

Article

Memristor-CMOS Hybrid Circuit for Temporal-Pooling of Sensory and Hippocampal Responses of Cortical Neurons

Tien Van Nguyen , Khoa Van Pham  and Kyeong-Sik Min * 

School of Electrical Engineering, Kookmin University, Seoul 02707, Korea; tiennv@kookmin.ac.kr (T.V.N.); khoapv@kookmin.ac.kr (K.V.P.)

* Correspondence: mks@kookmin.ac.kr

Received: 24 February 2019; Accepted: 13 March 2019; Published: 15 March 2019



Abstract: As a software framework, Hierarchical Temporal Memory (HTM) has been developed to perform the brain's neocortical functions, such as spatial and temporal pooling. However, it should be realized with hardware not software not only to mimic the neocortical function but also to exploit its architectural benefit. To do so, we propose a new memristor-CMOS (Complementary Metal-Oxide-Semiconductor) hybrid circuit of temporal-pooling here, which is composed of the input-layer and output-layer neurons mimicking the neocortex. In the hybrid circuit, the input-layer neurons have the proximal and basal/distal dendrites to combine sensory information with the temporal/location information from the brain's hippocampus. Using the same crossbar architecture, the output-layer neurons can perform a prediction by integrating the temporal information on the basal/distal dendrites. For training the proposed circuit, we used only simple Hebbian learning, not the complicated backpropagation algorithm. Due to the simple hardware of Hebbian learning, the proposed hybrid circuit can be very suitable to online learning. The proposed memristor-CMOS hybrid circuit has been verified by the circuit simulation using the real memristor model. The proposed circuit has been verified to predict both the ordinal and out-of-order sequences. In addition, the proposed circuit has been tested with the external noise and memristance variation.

Keywords: memristor-CMOS hybrid circuit; temporal pooling; sensory and hippocampal responses; cortical neurons; hierarchical temporal memory; neocortex

1. Introduction

The neocortex occupying most of the brain's surface area has been believed to perform the most human-like functions such as intelligence, cognition, etc. among all human organs. It is just 2.5-mm thick and is composed of six layers [1–3]. All six neocortical layers have the same columnar architecture, where the neocortical neurons are connected in both the vertical and horizontal directions to form various feedback and feedforward paths to communicate with each other. Anatomical experiments have observed the columnar architecture consistently through the entire neocortex [4,5]. This fact may hint that there is a canonical neural circuitry that can describe various neocortical functions with one model [6].

In this paper, we try to develop a memristor-CMOS hybrid circuit that can emulate the neocortex's canonical neural circuitry by combining nanoscale memristor crossbars with CMOS peripheral circuits. Memristors have been studied intensively for many years for their possible use of neuromorphic hardware since the first experimental demonstration [7,8]. This is because the memristive behavior seems very similar with the biological synaptic plasticity, where the synaptic connection can be strengthened and weakened dynamically according to the sensory stimulus [9]. The ionic dynamics of memristors can also be used in implementing the reservoir computing hardware, where the cognitive

function can be processed simply by applying the time-domain signals to the memristor-based reservoir [10]. Moreover, the memristor crossbars can be built in a 3-dimensional architecture by a CMOS-compatible fabrication process, where the 3-dimensionality is very similar to the anatomical view of the real biological neuron-synapse connections in the neocortex [11,12]. Also, the memristor crossbar can perform a bitwise parallel operation which has been thought as one of the key aspects of energy-efficient computing of the human brain's cognition, compared to modern state-of-the-art computers [13,14].

As a software framework for modeling the neocortical function, Hierarchical Temporal Memory (HTM) has been developed recently [15–20]. Figure 1a shows a functional block diagram of HTM that is composed of the Spatial Pooler (SP) and Temporal Memory (TM). SP receives the sensory information to learn the cortical representation. As a result, SP generates Sparse Distributed Representation (SDR) [16]. SDR is a mathematical description for representing the cortical neurons that may be activated or deactivated in response to the sensory information from the cochlea, retina, etc. Actually, SP was proposed as a software algorithm in the HTM software framework [15–20]. To realize the spatial pooling with hardware, we developed the spatial-pooling memristor crossbar circuit in a previous work, where the circuit could convert the sensory information to the SDR that meant the representation of cortical neurons [21].

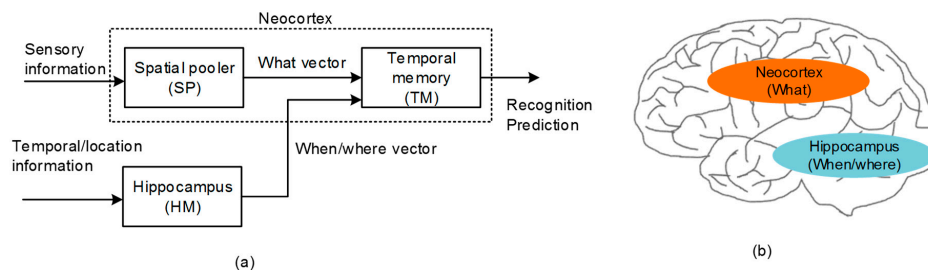


Figure 1. (a) The functional block diagram of Hierarchical Temporal Memory (HTM): The spatial pooler receives the sensory information from various sensory organs and forms the Sparse Distributed Representation (SDR) output representing the collective cortical neurons activated in response to the sensory information. The temporal memory learns the sequence of items that are represented by the SDR vectors by combining the sensory information with the temporal information. (b) The cross-sectional view of the human brain: Here, the neocortex and hippocampus regions are shown for processing the “what” and “when/where” information, respectively.

The temporal memory (TM) in Figure 1a puts together the “what” and “when/where” vectors that come from the spatial pooler and hippocampus model, respectively [22,23]. By combining the “what” with “when/where”, the temporal memory can perform both the spatial recognition and the temporal prediction. For the temporal prediction, the representation of temporal succession (“when”) should be segregated from the representation of the content (“what”), as shown in Figure 1a [24,25]. From the experimental observations, the hippocampus has been known to play a central role in encoding the information of ordinal sequences (“when”) [25,26], whereas the representation of the content (“what”) has been known to come from the neocortex, as indicated in Figure 1b.

The representation of the temporal sequence (“when”) can be extended to the spatial sequence (“where”) [27,28]. For example, the order of the words during reading depends on where one is looking (“where”). However, the order of the words during listening can be interpreted as the temporal sequence (“when”). Actually, every principal neuron in the hippocampus can work as either a “place cell” or “time cell” [29]. By doing so, the hippocampus can model both the temporal (“when”) and spatial (“where”) information with the same kinds of representation. Thus, we can think that the spatial sequence of location information is one case of the temporal sequence [26].

