j)$ 1{G_{high} or G_{low}}; G_{high} and G_{low} represent the high- and low-conductance states of the synaptic device, respectively (subscripted numbers indicate the order of each network when multiple networks are involved). The input pattern information is delivered into the network by two types of vectors: $u_1(i)$ and $w_1(i)$ denote the probability- and write-vector, respectively. When an input pattern needs to be distinguished from previously trained patterns (*i.e.*, recognizing phase), $u_1(i)$ is applied to the network. $u_1(i)$ corresponds directly to each pixel of information of the input pattern such as the intensity, which is rescaled to $0 \leq u_1(i) \leq 1$. When an input pattern needs to be trained by updating the synaptic weight (*i.e.*, training phase), $w_1(i)$ instead of $u_1(i)$ is applied to the network, where $w_1(i)$ w{0 or 1} is stochastically determined by learning probability

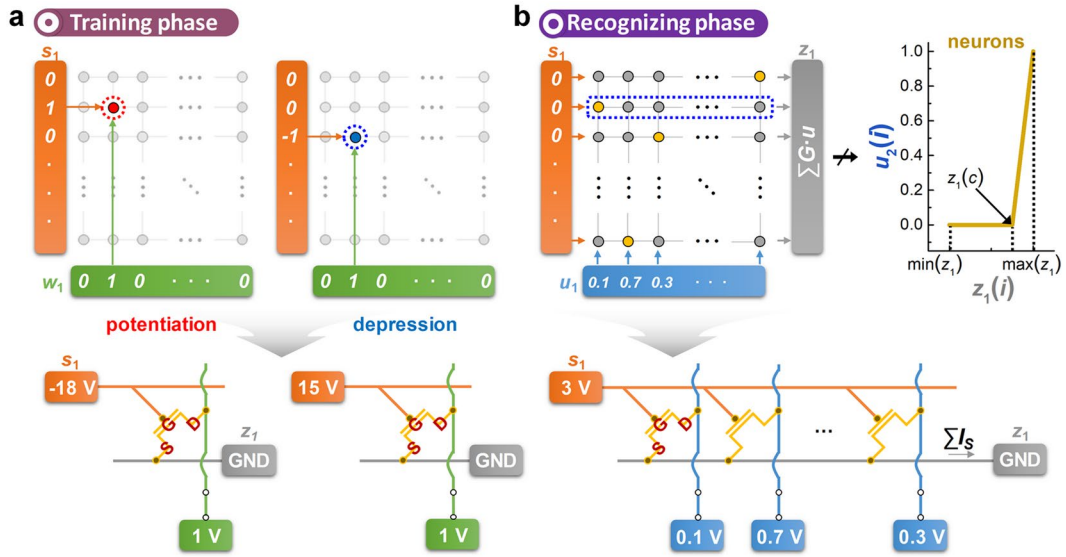

Figure 2. (**a**) The training phase of BNN: $G_1(i, j)$ is updated to G_{high} when $w_1(i)\cdot s_1(i)$ wh1, updated to G_{low} when $w_1(i)\cdot s_1(i)$ wh-1, and maintained its state when $w_1(i)\cdot s_1(i)$ wi0. Each element of $s_1(i)$ and $w_1(i)$ corresponds to V_G and V_D, respectively. (**b**) The recognition phase of BNN: $u_1(i)$ represents V_D, and $z_1(i)$ is utilized for either classifying the input pattern or determining $u_2(i)$.

p le$\gamma\cdot u_1(i)$ (where γ is the learning rate, and $u_1(i)$ is used as a probability value to decide $w_1(i)$). Here, the weight updating of BNN is conducted in a supervised manner. To this end, select-vector $s_1(i)$ le{1 or −1 or 0} directs the training of the input pattern according to its label, where $s_1(i)$ he1, −1, and 0 represent 'potentiation', 'depression', 'no update' of the synaptic weight, respectively. Finally, the resultant outcome of the network is the summation vector $z_1(i)$ given as $\sum_{j=1}^{M} G_1(i, j)u_1(i, j)$, which is the sum of the products of $G_1(i, j)$ and $u_1(i)$ in a row direction. The subsequent $u_2(i)$ and $w_2(i)$ of the next network are determined by passing $z_1(i)$ through the designed neuron function (the detail of the neuron is discussed later).

