Convolutional Neural Network Based on Crossbar Arrays of (Co-Fe-B)x(LiNbO3)100−x Nanocomposite Memristors

Anna N. Matsukatova 1,2,*, Aleksandr I. Iliasov 1,2,*, Kristina E. Nikiruy 1,†, Elena V. Kukueva 1, Aleksandr L. Vasiliev 1, Boris V. Goncharov 1, Aleksandr V. Sitnikov 1,3, Maxim L. Zanaveskin 1, Aleksandr S. Bugaev 4, Vyacheslav A. Demin 1, Vladimir V. Rylkov 1,5 and Andrey V. Emelyanov 1,4,*

1 National Research Center “Kurchatov Institute”, 123182 Moscow, Russia
2 Faculty of Physics, Lomonosov Moscow State University, 119991 Moscow, Russia
3 Department of Solid State Physics, Faculty of Radio Engineering and Electronics, Voronezh State Technical University, 394026 Voronezh, Russia
4 Moscow Institute of Physics and Technology, State University, 141700 Dolgoprudny, Russia
5 Kotelnikov Institute of Radio Engineering and Electronics RAS, 141190 Fryazino, Russia
* Correspondence: emelyanov_av@nrcki.ru
† Current address: Technische Universität Ilmenau, Ehrenbergstrasse 29, 98693 Ilmenau, Germany.

Abstract: Convolutional neural networks (CNNs) have been widely used in image recognition and processing tasks. Memristor-based CNNs accumulate the advantages of emerging memristive devices, such as nanometer critical dimensions, low power consumption, and functional similarity to biological synapses. Most studies on memristor-based CNNs use either software models of memristors for simulation analysis or full hardware CNN realization. Here, we propose a hybrid CNN, consisting of a hardware fixed pre-trained and explainable feature extractor and a trainable software classifier. The hardware part was realized on passive crossbar arrays of memristors based on nanocomposite (Co-Fe-B)x(LiNbO3)100−x structures. The constructed 2-kernel CNN was able to classify the binarized Fashion-MNIST dataset with ~ 84% accuracy. The performance of the hybrid CNN is comparable to the other reported memristor-based systems, while the number of trainable parameters for the hybrid CNN is substantially lower. Moreover, the hybrid CNN is robust to the variations in the memristive characteristics: dispersion of 20% leads to only a 3% accuracy decrease. The obtained results pave the way for the efficient and reliable realization of neural networks based on partially unreliable analog elements.

Keywords: memristor; resistive switching; nanocomposite; neuromorphic computing; convolutional neural network

1. Introduction

Memristor-based neuromorphic computing systems (NCSs) provide a fast, high-computational, and energy-efficient approach to neural network (NN) training and solving cognitive problems (pattern and speech recognition, big data processing, prediction, and so on) [1,2]. Memristors could be organized in large crossbar arrays (with critical dimensions down to 6 nm [3]) to perform vector–matrix multiplication in a natural one-step method by weighted electrical current summation (according to the Ohm’s and Kirchhoff’s laws) [4]. In contrast, being the most massively parallel operation in NN learning and inference, vector–matrix multiplication is extremely time- and energy-expensive in traditional von Neumann architectures [2]. Owing to this difference, memristor-based NCSs are of high interest. Memristors have already been successfully implemented for diverse NCS realizations, and such schemes as perceptrons [5,6], spiking [7,8], or long short-term memory [9] networks and others (including NN circuits of Pavlov’s associative memory) [10–12] have been demonstrated. Most of these NCSs are usually trained by various types of gradient
descent learning algorithm, the hardware realization of which is challenging because of unreliable cycle-to-cycle (c2c) and device-to-device (d2d) variations of memristive devices [2]. Several approaches have been proposed to partially mitigate these problems, including reservoir computing [13] and convolutional [14,15] NNs. The latter one is of particular interest as it allows to reduce the number of required weights (i.e., memristors) compared with fully connected NNs and, at the same time, demonstrates excellent performance in object recognition and image processing [15,16]. Convolutional NNs (CNNs) consist of two main parts: the feature extractor (convolutional layers) and classifier (fully connected layers). Convolutional layers extract feature maps from the input images by applying filters (kernels) of different dimensions, which allows decreasing the number of inputs. Most studies on memristor-based CNNs use either software models of memristors to emulate both parts of CNNs [16–21] or fully hardware parts of CNNs [14,22,23]. However, it significantly complicates the evaluation of the memristor-based convolutional layer efficiency, which should not be neglected. Generally, CNNs are prone to learning trustworthy features and overfitting. An illustrative example of this case includes a CNN trained to classify images of huskies and wolves, which instead learned background features of the images, such as the presence of snow [24]. This highlights the importance of so-called explainability of the NNs. One possible way to control the feature extractor weights is to train the convolutional layers ex situ via a traditional computing system, and then transfer them to the memristors (hybrid training) [22]. Multiple software algorithms may help visualize and study the extracted features in this case [25]. However, this approach implies that a high-cost training process alongside with its verification should be performed in the software before the transfer of the convolutional kernel weights to the hardware. Moreover, the convolutional kernels may need to be retrained if the classification problem is changed.