Though HTM has been developed as the software framework for performing the neocortex's cognition, it should be realized with hardware not only to mimic the neocortex's function but also to exploit its architectural benefit. One reason for this need of a hardware version is the demand of the

edge-computing devices in the Internet of Things (IoT) era [30–32]. For the near-sensor processing and computing of IoT devices, the speed and power benefit due to the bitwise parallel-processing of memristor crossbars can be very important in terms of the possibility of real-time and on-chip cognitive functions for various edge-computing applications [32]. Thus, the neocortex’s cognitive function combined with the crossbar’s architectural merit can accelerate the transition from the HTM software framework to its hardware emulator [30].

To implement HTM by hardware not software, in this paper, we propose a new memristor-CMOS hybrid circuit for realizing the temporal-pooling function of the human brain, which is composed of the input and output layers, to mimic the temporal prediction of neocortical neurons. In the hybrid circuit, the input layer has proximal and basal/distal dendrites to combine the sensory information with the temporal/location information. The output layer composed of the same circuitry with the input layer can perform a prediction by integrating the temporal information through the basal/distal dendrites. In this paper, the input and output layers realized with the memristor-CMOS hybrid circuit are verified to perform the temporal recognition and prediction that are the same functions within the human brain’s neocortex.

2. Proposed Methods

Memristor crossbars are thought to be very suitable in mimicking the anatomical and functional architecture of neocortex. Memristive behaviors seem similar with the synaptic plasticity of biological neurons. Moreover, the 3-D connectivity of crossbars can be useful in realizing the real neuronal 3-D architecture of the neocortex. Also, the crossbars can perform a bitwise parallel computation, as the pyramidal neurons do in the neocortical layers. To develop the neocortex-mimicking memristor crossbar, first, we need to understand the functional model of neocortical columns and layers [33,34].

Figure 2a shows the conceptual model of temporal memory composed of input-layer and output-layer neurons [23]. From previous experimental observations, the HTM theory deduced a couple of rules to describe the neocortex’s operation. First, it is assumed that the input-layer neurons receive the sensory information through the single proximal dendrite [23]. The synapses connected to this proximal dendrite are involved in only local signal-processing, as shown in Figure 2a. They do not communicate with the neurons outside the local region. The proximal dendrite is more likely to form short-distance vertical connections to receive the sensory information. On the contrary, the basal/distal dendrite is responsible for long-distance horizontal communication [23]. The dendrite can receive information from distantly located regions such as the hippocampus. One thing to note is that one neocortical neuron is allowed to have only single proximal dendrite. However, the basal/distal can have multiple.

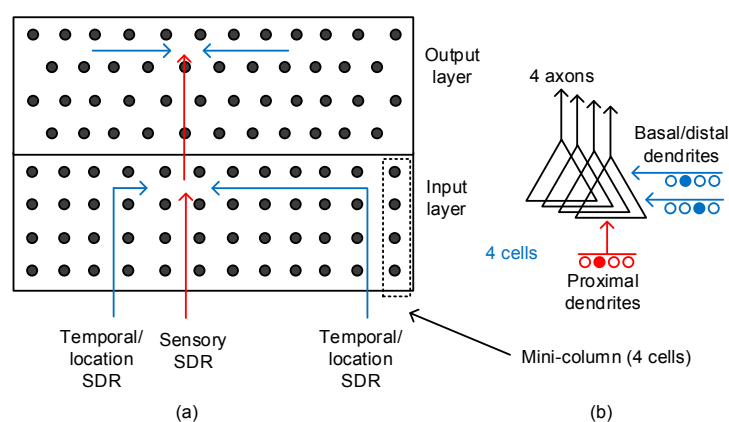


Figure 2. (a) The conceptual model of temporal memory architecture: The red and blue lines represent the proximal and basal/distal dendrites, respectively. (b) The schematic of a pyramidal neuron with a single proximal dendrite and multiple basal/distal ones. The number of output axons can be multiple too. Here, we showed 4 axons to constitute one mini-column with 4 cells. The pyramidal neurons are known as the majority of neocortical neurons.

Figure 2a also shows the output-layer neurons that are basically the same as the input-layer ones. The proximal dendrite is for short-and-direct connections from the input-layer neurons. The output-layer neurons can receive long-distance information horizontally through multiple basal/distal dendrites for the temporal integration of “when” vectors. The two-layer model is regarded as a general feature of the neocortex and can be used as an elemental unit in realizing the memristor-based temporal-pooling crossbar [23].

Figure 2b shows the schematic of a pyramidal neuron that incorporates the axonal and dendritic connections. The proximal dendrite receives the direct feed-forward inputs from the sensory organs. The basal/distal dendrite can be driven by the long-distance signals from far away regions, such as the hippocampus.

Figure 3 shows the conceptual schematic of the temporal-pooling memristor crossbar composed of input-layer and output-layer neurons. The input-layer neurons receive sensory SDR and temporal/location SDR from the spatial pooler and hippocampus model, respectively. The sensory SDR vectors are connected with the proximal dendritic synapses. The basal/distal dendrites of the input-layer neurons receive hippocampal responses that contain the temporal and location information.

The output-layer neuron in Figure 3 has the same circuitry as the input-layer neuron, as shown in Figure 2b. The proximal connection of the output-layer neuron comes from the axonal output of an input-layer neuron. The basal/distal dendrite can make the output-layer neuron a predicted state by depolarizing its body. If the body is depolarized enough by the previous basal/distal dendritic inputs, it can fire spikes sooner than the other output-layer neurons, if they are not in the predicted state. If the output-layer neuron is not in the predicted state, it cannot fire spikes, even though it receives the same feedforward input as the predicted-state neuron. Only the predicted-state neuron which is depolarized already can fire spikes in response to the proximal dendritic input.

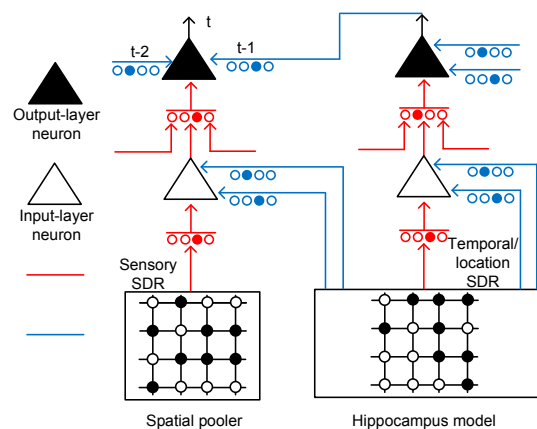


Figure 3. The conceptual schematic of a temporal-pooling memristor crossbar composed of input-layer and output-layer neurons: The input-layer neuron receives sensory SDR and temporal/location SDR from the spatial pooler and hippocampus model, respectively. The sensory and temporal/location SDR are generated from the spatial-pooling memristor crossbar that was developed in a previous work [21]. The output-layer neuron can perform a prediction by integrating the temporal information through multiple basal/distal dendrites.