For the physical implementation of BNN, the GAA silicon nanosheet transistor contributes to a synaptic device, where the embedded charge trap layer (silicon nitride) in the gate dielectric enables adjustable digital-type channel conductance (*i.e.*, synaptic weight modulation). The fabrication process, the device variability, and the digital-type switching performance are discussed in Supplementary Information Note 1. In the configuration of the synaptic transistor array (Fig. 1b), $s_1(i)$ corresponds to the gate voltage (V_G) of the synaptic transistors in a particular row, and either $u_1(i)$ or $w_1(i)$ corresponds to the drain voltage (V_D). The source current of each synaptic transistor (I_S) is determined by the channel conductance (G_{high} or G_{low}) and V_D, and consequently, the integrated I_S of each row ($\sum I_S = \sum G \cdot V_D$) represents $z_1(i)$. Figure 1c shows the implemented test board with an integrated synaptic transistor array, and Fig. 1d shows the microscope images of the synaptic transistors (the array measurement setup using a test board is presented in Supplementary Information Note 2).

BNN has two different modes of operation, *i.e.*, training and recognizing phases. The training phase of BNN to update the synaptic weight (Fig. 2a) is conducted through the cooperation of $w_1(i)$ and $s_1(i)$, which leads to three different consequences: $G_1(i, j)$ is updated to G_{high} when $w_1(i)\cdot s_1(i)$ wh1 (*i.e.*, $w_1(i)$ e1 and $s_1(i)$ a1), updated to G_{low} when $w_1(i)\cdot s_1(i)$ wh-1 (*i.e.*, $w_1(i)$ e1 and $s_1(i)$ a-1), and maintains its state when $w_1(i)\cdot s_1(i)$ ws0 (*i.e.*, $w_1(i)$ .e0 or $s_1(i)$ o0); these are referred to as 'potentiation', 'depression', and 'no update,' respectively. Because the higher learning probability p (($c\gamma\cdot u_1(i)$) leads to $w_1(i)$ becoming 1 more often, the larger $u_1(i)$ results the potentiation/depression of synaptic weight more frequently. In terms of synaptic transistor operation, $s_1(i)$ n {1, −1, 0} corresponds to V_G −18 V, 15 V, and 3 V, respectively. Similarly, $w_1(i)$ S{0, 1} corresponds to V_D 0,floating and 1 V, respectively. Consequently, $w_1(i)\cdot s_1(i)$ o {1, −1, 0} leads to 'increase', 'decrease' and 'maintain' the channel conductance of the synaptic transistor, respectively, according to the configuration of V_G and V_D.

Next, the recognizing phase is conducted by applying $u_1(i)$ to the network instead of $w_1(i)$, as shown in Fig. 2b (since the weight update is not required during the recognizing phase, all $s_1(i)$ are set to 0). The purpose of the recognizing phase is twofold: (1) classification of the input pattern by matching with previously trained patterns, and (2) generation of $u_2(i)$ for transferring the input pattern information to the next network. As mentioned above, $u_1(i)$ involves each pixel of information of the input pattern, and the resultant $z_1(i)$ is the sum of $G_1(i, j)\cdot u_1(i)$ in a row direction. If $z_1(i)$ is the output of the last network, $z_1(i)$ is used to classify the input pattern. The maximum $z_1(i)$ indicates the estimated label for a given input pattern (the detail classification process will be discussed in later). However, when multiple networks are involved in the system, $u_2(i)$ of the next network is generated by exploiting $z_1(i)$. In detail, $u_2(i)$ is determined by passing $z_1(i)$ through the designed neuron function: $u_2(i)$ is zero when $z_1(i)$ i$z_1(c)$, and $u_2(i)$ is increased linearly to 1 when $z_1(i) \geq z_1(c)$. A critical point, $z_1(c)$, is given according to the total number of labels (l) (*e.g.*, l g10 in MNIST dataset, and c ndN/l). Because of the discontinuity of the neuron function, a relatively small value of $z_1(i)$ cannot be delivered to the next network. In other words, only meaningful information (features) of the input pattern can be transferred to the next network, which increases