In this work, we propose a general approach to the implementation of a memristor-based CNN—a hybrid network, consisting of a hardware fixed pre-trained and explainable feature extractor and a trainable software classifier. Unlike the hybrid training process, where the weight update of a hardware memristor-based NN is made according to the software ex situ training results at each training cycle [22], in our case, no additional training of the CNN memristive part is needed. The hybrid CNN possesses the advantages of both hardware systems in terms of energy and computational efficiency and software systems in terms of architectural flexibility. The usage of convolutional layers with universally recognized fixed kernels (horizontal and vertical) gives hope that such a hybrid network would be eligible for any given image classification problem, and only the weights of its classifier part would require some fine-tuning. Here, we test this approach on the Fashion-MNIST (F-MNIST) image recognition problem [26]. The main goal of this work is to estimate the efficiency of the fixed memristor-based convolutional layers compared with the ideal software trainable ones.

2. Materials and Methods

2.1. Device Fabrication

Arrays of crossbar M/NC/LNO/M memristors were created in a technological route based on laser photolithography on Heidelberg 66fs lithograph (patterning electrode buses), ion-beam sputtering on the original system: in the beginning of target LiNbO$_3$ ($\approx$10 nm), and then composite target (Co$_{40}$Fe$_{40}$B$_{20}$)$_x$(LiNbO$_3$)$_{100-x}$ with $x \approx 10–25$ at.% ($\approx$230 nm). Plasma chemical deposition via Trion Oracle III was used for deposition of the isolating Si$_3$N$_4$ layer to isolate the edge regions of memristors in the array from possible electrical breakdown effects (see details in [27]). Multiple crossbar arrays of different sizes were created on one silicon wafer in one technological cycle (Figure 1a), and one of them (16 × 16) was used in this work.

2.2. Electrical Measurements

Electrophysical characterization of memristors as well as their switching to different required resistance states during the preparation of convolutional kernels were performed
using the source measurement unit (National Instruments PXIe-4140). To provide electrical signals, representing pixels of image, a digital-to-analog converter (National Instruments PXIe-6738) was used. Output currents (the results of convolutional operations) were measured using a two-channel digital oscilloscope (National Instruments PXIe-5110). For I–V measurements, the compliance current was set to 100 mA for both voltage polarities and the rate of voltage scan was set to 2 V/s.

Figure 1. (a) Photo of a single silicon wafer with multiple 4 × 4 and 16 × 16 memristor crossbar arrays; (b) dark field TEM image of a single memristor of an array with denoted layers; (c) I–V characteristics of all utilized NC memristors with optimal $x \approx 23$ at.\% ($x$ is the metal concentration in the LNO NC); (d) retention time of a single memristor from the crossbar array (only a few selected data points are presented for better visibility, dashed lines represent the average values of the resistances).