In Figure 4a, we propose a memristor-CMOS hybrid circuit that has the input and output layers for the temporal-pooling of sequences such as words, sentences, etc. Here, the sensory SDR vectors enter the proximal dendrites of m_0, m_1, m_2 , etc. The temporal/location SDR vectors are connected to the basal/distal dendrites of m_3, m_4, m_5 , etc. One thing to note in Figure 4a is that each neuron is allowed to have only a single proximal dendrite. However, for the basal/distal ones, the neuron can have multiple dendrites, as explained in Figure 2. The sensory SDR and temporal/location SDR are collectively received by the input-layer neurons. The column current of the sensory SDR “A” is delivered to C_0 , where the column current is converted to a voltage and then compared with the

threshold. The detailed schematic of C_0 is shown in Figure 4b. Similarly, the column current of “B” is delivered to C_1 . The temporal/location SDR vectors of “#1”, “#3”, and “#2” generate the row currents which are delivered to C_2 , C_3 , and C_4 , respectively. A_0 , A_1 , and A_2 are the AND gates that combine the sensory information of “A” with the temporal/location SDRs of “#1”, “#3”, and “#2”, respectively. The outputs of A_0 , A_1 , and A_2 are represented with i_0 , i_1 , and i_2 , respectively. They enter the pulse-type set-reset latches of L_0 , L_1 , and L_2 , respectively. The pulse-type set-reset latch is shown in Figure 4c. L_0 can be set if the SDR “A” and SDR “#1” are recognized at the same time. L_1 is set for “A” and “#3”. L_2 is switched to the SET state for “A” and “#2”. Similarly, L_3 , L_4 , and L_5 can respond to the input SDR of “B#1”, “B#3”, and “B#2”, respectively. The set-reset latch in Figure 4c is reset by the delayed version of the “EOW_P” pulse from the delay line τ . Here, “EOW_P” means the pulse indicating the end of the word. “EOW_P” is generated when the word ends.

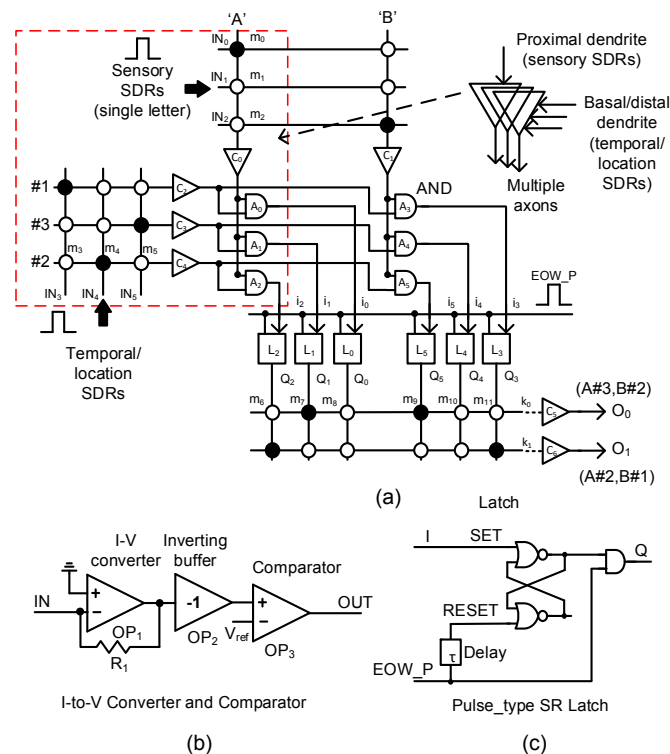


Figure 4. (a) The schematic of the proposed memristor-CMOS hybrid circuit for the temporal pooling of sequences such as words, sentences, etc: The input layer is composed of the memristor crossbars for sensory and temporal/location SDRs, the current-to-voltage converters, comparators, the AND gates, etc. The output layer is composed of the memristor crossbars, converters, comparators, latches, etc. (b) The schematic of the current–voltage converter and comparator and (c) the schematic of the pulse-type set-reset latch.

As mentioned earlier, if the sensory SDR of letter “A” and the location SDR of “#3” are applied to the input-layer neuron, Q_1 is activated. Similarly, when the sensory SDR of “B” and the location SDR of “#2” are recognized, Q_5 becomes high. When “EOW_P” is activated, the two latches of L_1 and L_5 keep Q_1 and Q_5 high, respectively, until the reset. Assuming that the dendritic synapses of the output neuron O_0 are already put in the predicted state with “A#3” and “B#2”, m_7 and m_9 are already programmed LRSs (Low Resistance States) as a result of crossbar training. Here, the solid and open circles represent LRS and HRS (High Resistance State), respectively. At end-of-word, if the row current of k_0 is larger than the output-layer neuron’s threshold, O_0 becomes high. Actually, we can think that the k_0 current represents the integration of temporal responses to the sensory/location SDRs of “A#3” and “B#2” because “A#3” and “B#2” were already recognized at the previous time. Similarly, if “A#2” and “B#1” are recognized one by one, the row current k_1 becomes larger than the threshold and can activate O_1 .

Figure 5a shows a current–voltage relationship of the measured memristor that was obtained by a Keithley-4200 (Semiconductor Characterization System, Tektronix, Inc., Beaverton, OR, USA) [35]. The measured memristor’s film is a Pt/LaAlO₃/Nb-doped SrTiO₃ stacked layer [35]. Here, the LRS and HRS were measured as 10 kΩ and 1 MΩ, respectively. The black line in Figure 5a represents the behavioral model of memristors [35]. The measured data are represented with the red line. The behavioral model described by Verilog-A was used in the circuit simulation of the hybrid circuits of memristors and CMOS in this paper. Here, the circuit simulation was performed using CADENCE SPECTRE (Cadence Design Systems, Inc., San Jose, CA, USA) and SAMSUNG 0.13-μm circuit simulation parameters [36]. The mathematical equations of the Verilog-A model of memristors were explained in a previous publication in detail [35].