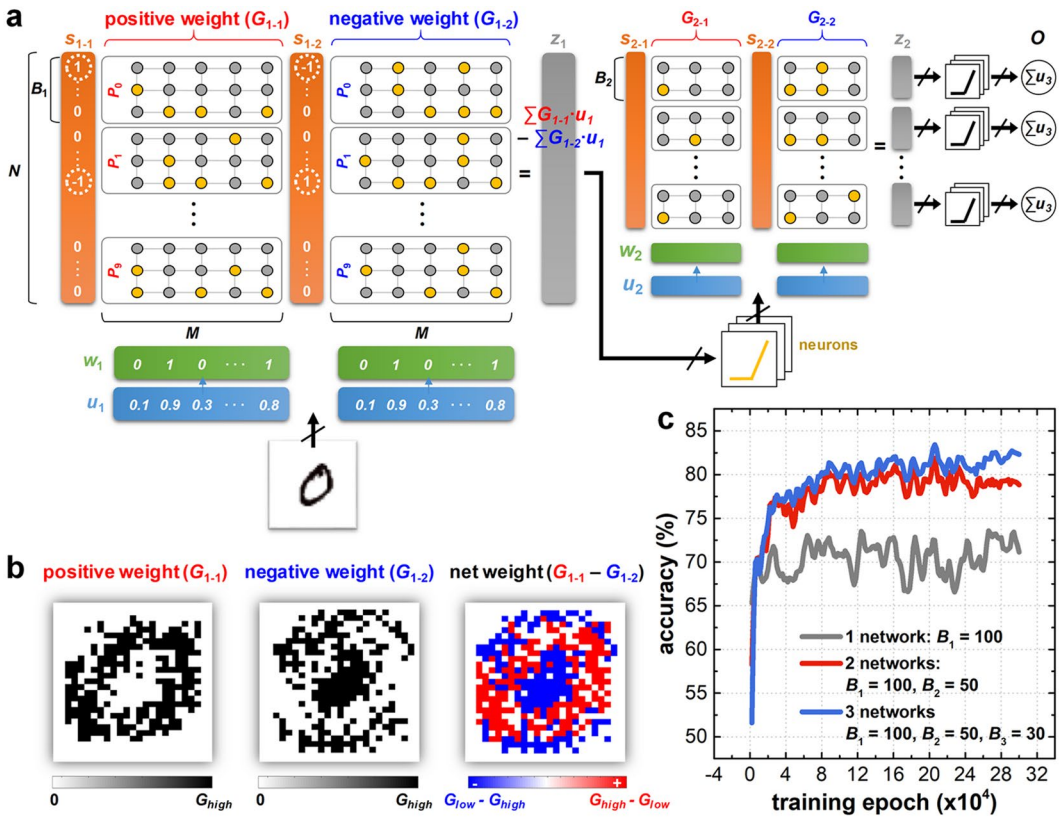

Figure 3. (**a**) Schematic of the network architecture for handwritten digit classification with two networks (G_1 and G_2). Each network is divided into two subnetworks (*e.g.*, $G_{1\text{-}1}$ and $G_{1\text{-}2}$) to represent positive and negative synaptic weights, respectively. This subnetwork is partitioned again to the buckets ($P_0 \sim P_9$), where each bucket is trained on the input patterns according to the label. (**b**) One example of synaptic weights after 60000 times of the training epoch: one row at the bucket P_0 is selected from $G_{1\text{-}1}$ and $G_{1\text{-}2}$, and the resultant $G_{1\text{-}1}$–$G_{1\text{-}2}$ are plotted, respectively. (**c**) The evolution of classification accuracy as a function of the training epoch, which is also affected by the network configuration (*i.e.*, number of networks, bucket size, learning rate). The learning rate γ of all results is 0.2.