2.3. TEM

The structural investigations were carried out in a scanning/transmission electron microscope (S/TEM) Osiris (Thermo Fisher Scientific, Carlsbad, CA, USA), equipped with a high angle annular dark field (HAADF) detector (Fischione, Pittsburgh, PA, USA) and an energy-dispersive X-ray (EDX) Bruker Super-X spectrometer (Bruker, Billerica, MA, USA) for the chemical composition analysis.

2.4. Hardware Convolutional Layer Implementation

The kernels were implemented on a crossbar array as follows. As long as memristors cannot provide negative weights directly, the kernels were split into two parts—with positive and negative weights only. They defined target resistances of memristors in two columns: devices, corresponding to non-zero weight (1 for the first column and −1 for the second one) were set to $R_{\text{on}}$ state, while others (−1 and 0 for the first column and 0 and 1 for the second one) were set to $R_{\text{off}}$ state. Each memristor was set to the target state individually, performing the corresponding half of the I–V loop (positive polarity for the $R_{\text{on}}$ state, and vice versa). Vertical electrodes were grounded via two equal resistors;
this way, the current-to-voltage conversion was performed to measure output currents using an oscilloscope. When each patch of the image is fed to the crossbar array, these output currents from two columns represent the result of convolutional operation owing to intrinsic properties of the crossbar array geometry—input voltages, i.e., the pixels of the patch, are weighted with memristor conductance, i.e., the weights of the convolutional kernel. As stated in the main text, the resulting output current equaled to the difference between the currents from two array columns. In order to speed up the work, the resulting currents for every possible $3 \times 3$ binarized image patch (total of $2^{3\times3} = 512$ possible patches) were measured. After all 512 currents had been measured, they were normalized to a $[0,1]$ interval and used in the software classifier. These operations were performed for horizontal and vertical filters successively.

2.5. Neural Network Simulation

Initial data preprocessing included images’ binarization, i.e., each pixel of the image with a higher value than some threshold value equaled 0, while all others equaled 1. The threshold value equaled 10 and was estimated by eye, in order not to make it another hyperparameter of the CNN. The F-MNIST dataset is pre-divided into training (60,000 images) and testing (10,000 images) datasets. The original training dataset was then divided into training (50,000) and validation (10,000) datasets. The CNN hyperparameters were adjusted with the validation dataset, while the test dataset was used only for the final evaluation of the CNN performance. The hyperparameters list and detailed discussion of their fitting are presented in Supplementary Note S1 (Figures S1–S3, Table S1). The CNN was simulated in the PyTorch framework.

3. Results and Discussion

In this work, the LiNbO$_3$ (LNO)-based memristors were used, as they are of emerging interest [28–30], especially those with embedded metal nanogranules [31]. In our studies, the capacitor metal/nanocomposite/metal (M/NC/M) structures based on (Co-Fe-B)$_x$(LiNbO$_3$)$_{100-x}$ NC were fabricated by ion-beam sputtering with a metal content $x \approx 8$–25 at.% [31]. The NC films, along with the metal nanogranules of 3–6 nm, contained a large number of dispersed Co (Fe) atoms (up to $\sim 10^{21}$–$10^{22}$ cm$^{-3}$). At some optimal $x \approx 8$–15 at.%, the M/NC/M structures manifest stable resistive switching (RS) through a multifilamentary mechanism [32]; demonstrate high endurance, long retention, and multilevel RS [33,34]; and can be successfully used in NCSs [27,35].

In this work, we fabricated NC structures with a thin built-in LNO layer near the bottom electrode (i.e., structures like M/NC/LNO/M), which plays a critical role in the realization of stable RS [32]. The arrays of the NC memristors were fabricated in a $16 \times 16$ crossbar architecture (Figure 1a). Then, their characteristics were studied in order to verify the eligibility of the NC crossbar memristors for the hardware realization of different $3 \times 3$ convolutional kernels.