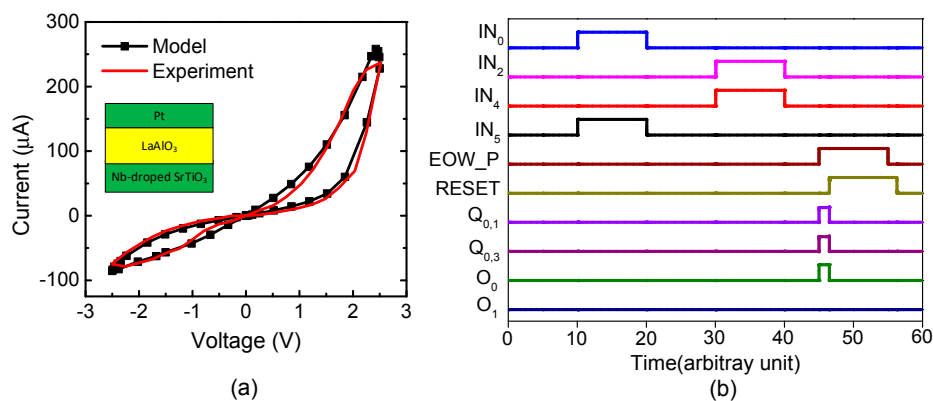


Figure 5. (a) The current–voltage relationships of memristors for the measurement and Verilog-A model: The black line represents the Verilog-A model of memristors used in the circuit simulation in this paper [35]. The red line is for the measurement [35]. The details of the measurement and the Verilog-A model were explained well in a previous publication [35]. (b) The waveforms of the proposed memristor-CMOS hybrid circuit for temporal pooling shown in Figure 4.

Figure 5b shows the waveforms of Figure 4 obtained from the CADENCE (Cadence Design Systems, Inc., San Jose, CA, USA) circuit simulation with the memristor’s Verilog-A model in Figure 5a and the SAMSUNG 0.13-μm SPICE parameters. First, we assumed the sensory SDR of letter “A” and the location SDR of “#3” are generated by the spatial pooler. As a result, the IN₀ and IN₅ pulses are high, while the others are low in Figure 5b. By doing so, Q₁ becomes high. Second, if the spatial pooler generates the sensory SDR of letter “B” and the location SDR “#2”, Q₅ becomes high. At end-of-word, the pulse of “EOW_P” is enabled and the output neuron O₀ becomes active. Here, the output neuron is already put in the predicted state by the previous signals of Q₁ and Q₅. After the output neuron O₀ fires a pulse, O₀ returns to low, as the typical integrate-and-fire neuron acts. To do so, the “EOW_P” pulse goes through the delay line τ and its delayed pulse resets the set-reset latches. The integrate-and-fire operation is realized very simply using the digital CMOS gates and the memristor crossbar, as shown in Figure 4a–c.

3. Results

In this paper, we tested the proposed memristor-CMOS hybrid circuit of temporal pooling in Figure 4a with an EMNIST (Extension of Modified National Institute of Standards and Technology) data-set of handwritten letters [37]. For training the memristor crossbar to recognize EMNIST handwritten letters, we applied the simple Hebbian learning to 26 EMNIST letters from “a” to “z”. The operational steps of simple Hebbian learning of memristor crossbars is shown in Figure 6. Here, first, we initialized the memristor crossbar. Second, we calculated the amount of overlap between the input vector and the crossbar’s column or row. If the crossbar’s column or row has an overlap larger than the threshold, the column or row is activated. In this case, the permanence values of matched and unmatched memristor cells belonging to the activated column or row are increased and decreased

according to the predetermined parameter Δ , respectively. If the permanence value becomes larger than 1 or less than 0, the memristor corresponding to the permanence is strengthened or weakened according to the memristor programming circuit. Here, we used the typical $V_{DD}/2$ scheme for programming memristors. One thing to note is that the memristor programming based on Hebbian learning does not need the complicated backpropagation calculation [21]. By doing so, the proposed memristor-CMOS hybrid circuit can be very suitable to online learning because the hardware complexity of Hebbian learning is much simpler than that of a backpropagation-based system.

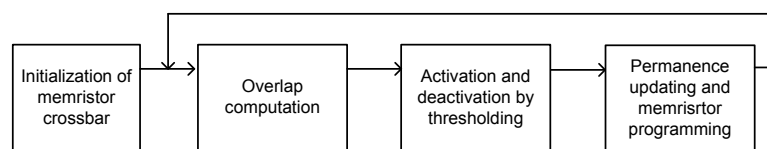


Figure 6. The operational steps of simple Hebbian learning of memristor crossbars: initialization, overlap computation, activation and deactivation by thresholding, and permanence updating and memristor programming. Here the memristor programming based on Hebbian learning does not need the complicated backpropagation calculation.

In this test, the 26 EMNIST letters have 60,000 training vectors. Each image is composed of 20×20 gray pixels. To estimate the recognition rate, we tested 10,000 execution vectors of an EMNIST letter. The first row in Figure 7 shows 4 images of EMNIST letters. They are “c”, “o”, “m”, and “e”, respectively. EMNIST vectors are randomized first and then applied to the spatial-pooling memristor crossbar proposed in a previous work [21]. The second row in Figure 7 shows the randomized version of the EMNIST vectors. It should be noted that the memristor-CMOS hybrid circuit does not need to use the complicated random number generation circuit. Once we decided the randomization function, we applied the same function to all the test vectors without changing it for every vector [21]. Thus, we did not use the random number generator circuit in a previous work [21].

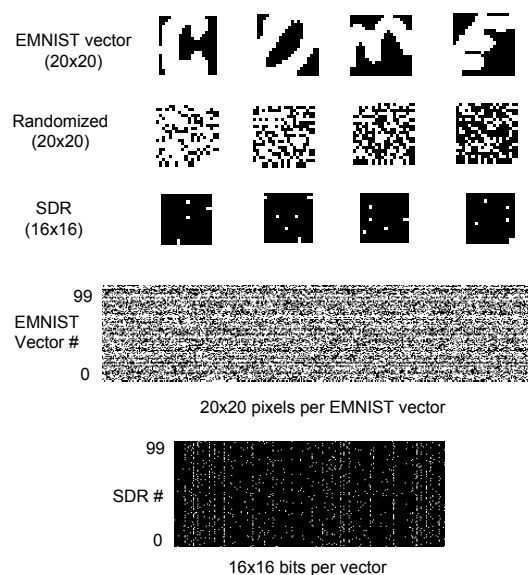


Figure 7. The first row shows the EMNIST handwritten letters of “c”, “o”, “m”, and “e”, respectively. The second row shows randomized images of EMIST handwritten letters. The third row are the SDRs that are obtained from the spatial-pooling memristor crossbar for the randomized images of “c”, “o”, “m”, and “e”. The fourth row shows 100 EMNIST input vectors. Each EMNIST vector is composed of 20×20 pixels. The fifth row shows 100 SDRs with 16×16 bits which are obtained from 100 EMNIST input vectors with 20×20 pixels. Among the 16×16 bits, only 2% of the bits become active to maintain the sparsity ratio around 2% by spatial-pooling for EMNIST vectors [21].