the classification accuracy by introducing multiple (deeper) networks. In terms of synaptic transistor operation, $u_1(i)$ corresponds directly to V_D ranged from 0 to 1 V. Then, integrated I_S row by row represents $z_1(i)$.

In the following, the pattern classification ability of BNN is verified by three proof-of-concept examples: the first example is handwritten digit classification (MNIST dataset) verified by the simulation. Figure 3a shows the schematic of BNN including two networks (G_1 and G_2): note that the first network G_1 is divided into two subnetworks, one of which represents a positive weight value ($G_{1\text{-}1}$) and the other that represents a negative weight value ($G_{1\text{-}2}$). Again, $G_{1\text{-}1}$ and $G_{1\text{-}2}$ are partitioned into buckets (depicted as $P_0 \sim P_9$, the size of each bucket is B_1). Each bucket is assigned to train only a specific input pattern according to the label (*e.g.*, digit '0' pattern is only trained at the bucket P_0). Because the total labels (l) of the MNIST dataset are 10, G_1 is accordingly partitioned into 20 buckets and Nar$l \cdot B_1$. Under this configuration, each pixel intensity value of the MNIST dataset (28 × 28 pixels) is rescaled to the range between 0 and 1, which becomes $u_1(i)$ as it is (i (a1 to M, M t784). Then, $w_1(i)$ is given by $u_1(i)$ according to the learning probability p. Next, to generate $s_{1\text{-}1}(i)$ and $s_{1\text{-}2}(i)$ for adjusting the weights properly, the following steps are conducted sequentially (Fig. 3a). Step 1: in $G_{1\text{-}1}$, one row (r_1^{th} row) is randomly selected from the bucket belonging to the label of the input pattern, and $s_{1\text{-}1}(r_1)$ is set to 1. Step 2: in $G_{1\text{-}1}$, another row (r_2^{th} row) is randomly selected from the buckets that do not belong to the label of the input pattern, and $s_{1\text{-}1}(r_2)$ is set to −1. Step 3: all $s_{1\text{-}1}(i)$ except i er$_1$ and r_2 are set to 0. Step 4: $s_{1\text{-}2}(i)$ of $G_{1\text{-}2}$ is given as $-s_{1\text{-}1}(i)$. Following these sequences, a chosen input pattern is trained only in the r_1^{th} row of $G_{1\text{-}1}$ during Step 1. However, since the weight of r_1^{th} row is only potentiated due to $s_{1\text{-}1}(r_1) = -1$, most of the weight will be potentiated if the training phase is repeated continuously. Therefore, during Step 2, the weight of r_2^{th} row of $G_{1\text{-}1}$ should be depressed according to the input pattern. Interestingly, because $s_{1\text{-}2}(i) = -s_{1\text{-}1}(i)$, the bucket of $G_{1\text{-}2}$ is trained oppositely to the bucket of $G_{1\text{-}1}$ during Step 3 and Step 4. For example, digit '0' pattern is trained at the bucket P_0 in $G_{1\text{-}1}$. In contrast, symmetrical P_0 in $G_{1\text{-}2}$ is trained to the features of other digits (*e.g.*, '1' to '9'). Consequently, the resultant $z_1(i)$, defined as $\sum_{j=1}^{M} G_{1-1}(i, j)u_1(i, j) - G_{1-2}(i, j)u_1(i, j)$, contains the feature information of the input pattern corresponding to the label excluding the features other than itself.