The results of transition electron microscopy (TEM) provided a descriptive picture of the layer thickness (Figure 1b) and composition (elemental maps are presented in Figure S4) for a single memristive element from a $16 \times 16$ crossbar array. High-resolution images (Figure S5) confirmed the presence of a $\sim 10$ nm thick pure LNO layer near the bottom electrode. Figure 1c presents the current–voltage ($I$–$V$) characteristics of all 18 memristors (9 rows and 2 columns) used in this work, five cycles for each one. As can be seen in Figure 1c, while $c_{2c}$ variations for these memristors are negligible, the $d_{2d}$ variations are more pronounced. Two groups of devices could be selected from the $I$–$V$ curves (shaded in different colors in Figure 1c). Each group represents devices from different columns of the crossbar array. The variations in the resistive switching voltage of these groups are associated with different resistances of crossbar busses, which act as load resistances, and some additional voltage drops on them. The bus resistances should be reduced to decrease the $d_{2d}$ variations in a crossbar array. However, as shown below, all of these devices can operate as equivalent parts of a convolutional kernel. Another
important memristive characteristic for the convolutional layer implementation is retention time—after the weights of all memristors are adjusted to represent the chosen kernel, they should not vary. As can be seen from Figure 1d, the resistance drift from both initial states (high resistance state, $R_{\text{off}}$, and low resistance state, $R_{\text{on}}$) is negligible compared with their difference $R_{\text{off}}-R_{\text{on}}$. It should be noted that the resistance values of the obtained memristors are not high enough ($\leq 1 \, \text{k}\Omega$), probably because of the small thickness of the LNO layer in the M/NC/LNO/M structures under study.

Figure 2a illustrates the proposed NN architecture. The original F-MNIST images were binarized in order to simplify the implementation of the hardware part. The features were then extracted with either a horizontal/vertical filter or both at once (the example of the extracted features is presented in Figure 2a). Then, the obtained feature matrices were flattened, normalized, and fed to the fully connected classifying layers (676 input neurons in the case of the CNN with one filter and 1352 in the case of the CNN with two filters). Figure 2b demonstrates the hardware feature extractor implementation. The image was divided into $3 \times 3$ patches; each patch was then flattened and fed to the crossbar array (i.e., the corresponding voltages were applied, 1 V for “1” pixels of the patch and 0 V for “0” pixels). The crossbar array acted either as a horizontal or a vertical kernel, i.e., the weights of nine memristive devices were adjusted to represent the chosen normalized and flattened $3 \times 3$ kernel. In order to obtain negative weights of the kernels, two columns of the crossbar array were used (i.e., nine rows-inputs and two columns-outputs were used in this work, as specified in the figure), and the resulting output current equaled the difference between the currents from both columns. A more detailed discussion of the feature extractor implementation can be found in Section 2.

The proposed NN architecture (Figure 2a) was additionally simulated in the software for subsequent comparison with the hybrid NN results. Figure 3a presents the accuracy of the one-filter NN during training, estimated on a validation dataset (the discussion of the initial dataset portioning can be found in Section 2). Figure 3b demonstrates the results.
for the two-filter NN. Three main conclusions can be drawn from these figures. Firstly, the binarization of the images does not lead to an accuracy decrease, so such a simplification can be done. Secondly, the CNNs with fixed filters do not concede to the trainable ones dramatically. Finally, during the first training epochs, the trainable filter usage leads to high accuracy variations; thus, such CNNs concede to the CNNs with fixed filters at the beginning of the training process. These results authorized the creation of a hybrid CNN.

Figure 2. Schematic illustration of (a) the proposed neural network architecture and (b) the hardware convolutional layer implementation on the memristive crossbar array. The proposed NN architecture (Figure 2a) was additionally simulated in the software for subsequent comparison with the hybrid NN results.

Figure 3. Software modeling of the convolutional neural network with (a) one and (b) two kernels (filters). The image parameters (standard or binarized image) as well as the kernel weight parameters (trainable or fixed weights) are specified in the graphs. Each curve was obtained 10 times; the mean values with their standard deviations are presented in the graphs.