If we perform the spatial pooling with 256 columns, we can obtain 16×16 SDRs from 20×20 EMNIST input vectors. The third row in Figure 7 shows the SDRs that are obtained from the spatial-pooling memristor crossbar for the randomized images of “c”, “o”, “m”, and “e”, respectively. The fourth row in Figure 7 shows the pixel map of 100 EMNIST test vectors. The average sparsity of the EMNIST test vectors is as high as 55.8%; that means 55.8% of the pixels can be white. On the contrary, the SDRs from the spatial-pooling crossbar have a sparsity as low as 2%. The fifth row shows 100 SDRs with 16×16 bits. Among the 16×16 bits of each SDR, only 2% of the bits become active by the spatial-pooling of the 20×20 -pixel EMNIST vector. This low sparsity of SDRs is very useful in cognitive computations such as union, pattern matching, etc. [38]. In addition, the small number of active bits can reduce the number of LRS cells in a memristor crossbar. By doing so, the power consumption and sneak-leakage problem can be improved much in the spatial-pooling crossbar [39].

To test the temporal-pooling memristor-CMOS hybrid circuit in Figure 4a, we put together EMNIST handwritten letters to form arbitrary words. Figure 8 shows the recognition rate of the proposed temporal-pooling circuit for the 40 words tested in this paper. The recognition rate of words is estimated as high as 95.6%, 99.1%, and 99.3% for 256-bit SDRs, 1024-bit SDRs, and 4096-bit SDRs, respectively. One thing to note is that the recognition rate of words is much better than the recognition rate of EMNIST letters. This is because the temporal-pooling circuit interprets both sensory and temporal/location information together, as indicated in Figure 2. Combining the sensory SDRs with the location SDRs makes the recognition of words better than the recognition of letters. Figure 8 also shows the recognition rate of sentences as high as the rate of words. The recognition rate of sentences is simulated 96.5%, 99.3%, and 99.7% for 256-bit SDRs, 1024-bit SDRs, and 4096-bit SDRs, respectively. Here, the number of sentences tested in Figure 8 is 10.

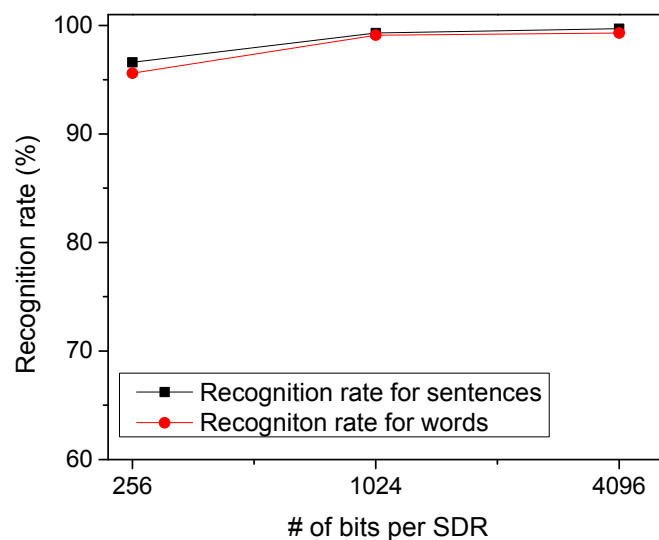


Figure 8. The recognition rate of words and sentences with varying the number of bits per SDR.

Figure 9 shows the recognition rate by varying the amount of noise added to SDRs. The noise is added by randomly flipping a fraction of the active bits to inactive, and vice versa so that the sparsity can be maintained constant. As shown in Figure 4a, the temporal-pooling circuit receives both the sensory and location SDRs from the spatial pooler. Here, the red circles represent the recognition rate for the noise added to the sensory SDRs. The black boxes are for the noise added to the location SDRs. From this figure, the noise added to the location SDRs seems more critical in terms of recognition rate. If a noise as large as 40% is added to the location SDRs, the recognition rate becomes as low as 45.3%. However, the rate can be as high as 92.5% for the same amount of noise added to the sensory SDRs.

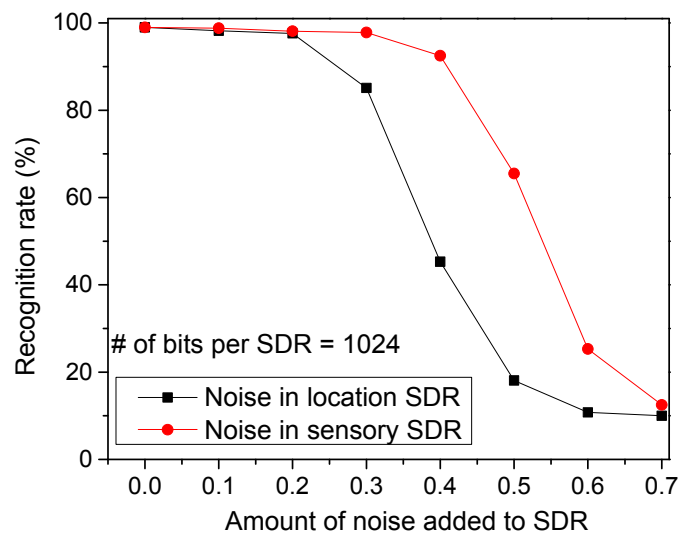


Figure 9. The recognition rate of words by varying the amount of noise added to location SDRs and sensory SDRs: The red circles represent the recognition rate for the noise added to the sensory SDRs. The black boxes are for the noise added to the location SDRs.

In Figure 10, we assumed the statistical distribution of LRS and HRS with the memristance variation = 10%, as shown in the inset figure. Here, the main figure shows the recognition rate by varying the amount variation in memristance from 0% to 15%. Here, the median values of HRS and LRS are assumed as 1 MΩ and 10 KΩ, respectively. Though the variation is as large as 15%, the recognition rate is still as high as 85.9%. The loss of recognition rate for the variation = 15% is only as small as 13.2% compared to the variation = 0%.