The training phase of the second network G_2 is the same as the training phase of G_1. The only difference is, if G_2 is the last network, $z_2(i)$ results in the final output $O(i)$ are given by the sum of the neuronal output over the

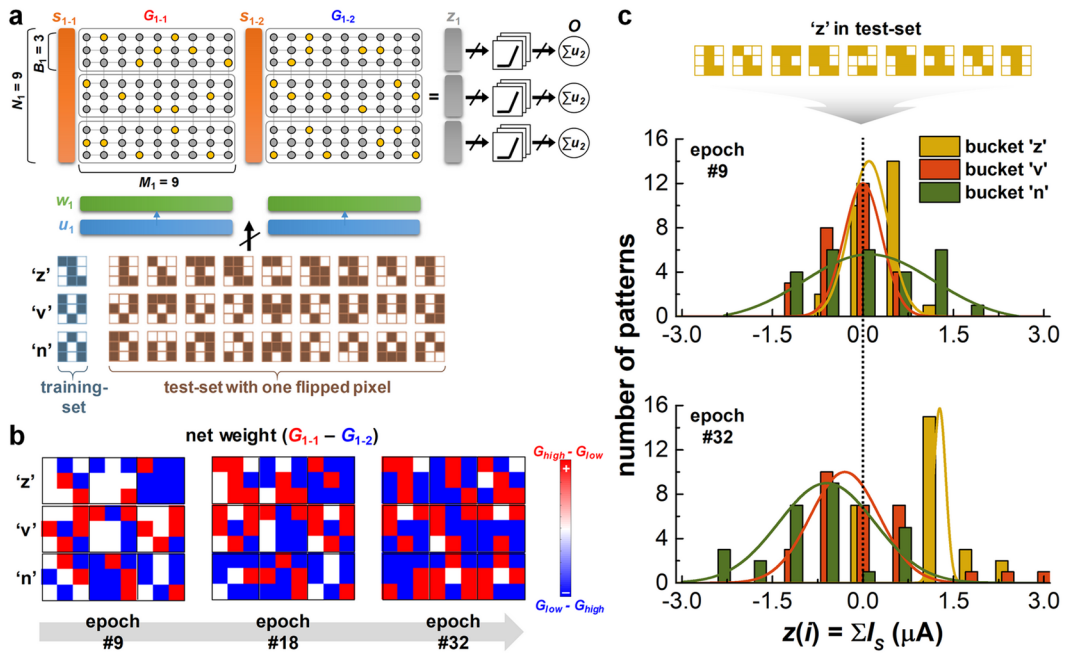

Figure 4. (**a**) Schematic of the network architecture for 3 × 3 binary pattern classification (the letters 'z', 'v', and 'n') with integrated 162 (9 × 9 × 2) synaptic transistors. The training set used in the training phase consisted of 3 correct patterns. The test set used in the recognizing phase (to evaluate the classification accuracy) consisted of 27 patterns with one flipped pixel from the training set. (**b**) The evolution of net synaptic weights ($G_{1\text{-}1}$–$G_{1\text{-}2}$) is a function of the training epoch. (**c**) Measured results of obtained $z(i)$ (*i.e.*, integrated I_S in row direction) when the training epoch is 9 and 32. When the test pattern 'z' is applied to the network during the recognition phase, the resultant $z(i)$ is different from that of each bucket; the $z(i)$ obtained from bucket 'z' is obviously larger than the others.