Figure 4a,b compare the results obtained from the hybrid NNs and full software systems with one and two filters correspondingly. The best simulation results from Figure 3a,b were chosen to make the comparison under stringent conditions. It can be seen that the hybrid NNs with one filter considerably concede to the simulated NN with a trainable filter. However, the results after 100 epochs for the two-filter hybrid NN are comparable to the simulation results. A more complete set of features, obtained using both horizontal and vertical filters, was generated for each image in this case, leading to a higher accuracy score. This result raises hope that some expanded set of filters may be created, which would extract all of the most important features from the images without additional training of the filters, thus leading to the creation of a generic hybrid NN with enhanced classification accuracy.

Generally, the F-MNIST dataset was developed in order to replace the conventional MNIST digit dataset. Some modern software NNs can classify the MNIST digit dataset with an accuracy >99%, which makes it too simple for the NN performance evaluation [36]. In contrast, quite elaborate software NN architectures are required in order to surpass 90% accuracy for the F-MNIST dataset [26,37]. Meanwhile, the number of trainable parameters for such NNs equals 500–700 k (grayscale F-MNIST). The same goes for the memristor-based CNNs, e.g., sixteen 9 × 9 convolutional kernels were needed to reach ~87% accuracy (binarized F-MNIST) [16]. Another memristive CNN example demonstrated ~93% with ~3.5 M parameters (grayscale F-MNIST) [38]. In our study, the test classification accuracy of the hybrid CNN with two filters equaled ~84%, while the number of trainable parameters equaled ~44 k. This is a high enough accuracy value for a hybrid system with such a small number of parameters. Most mistakes were made for the classes, which are almost indistinguishable in the case of the binarized images (Figure 4c). The reduced number of trainable parameters leads to a less sophisticated and more robust training process, so introduction of the memristors does not dramatically decrease the classification accuracy.
parameters (trainable or fixed weights) are specified in the graphs. Each curve was obtained 10 times; the mean values with their standard deviations are presented in the graphs.

Figure 4a,b compare the results obtained from the hybrid NNs and full software systems with one and two filters correspondingly. The best simulation results from Figure 3a,b were chosen to make the comparison under stringent conditions. It can be seen that the hybrid NNs with one filter considerably concede to the simulated NN with a trainable filter. However, the results after 100 epochs for the two-filter hybrid NN are comparable to the simulation results. A more complete set of features, obtained using both horizontal and vertical filters, was generated for each image in this case, leading to a higher accuracy score. This result raises hope that some expanded set of filters may be created, which would extract all of the most important features from the images without additional training of the filters, thus leading to the creation of a generic hybrid NN with enhanced classification accuracy.

Figure 4. Comparison of the smoothed software modeling results to the hybrid NN ones with (a) one and (b) two kernels (filters). (c) Confusion matrix for the hybrid CNN with two filters, evaluated on the test dataset, with examples of the misclassified images for the classes with the most frequent mistakes. The maximum ideal values for the main diagonal items equal 100%.

Moreover, the influence of the variability introduction to the hardware CNN part on its performance was studied. The weights of the hardware filters (i.e., resistive states of the memristive crossbar) in the hybrid CNN are set once and are not tuned in the future. Thus, memristors should remain in their initial resistive states to produce a reliable output from the feature extractor. It can be seen in Figure 1d that our memristors have good retention, i.e., their states are not changed in time even if no external voltage is applied. However, it is known that memristive systems are in general prone to variations in characteristics [2]. It is probable that, after a longer time, some variations may appear as a result of internal microscopic degradation or external impacts. Therefore, we examine the influence of such possible variations on the classification accuracy. In order to simulate the variation introduction to the CNN, each output current from the memristive convolutional layer was chosen from a normal distribution, for which the mean value equaled the experimental results and the coefficient of variation was chosen from 0 to 100%. Only a high coefficient of variation (e.g., 100%) led to a considerable degradation of the training process and a decrease in the test dataset accuracy. All obtained data are summarized in Table S2 and demonstrate the robustness of the hybrid CNN to the variations in the memristor characteristics: dispersion of 20% leads to only a 3% decrease in accuracy.