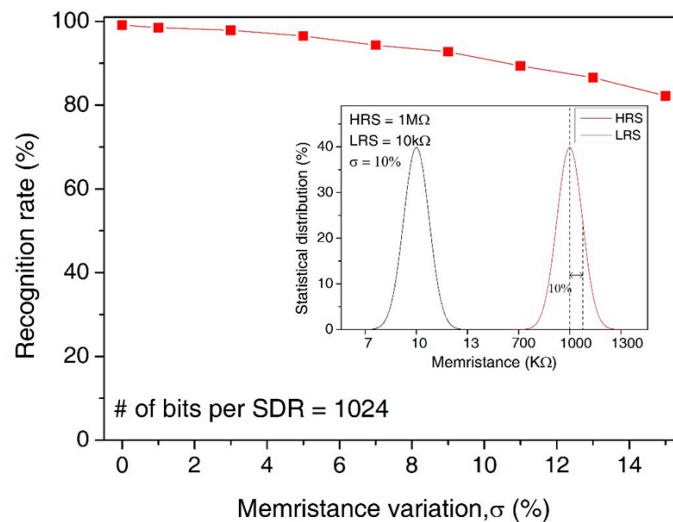


Figure 10. The recognition rate of words by increasing the percentage variation in memristance from 0% to 15%: The inset figure shows the statistical distribution of LRS and HRS for the memristance variation = 10%.

Figure 11 shows the prediction rate of sentences by increasing the number of words sensed in the tested sentence. Here, we tested two cases of sequences which are ordinal and out-of-order sequences, respectively. First, let us explain the ordinal sequence. Assume that we try to recognize two sequences of “A-B-C-D-E” and “A-B-C-E-D”. Here, the first three SDRs are “A-B-C” which are the same for both sequences. Also, the fourth and fifth SDRs are different each other. If the first SDR of “A” comes to the memristor crossbar, it cannot distinguish the two sequences. Similarly, for the second SDR

of “B”, the crossbar circuit also cannot make a judgement between “A-B-C-D-E” and “A-B-C-E-D”. However, if the fourth SDR is given to the crossbar, it can predict if the fifth SDR will be “E” or “D” according to the fourth SDR information. This is called the ordinal prediction in Figure 11, where the temporal-pooling circuit can predict the ordinal sequence of SDRs. Figure 11 shows the prediction rate of ordinal sentences. The prediction rate starts from zero. This means the crossbar can predict nothing at the starting time of prediction. If the first SDR is given, the crossbar starts to predict the rest words of the tested sentence. As the crossbar receives more words from the spatial pooler, the prediction becomes more accurate, as shown in Figure 11. When the crossbar receives the final SDR at the end of sentence (period symbol), the prediction rate in Figure 11 becomes equal to the recognition rate of sentences in Figure 8.

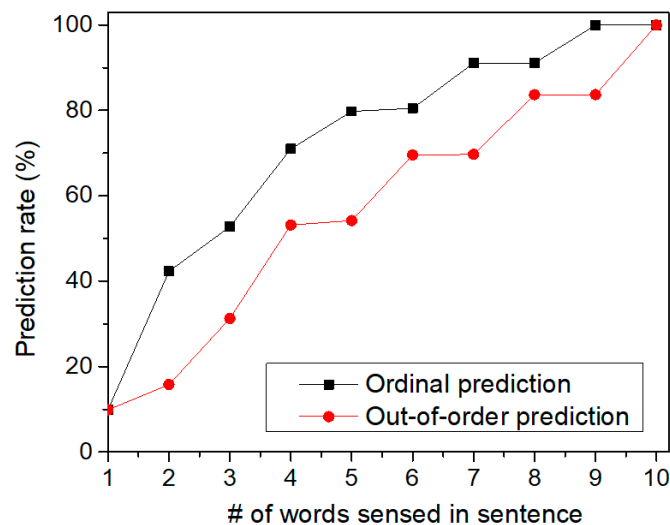


Figure 11. The prediction rate of sentences by increasing the number of words sensed for recognizing the sentences: Here, both the ordinal and out-of-order sequences can be recognized by the temporal-pooling memristor crossbar circuit proposed in this paper.

We also tested the prediction for out-of-order sequences in Figure 11. In the out-of-order prediction, the sequence of SDRs are out of order. In spite of the out-of-order sequence, the crossbar can accumulate the information of the sensed words over time. By doing so, the temporal-pooling circuit can guess what word should come next. This out-of-order prediction is exactly the same case as the crossword puzzle problem. In solving the crossword puzzle, we predict the word by accumulating the information of letters in the out-of-order sequence over time. When the temporal-pooling circuit is given only half words in the tested sentence, Figure 11 indicates that the crossbar can predict the ordinal and out-of-order sentences as accurate as 79.8% and 54.2%, respectively.

4. Discussion

In this section, we compare the proposed memristor-CMOS hybrid circuit with the previous sequential memristor crossbar [40] in terms of the memristor crossbar area, power consumption, and prediction of the ordinal and out-of-order sequences. The previous sequential memristor crossbar was designed not to consider the concept of location SDR in the crossbar, unlike the proposed temporal-pooling hybrid circuit in this paper [40]. Thus, the previous sequential scheme can recognize only the ordinal not the out-of-order sequence [40]. This is a very big disadvantage of the previous sequential scheme. For the power consumption, we had to program memristor cells of the serial chain one by one in the previous sequential crossbar to recognize the ordinal sequences [40]. This results in a large amount of programming power consumption in the previous scheme. On the contrary, the proposed temporal-pooling hybrid circuit does not demand the memristor programming in recognizing both the ordinal and out-of-order sequences. By doing so, the power consumption of the

proposed temporal-pooling circuit can be almost as small as 1/29 of the previous scheme, as indicated in Table 1. One more thing to note is the CMOS peripheral circuit in Figure 4a consumes only a negligible amount of the power than the memristor crossbar. Actually, most of the power is consumed in the LRS cells in the crossbar. Thus, minimizing the number of LRS cells in the memristor crossbar is very critical not only for alleviating the sneak leakage problem but also for reducing the power consumption [21]. Comparing the memristor crossbar's area between the previous and proposed schemes indicates the number of memristors of the proposed temporal-pooling hybrid circuit is estimated almost the same with that of the previous scheme, as shown in Table 1.

Table 1. A comparison of the memristor crossbar area, power consumption, and prediction of the ordinal and out-of-order sequences.