rows from the bucket of each label. The maximum $O(i)$ designates the estimated label for a given input pattern. Accordingly, the classification accuracy is evaluated regarding agreement between the desired and estimated labels. Figure 3b shows one example of synaptic weights after the training of the MNIST dataset is finished, *i.e.*, one arbitrarily selected row at the bucket P_0 in $G_{1\text{-}1}$ and $G_{1\text{-}2}$. The synaptic weights of $G_{1\text{-}1}$ contain the feature of digit '0' pattern. In contrast, the synaptic weights of $G_{1\text{-}2}$ contain the features of other digits except '0'. The net synaptic weight ($G_{1\text{-}1}$–$G_{1\text{-}2}$) has both positive and negative values, which helps to improve the classification accuracy by emphasizing a distinctive feature of the digit '0' pattern (the impact of negative synaptic weight $G_{1\text{-}2}$ on the classification accuracy is discussed in Supplementary Information Note 3). Finally, the classification accuracy of the MNIST dataset is shown in Fig. 3c as a function of the training epoch, where the number of networks alters the accuracy. With a single network, the accuracy merely reaches approximately 70% with B_1 it 100, while deploying one more network improves the accuracy up to approximately 80% with B_1 it 100, B_2 050. Improvement in the accuracy continues onwards with more networks (*e.g.*, three networks; blue curve in Fig. 3c), although the effect decreases. Additional accuracy tests depending on different parameters (*e.g.*, learning rate or bucket size) are presented in Supplementary Information Note 4.

The second example is the face image (Yale dataset) classification. Because the classification procedure is exactly equal to that of the MNIST dataset discussed above, the results will be discussed in Supplementary Information Note 5. The last example is the experimental demonstration of BNN, where 3 different 3 × 3 binary patterns (denoted as the letters 'z', 'v', 'n')[8] are classified. As shown in Fig. 4a, bucket size B_1 is set to 3 (due to the limit of the fabricated array size), and thus M_1 rr 3 × 3, N_1 , 3 · B_1, the total number of used synaptic transistors is 9 × 9 × 2 + 162 cells. By applying the supervised online training scheme discussed above, Fig. 4b shows the evolution of the weights as a function of training epoch. When the patterns in the training set, *i.e.*, the patterns 'z', 'n', and 'v', are consecutively applied to the network during the training phase, each pattern is trained at the corresponding bucket of the network, which is defined as one training epoch. Then, to evaluate the pattern classification accuracy, the test set patterns (with one flipped pixel from the training set, the total number of patterns in the test set is 27) are applied to the network. Figure 4c shows resultant $z(i)$ in a different training epoch (the data show only when the test pattern 'z' is applied to the network. The data for the test patterns 'v' and 'n' are presented in Supplementary Information Note 6). Note that the $z(i)$ values obtained from each bucket are almost similar when the training epoch is only 9, which means that the test pattern 'z' cannot be classified properly. In contrast, after the training epoch is 32, $z(i)$ obtained from bucket 'z' is much larger than the others, which indicates that the test pattern 'z' can be classified. When the training and recognizing phases are repeated, The classification accuracy is finally reached 100% after 24 times of the application of the training epoch (see Supplementary Information Note 6).

To classify the 3 different 3×3 binary patterns mentioned above, the number of synaptic transistors required in BNN (162 cells) is greater than the number of synaptic devices used in the previous memristor array[8] (60 memristors). However, BNN is believed to be more appropriate for large-scale on-chip implementation due to the high controllability and sustainability of the digital-type conductance switching property, which has already been confirmed by the advanced conventional digital devices. In addition, because the synaptic transistor itself acts as a selector, the chronic problems in memristor crossbar arrays, such as a sneaky current path, can be solved without any further efforts. Moreover, a peripheral driving circuitry, as well as synaptic devices, can also be implemented using the equivalent device technology, which enables a considerably easier full-system integration.

In summary, the binarized neural network is implemented using a gate-all-around silicon nanosheet transistor that exhibits highly reliable and accurately controllable channel conductance modulation in a digital manner. With a supervised online training scheme, pattern classification tasks are experimentally demonstrated. Due to the use of advanced digital device technology, further monolithic integration with neuronal circuits and final brain-like cognitive computing system from an artificial neural network could be realized on a small chip. Considering only a single synaptic device, the demonstrated synaptic transistor in this study may require more energy consumption compared to existing memristors. However, considering the large-scale array of synaptic devices, the energy consumption from the sneaky-current flow will be more critical[37]. However, the existing memristors cannot prevent this problem completely without introducing an additional selector device. In contrast, transistor-based synaptic device arrays can avoid this issue without any further effort, which will certainly be beneficial in terms of system-level energy consumption. Therefore, the binarized neural network can provide the breakthrough for the device-level of the present neuromorphic system research based on analog-manner synaptic devices and enable us to provide a novel direction and inspiration for neuromorphic engineering in the future.