4. Conclusions

In summary, we fabricated and studied the crossbar arrays of nanocomposite-based (Co-Fe-B)x(LiNbO3)100−x memristors. Memristors in a single crossbar array demonstrate negligible c2c variations, while d2d variations are more pronounced, which was attributed
to the considerable impact of the crossbar busses’ resistances to the total resistance of the structure. Using the nanocomposite-based crossbar arrays, we implemented a hybrid CNN, consisting of a hardware feature extractor with one/two kernels and a software classifier. The two-kernel CNN was able to classify the binarized Fashion-MNIST dataset with an accuracy of ~84%. The performance of the hybrid CNN is comparable to the full software and full hardware (memristive) systems, while the number of trainable parameters for the hybrid CNN is substantially lower. Moreover, the hybrid CNN is shown to be robust to the variations in memristive characteristics. The obtained results raise hope that enhanced performance may be achieved for any given image classification task in the future, if some expended set of fixed kernels is created for the hybrid CNN.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/nano12193455/s1, Figure S1: All tested via Optuna hyperparameter values with the resulting objective values (accuracy scores). The best values of each hyperparameter are denoted with bolder and brighter lines; Figure S2: The relative importance of each hyperparameter of the CNN, calculated via Optuna; Figure S3: Software modeling of the CNN with binarized F-MNIST and trainable filter for different CNN hyperparameters (Optuna and manually fine-tuned); Table S1: The Optuna-optimized hyperparameters and manually fine-tuned hyperparameters; Figure S4: Elemental EDX-maps of the M/NC/LNO/M memristor from a crossbar array; Figure S5: High resolution dark field TEM image of the area near the bottom electrode (at the edge of the crossbar busses intersection) of the M/NC/LNO/M memristor from a crossbar array; Table S2. The influence of the introduced variation on the training process of the 2-kernel hybrid CNN. Reference [39] is cited in the Supplementary Materials.

Author Contributions: Conceptualization and methodology, K.E.N., V.A.D., V.V.R. and A.V.E.; validation and formal analysis, E.V.K., A.L.V. and B.V.G.; investigation and data curation, A.I.I.; resources, A.V.S., M.L.Z. and A.S.B.; software, A.N.M.; writing—original draft preparation, review and editing, A.N.M., V.V.R. and A.V.E.; visualization, A.N.M. and A.I.I.; supervision, project administration and funding acquisition, V.A.D., V.V.R. and A.V.E. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Russian Science Foundation (project № 22-19-00171) in the part of crossbar array fabrication and their characterization and by the Ministry of Science and Higher Education of the Russian Federation (MK-2203.2021.1.2) in the part of hardware convolutional layer implementation. A.N. Matsukatova acknowledges financial support from Non-commercial Foundation for the Advancement of Science and Education INTELLECT in the neural network simulation part.