Scheme	The Previous Sequential Memristor Crossbar [40]	The Proposed Memristor-CMOS Hybrid Circuit of Temporal Pooling
The number of memristors (Memristor crossbar area)	17556	17027
The amount of power consumption (LRS = 1 M Ω , HRS = 100 M Ω)	151.5 μ W	5.24 μ W
Prediction of ordinal sequences	O	O
Prediction of out-of-order sequences	X	O

Finally, we discuss here the practical applications of Hebbian-based HTM algorithm. Actually, if we compare the Hebbian-based HTM algorithm with the previous deep-learning ones such as Convolutional Neural Networks, etc. for recognizing the benchmark image data-set, the deep learning outperforms the Hebbian-based HTM [21]. However, according to Numenta Inc. that developed HTM algorithm, the biologically inspired HTM can work best with data that meets the following characteristics: streaming data rather than batch data files, data with time-based patterns, many individual data sources where hand crafting separate models is impractical, subtle patterns that cannot always be seen by humans, and data for which simple techniques such as thresholds yield substantial false positives and false negatives [41]. This means that the Hebbian-based HTM algorithm can be more suitable to the area of Human-like sensory information such as the streaming data composed of anomaly patterns, as we showed in the case of the out-of-order prediction in Figure 11. On the contrary, for a static image data-set such as MNIST, CIFAR, etc., the conventional deep learning techniques can be better than HTM [21]. The real practical applications of the Hebbian-based HTM algorithm were explained in detail in previous publications [41,42]. In addition, the experimental results of memristor crossbars with Hebbian learning were shown in previous publications [43,44], where memristor's conductance was trained by the Hebbian algorithm for various neuromorphic applications.

5. Conclusions

As a software framework, Hierarchical Temporal Memory (HTM) has been developed to perform the brain's neocortical functions such as spatial and temporal pooling in software. However, it should be realized with hardware not software not only to mimic the neocortex's function but also to exploit its architectural benefit. To do so, in this paper, we proposed the memristor-CMOS hybrid circuit to realize the temporal-pooling function of human brain, which is composed of the input and output layers to mimic the neocortical neurons. In the hybrid circuit, the input layer has proximal and basal/distal dendrites to combine the sensory information with the temporal/location information caused from the brain's hippocampus. Using the same crossbar architecture, the output layer can perform predictions by integrating the temporal information through the basal/distal dendrites. For training the memristor-CMOS hybrid circuit, we used only simple Hebbian learning, not the complicated backpropagation algorithm. Due to the simple hardware of Hebbian learning, the hybrid circuit can be thought very suitable to online learning.

The proposed memristor HTM circuit was verified by the circuit simulation using memristor's Verilog-A model obtained from the measurement. The proposed crossbar circuit was tested to recognize words and sentences that are composed of EMNIST data-set of handwritten letters. The recognition rate for sentences was estimated as high as 96.5% for 256-bit Sparse Distributed Representation (SDR). In addition, the proposed circuit was tested with the external noise and memristance variation. The proposed temporal-pooling circuit also was verified to perform both the ordinal and out-of-order predictions. When the proposed circuit was given only half words in the tested sentence, it could predict the ordinal and out-of-order sequences with the accuracy of 79.8% and 54.2%, respectively.

Author Contributions: All authors contributed to the submitted manuscript of the present work. K.S.M. defined the research topic. T.V.N. and K.V.P. performed the simulation and measurement. K.S.M. wrote the manuscript. All authors read and approved the submitted manuscript.

Funding: The work was financially supported by NRF-2015R1A5A7037615, MOTIE/KEIT (10052653), and ETRI grant (18ZB1800).

Acknowledgments: The CAD tools were supported by the IC Design Education Center (IDEC), Daejeon, Korea. The authors are very grateful for the device fabrication by Yeon Soo Kim, Chansoo Yoon, Sanik Lee, and Bae Ho Park, Konkuk University, Seoul, Korea.

Conflicts of Interest: The authors declare that they have no competing interests.

References

- Hawkins, J.; Blakeslee, S. *On Intelligence: How a New Understanding of the Brain Will Lead to the Creation of Truly Intelligent Machines*; Henry Holt & Company: New York, NY, USA, 2004.
- Horton, J.C.; Adams, D.L. The cortical column: A structure without a function. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **2005**, *360*, 837–862. [[CrossRef](#)] [[PubMed](#)]
- Thomson, A.M. Neocortical layer 6, a review. *Front. Neuroana* **2010**, *4*, 13. [[CrossRef](#)] [[PubMed](#)]
- Hensch, T.K.; Stryker, M.P. Columnar architecture sculpted by GABA circuits in developing cat visual cortex. *Science* **2004**, *303*, 1678–1681. [[CrossRef](#)] [[PubMed](#)]
- Muir, D.R.; Cook, M. Anatomical constraints on lateral competition in columnar cortical architectures. *Neural Comput.* **2014**, *26*, 1624–1666. [[CrossRef](#)] [[PubMed](#)]
- Douglas, R.J.; Martin, K.A.C.; Whitteridge, D. A canonical microcircuit for neocortex. *Neural Comput.* **1989**, *1*, 480–488. [[CrossRef](#)]
- Chua, L.O. Memristor—the missing circuit element. *IEEE Trans. Circuit Theory* **1971**, *18*, 507–519. [[CrossRef](#)]
- Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [[CrossRef](#)] [[PubMed](#)]
- Jo, S.H.; Chang, T.; Ebong, I.; Bhadviya, B.B.; Mazumder, P.; Lu, W. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* **2010**, *10*, 1297–1301. [[CrossRef](#)] [[PubMed](#)]
- Du, C.; Cai, F.; Zidan, M.A.; Ma, W.; Lee, S.H.; Lu, W.D. Reservoir computing using dynamic memristors for temporal information processing. *Nat. Commun.* **2017**, *8*, 2204. [[CrossRef](#)] [[PubMed](#)]
- Kügeler, C.; Meier, M.; Rosezin, R.; Gilles, S.; Waser, R. High-density 3D memory architecture based on the resistive switching effect. *Solid State Electron.* **2009**, *53*, 1287–1292. [[CrossRef](#)]
- Shulaker, M.M.; Wu, T.F.; Pal, A.; Zhao, L.; Nishi, Y.; Saraswat, K.; Wong, H.-S.P.; Mitra, S. Monolithic 3D integration of logic and memory: Carbon nanotube FETs, resistive RAM, and silicon FETs. In Proceedings of the IEEE International Electron Devices Meeting, San Francisco, CA, USA, 15–17 December 2014; pp. 638–641.
- Truong, S.N.; Shin, S.H.; Byeon, S.D.; Song, J.S.; Min, K.S. New twin crossbar architecture of binary memristors for low-power image recognition with discrete cosine transform. *IEEE Trans. Nanotechnol.* **2015**, *14*, 1104–1111. [[CrossRef](#)]
- Truong, S.N.; Ham, S.J.; Min, K.S. Neuromorphic crossbar circuit with nanoscale filamentary-switching binary memristors for speech recognition. *Nanoscale Res. Lett.* **2014**, *9*, 1–9. [[CrossRef](#)] [[PubMed](#)]
- Hawkins, J.; Ahmad, S.; Dubinsky, D. *Hierarchical Temporal Memory Including HTM Cortical Learning Algorithms*; Tech. Rep.; Numenta, Inc.: Palo Alto, CA, USA, 2011.
- Cui, Y.; Ahmad, C.; Hawkins, J. The HTM spatial pooler—A neocortical algorithm for online sparse distributed coding. *bioRxiv* **2016**. [[CrossRef](#)] [[PubMed](#)]