References

1. Bi, G. Q. & Poo, M. M. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* **18**, 10464–10472 (1998).
2. Mead, C. Neuromorphic Electronic Systems. *Proc. IEEE* **78**, 1629–1636 (1990).
3. Indiveri, G. & Horiuchi, T. K. Frontiers in neuromorphic engineering. *Front. Neurosci.* **5**, 118 (2011).
4. Yu, S. Neuro-Inspired Computing With Emerging Nonvolatile Memorys. *Proc. IEEE* **106**, 260–285 (2018).
5. Jo, S. H. *et al*. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* **10**, 1297–1301 (2010).
6. Kuzum, D., Yu, S. & Philip Wong, H.-S. Synaptic electronics: materials, devices and applications. *Nanotechnology* **24**, 382001 (2013).
7. Yang, J. J., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. *Nat. Nanotechnol.* **8**, 13–24 (2013).
8. Prezioso, M. *et al*. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61–64 (2015).
9. Zidan, M. A., Strachan, J. P. & Lu, W. D. The future of electronics based on memristive systems. *Nat. Electron.* **1**, 22–29 (2018).
10. Guo, X. *et al*. Modeling and Experimental Demonstration of a Hopfield Network Analog-to-Digital Converter with Hybrid CMOS/Memristor Circuits. *Front. Neurosci.* **9**, 488 (2015).
11. Choi, S., Shin, J. H., Lee, J., Sheridan, P. & Lu, W. D. Experimental Demonstration of Feature Extraction and Dimensionality Reduction Using Memristor Networks. *Nano Lett.* **17**, 3113–3118 (2017).
12. Sheridan, P. M. *et al*. Sparse coding with memristor networks. *Nat. Nanotechnol.* **12**, 784–789 (2017).
13. Du, C. *et al*. Reservoir computing using dynamic memristors for temporal information processing. *Nat. Commun.* **8**, 2204 (2017).
14. Jeong, Y., Lee, J., Moon, J., Shin, J. H. & Lu, W. D. *K* -means Data Clustering with Memristor Networks. *Nano Lett.* **18**, 4447–4453 (2018).
15. Zidan, M. A. *et al*. A general memristor-based partial differential equation solver. *Nat. Electron.* **1**, 411–420 (2018).
16. Burr, G. W. *et al*. Experimental Demonstration and Tolerancing of a Large-Scale Neural Network (165 000 Synapses) Using Phase-Change Memory as the Synaptic Weight Element. *IEEE Trans. Electron Devices* **62**, 3498–3507 (2015).
17. Yu, S. *et al*. Scaling-up resistive synaptic arrays for neuro-inspired architecture: Challenges and prospect. in *Technical Digest - International Electron Devices Meeting, IEDM* 17.3.1–17.3.4 (IEEE, 2015), https://doi.org/10.1109/IEDM.2015.7409718.
18. Querlioz, D., Bichler, O., Dollfus, P. & Gamrat, C. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* **12**, 288–295 (2013).
19. Yu, S. *et al*. A Low Energy Oxide-Based Electronic Synaptic Device for Neuromorphic Visual Systems with Tolerance to Device Variation. *Adv. Mater.* **25**, 1774–1779 (2013).
20. Kim, S., Lim, M., Kim, Y., Kim, H.-D. & Choi, S.-J. Impact of Synaptic Device Variations on Pattern Recognition Accuracy in a Hardware Neural Network. *Sci. Rep.* **8**, 2638 (2018).
21. Jeong, Y., Zidan, M. A. & Lu, W. D. Parasitic Effect Analysis in Memristor-Array-Based Neuromorphic Systems. *IEEE Trans. Nanotechnol.* **17**, 184–193 (2018).
22. Strukov, D. B. & Williams, R. S. Exponential ionic drift: fast switching and low volatility of thin-film memristors. *Appl. Phys. A* **94**, 515–519 (2009).