17. Ahmad, S.; Hawkins, J. Properties of sparse distributed representations and their application to hierarchical temporal memory. *arXiv*, 2015; arXiv:1503.07469.
18. Ahmad, S.; Hawkins, J. How do neurons operate on sparse distributed representations? A mathematical theory of sparsity, neurons and active dendrites. *arXiv*, 2016; arXiv:1601.00720.
19. Cui, Y.; Ahmad, C.; Hawkins, J. Continuous online sequence learning with an unsupervised neural network model. *arXiv*, 2015; arXiv:1512.05463.
20. Pietroń, M.; Wielgosz, M.; Wiatr, K. Formal analysis of HTM Spatial Pooler performance under predefined operation conditions. *arXiv*, 2016; arXiv:1607.00791v1.
21. Truong, S.N.; Pham, K.V.; Min, K.S. Spatial-pooling memristor crossbar converting sensory information to sparse distributed representation of cortical neurons. *IEEE Trans. Nanotechnol.* **2018**, *17*, 482–491. [[CrossRef](#)]
22. Hawkins, J.; Ahmad, S. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *arXiv*, 2016; arXiv:1511.00083.
23. Hawkins, J.; Ahmad, S.; Cui, Y. A theory of how columns in the neocortex enable learning the structure of the world. *Front. Neural Circuits* **2017**, *11*, 81. [[CrossRef](#)] [[PubMed](#)]
24. Zeki, S.; Shipp, S. The functional logic of cortical connections. *Nature* **1988**, *335*, 311–317. [[CrossRef](#)] [[PubMed](#)]
25. Ungerleider, L.G.; Mishkin, M. Two cortical visual system. In *Analysis of visual Behavior*; Goodale, M.A., Ingle, D.J., Mansfield, R.J., Eds.; MIT Press: Cambridge, MA, USA, 1982; pp. 549–586.
26. Friston, K.; Buzáki, G. The functional anatomy of time: What and when in the brain. *Trends Cogn. Sci.* **2016**, *20*, 500–511. [[CrossRef](#)] [[PubMed](#)]
27. Fortin, N.J.; Agster, K.L.; Eichenbaum, H. Critical role of the hippocampus in memory for sequences of events. *Nat. Neurosci.* **2002**, *5*, 458–462. [[CrossRef](#)] [[PubMed](#)]
28. Ergorul, C.; Eichenbaum, H. The hippocampus and memory for “what,” “where,” and “when”. *Learn. Mem.* **2004**, *11*, 397–405. [[CrossRef](#)] [[PubMed](#)]
29. Moser, E.I.; Kropff, E.; Moser, M.B. Place cell, grid cells, and the brain’s spatial representation system. *Annu. Rev. Neurosci.* **2008**, *31*, 69–89. [[CrossRef](#)] [[PubMed](#)]
30. Krestinskaya, O.; Dolzhikova, I.; James, A.P. Hierarchical temporal memory using memristor networks: A survey. *IEEE Trans. Circuits Syst.* **2018**, *2*, 380–395. [[CrossRef](#)]
31. Wijesinghe, P.; Ankit, A.; Sengupta, A.; Roy, K. An all-memristor deep spiking neural computing system: A step toward realizing the low-power stochastic brain. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 345–358. [[CrossRef](#)]
32. Krestinskaya, O.; James, A.P.; Chua, L.O. Neuro-memristive circuits for edge computing: A review. *arXiv* **2018**, arXiv:1807.00962.
33. Shipp, S. Structure and function of the cerebral cortex. *Curr. Biol.* **2007**, *17*, R443–R449. [[CrossRef](#)] [[PubMed](#)]
34. Douglas, R.J.; Martin, K.A. Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.* **2004**, *27*, 419–451. [[CrossRef](#)] [[PubMed](#)]
35. Truong, S.N.; Pham, K.V.; Yang, W.S.; Shin, S.H.; Pedrotti, K.; Min, K.S. New pulse amplitude modulation for fine tuning of memristor synapses. *Microelectron. J.* **2016**, *55*, 162–168. [[CrossRef](#)]
36. *Virtuoso Spectre Circuit Simulator User Guide*; Cadence Design System Inc.: San Jose, CA, USA, 2011.
37. Cohen, G.; Afshar, S.; Tapson, J.; Schaik, A.V. EMNIST: An extension of MNIST to handwritten letters. *arXiv* **2017**, arXiv:1702.05373.
38. James, A.P.; Fedorova, I.; Ibrayer, T.; Kudithipudi, D. HTM spatial pooler with memristor crossbar circuits for sparse biometric recognition. *IEEE Trans. Biomed. Circuits Syst.* **2017**, *11*, 640–651. [[CrossRef](#)] [[PubMed](#)]
39. Shin, S.H.; Byeon, S.D.; Song, J.S.; Truong, S.N.; Mo, H.S.; Kim, D.J.; Min, K.S. Dynamic reference scheme with improved read voltage margin for compensating cell-position and back ground-pattern dependencies in pure memristor array. *J. Semicond. Technol. Sci.* **2015**, *15*, 685–694. [[CrossRef](#)]
40. Truong, S.N.; Pham, K.V.; Yang, W.; Min, K.S. Sequential memristor crossbar for neuromorphic pattern recognition. *IEEE Trans. Nanotechnol.* **2016**, *15*, 922–930. [[CrossRef](#)]
41. The Numenta Anomaly Benchmark: The First Temporal Benchmark Designed for Anomaly Detection in Streaming Data, Whitepaper. Numenta, Inc., 2015. Available online: <https://numenta.com/assets/pdf/numenta-anomaly-benchmark/NAB-Business-Paper.pdf> (accessed on 25 February 2019).
42. Lavin, V.; Ahmad, S. Evaluating real-time anomaly detection algorithms—the Numenta anomaly benchmark. In Proceedings of the IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 9–11 December 2015; pp. 38–44.

43. Ziegler, M.; Riggert, C.; Hansen, M.; Bartsch, T.; Kohlstedt, H. Memristive Hebbian Plasticity Model: Device Requirements for the Emulation of Hebbian Plasticity Based on Memristive Devices. *IEEE Trans. Biomed. Circuits Syst.* **2015**, *9*, 197–206. [[CrossRef](#)] [[PubMed](#)]
44. Hansen, M.; Zahari, F.; Kohlstedt, H.; Ziegler, M. Unsupervised Hebbian learning experimentally realized with analogue memristive crossbar arrays. *Sci. Rep.* **2018**, *8*, 8914. [[CrossRef](#)] [[PubMed](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).