23. Wang, Z. *et al*. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat. Mater.* **16**, 101–108.
24. Kim, S., Choi, S. & Lu, W. Comprehensive physical model of dynamic resistive switching in an oxide memristor. *ACS Nano* **8**, 2369–2376 (2014).
25. Alibart, F., Gao, L., Hoskins, B. D. & Strukov, D. B. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. *IOP Publ. Nanotechnol. Nanotechnol.* **23**, 75201–7 (2012).
26. Gao, L., Alibart, F. & Strukov, D. B. Analog-input analog-weight dot-product operation with Ag/a-Si/Pt memristive devices. in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip* (*VLSI-SoC*) 88–93 (IEEE, 2012), https://doi.org/10.1109/VLSI-SoC.2012.7332082.
27. Gao, L., Chen, P.-Y. & Yu, S. Programming Protocol Optimization for Analog Weight Tuning in Resistive Memories. *IEEE Electron Device Lett.* **36**, 1157–1159 (2015).
28. Huang, C.-H., Chang, W.-C., Huang, J.-S., Lin, S.-M. & Chueh, Y.-L. Resistive switching of Sn-doped In_2O_3/HfO_2 core–shell nanowire: geometry architecture engineering for nonvolatile memory. *Nanoscale* **9**, 6920–6928 (2017).
29. Duran Retamal, J. R., Ho, C.-H., Tsai, K.-T., Ke, J. & He, J.-H. Self-Organized Al Nanotip Electrodes for Achieving Ultralow-Power and Error-Free Memory. *IEEE Trans. Electron Devices* **66**, 938–943 (2019).
30. Le, V.-Q. *et al*. Van der Waals heteroepitaxial AZO/NiO/AZO/muscovite (ANA/muscovite) transparent flexible memristor. *Nano Energy* **56**, 322–329 (2019).
31. Monzio Compagnoni, C. *et al*. Reviewing the Evolution of the NAND Flash Technology. *Proc. IEEE* **105**, 1609–1633 (2017).
32. Kang, D. *et al*. 256 Gb 3 b/Cell V-NAND Flash Memory With 48 Stacked WL Layers. *IEEE J. Solid-State Circuits* **52**, 210–217 (2017).
33. Peter R., H. Synaptic density in human frontal cortex - Developmental changes and effects of aging. *Brain Res.* **163**, 195–205 (1979).

34. Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).
35. Belhumeur, P. N., Hespanha, J. ∼P. & Kriegman, D. J. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE transactions on pattern analysis and machine intelligence* **19** (1997).
36. Kim, G. *et al.* Markov Chain Hebbian Learning Algorithm with Ternary Synaptic Units. *IEEE Access* **7**, 10208–10223 (2019).
37. Kim, S., Zhou, J. & Wei, D. L. Crossbar RRAM Arrays: Selector Device Requirements During Write Operation. *IEEE Trans. Electron Devices* **61**, 2820–2826 (2014).

Acknowledgements

This research was supported by the Nano·Material Technology Development Program (NRF-2016M3A7B4910430) and the Basic Science Research Program (NRF-2019R1A2C1002491, 2019R1A2B5B01069988, and 2016R1A5A1012966) through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning. This work was partially supported by the Future Semiconductor Device Technology Development Program (Grant 10067739) funded by the MOTIE (Ministry of Trade, Industry & Energy) and the KSRC (Korea Semiconductor Research Consortium).

Author Contributions

The manuscript was prepared by S.K. and S.-J.C. Device fabrication was prepared by B.C., J.Y., Y.L. and M.-H.K. Measurement and simulation were performed by S.K.

Additional Information

Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-019-48048-w.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.