Data Availability Statement: The utilized dataset and Pytorch framework are publicly available. The CNN architecture and its hyperparameters are provided in the text, which simplifies the replication of the software modeling results. The other data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: Measurements were carried out with the equipment of the Resource Centres (NRC “Kurchatov Institute”). The authors are thankful to N.Yu. Lukashina (illustrator) for high-quality illustrations and to Yu.V. Grishchenko and K.Yu. Chernoglayzov for lithographic patterning of the memristor array in crossbar architecture.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Ielmini, D.; Wong, H.-S.P. In-memory computing with resistive switching devices. Nat. Electron. 2018, 1, 333–343. [CrossRef]
2. Zhang, Y.; Wang, Z.; Zhu, J.; Yang, Y.; Rao, M.; Song, W.; Zhuo, Y.; Zhang, X.; Cui, M.; Shen, L.; et al. Brain-inspired computing with memristors: Challenges in devices, circuits, and systems. Appl. Phys. Rev. 2020, 7, 011308. [CrossRef]
3. Pi, S.; Li, C.; Jiang, H.; Xia, W.; Xin, H.; Yang, J.J.; Xia, Q. Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension. Nat. Nanotechnol. 2019, 14, 35–39. [CrossRef] [PubMed]
4. Xia, Q.; Yang, J.J. Memristive crossbar arrays for brain-inspired computing. Nat. Mater. 2019, 18, 309–323. [CrossRef] [PubMed]
5. Prezioso, M.; Merrikh-Bayat, F.; Hoskins, B.D.; Adam, G.C.; Likharev, K.K.; Strukov, D.B. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. Nature 2015, 521, 61–64. [CrossRef] [PubMed]
6. Shchanikov, S.; Zuev, A.; Bordanov, I.; Danilin, S.; Lukoyanov, V.; Korolev, D.; Belov, A.; Pigareva, Y.; Gladkov, A.; Pimashkin, A.; et al. Designing a bidirectional, adaptive neural interface incorporating machine learning capabilities and memristor-enhanced hardware. Chaos Solitons Fractals 2021, 142, 110504. [CrossRef]

7. Demin, V.A.; Nekhaev, D.V.; Surazhevsky, I.A.; Nikiruy, K.E.; Emelyanov, A.V.; Nikolaev, S.N.; Rylikov, V.V.; Kovalchuk, M.V. Necessary conditions for STDP-based pattern recognition learning in a memristive spiking neural network. Neural Netw. 2021, 134, 64–75. [CrossRef] [PubMed]

8. Sboev, A.; Davydov, Y.; Rybka, R.; Vlasov, D.; Serenko, A.A. Comparison of Two Variants of Memristive Plasticity for Solving the Classification Problem of Handwritten Digits Recognition. In Biologically Inspired Cognitive Architectures 2021, Proceedings of the 12th Annual Meeting of the BICA Society (BICA 2021), Kyoto, Japan, 13-19 September 2021; Springer: Cham, Switzerland; Volume 1032, p. 438.

9. Li, C.; Wang, Z.; Rao, M.; Belkin, D.; Song, W.; Jiang, H.; Yan, P.; Li, Y.; Lin, P.; Hu, M.; et al. Long short-term memory networks in memristor crossbar arrays. Nat. Mach. Intell. 2019, 1, 49–57. [CrossRef]

10. Sun, J.; Han, G.; Zeng, Z.; Wang, Y. Memristor-based neural network circuit of full-function pavlov associative memory with time delay and variable learning rate. IEEE Trans. Cybern. 2019, 50, 2935–2945. [CrossRef]

11. Sun, J.; Han, J.; Wang, Y.; Liu, P. Memristor-based neural network circuit of emotion congruent memory with mental fatigue and emotion inhibition. IEEE Trans. Biomed. Circuits Syst. 2021, 15, 606–616. [CrossRef]

12. Sun, J.; Han, J.; Liu, P.; Wang, Y. Memristor-based neural network circuit of pavlov associative memory with dual mode switching. AEUInt. J. Electron. Commun. 2021, 129, 153552. [CrossRef]

13. Yang, J.; Cho, H.; Ryu, H.; Ismail, M.; Mahata, C.; Kim, S. Tunable synaptic characteristics of a Ti/TiO2/Si memory device for reservoir computing. ACS Appl. Mater. Interfaces 2021, 13, 33244. [CrossRef] [PubMed]

14. Wang, Z.; Li, C.; Lin, P.; Rao, M.; Nie, Y.; Song, W.; Qiu, Q.; Li, Y.; Yan, P.; Strachan, J.P.; et al. In situ training of feed-forward and recurrent convolutional memristor networks. Nat. Mach. Intell. 2019, 1, 434–442. [CrossRef]

15. Qin, Y.-F.; Bao, H.; Wang, F.; Chen, J.; Li, Y.; Miao, X.-S. Recent progress on memristive convolutional neural networks for edge intelligence. Adv. Intell. Syst. 2020, 2, 2000114. [CrossRef]

16. Wang, L.; Diao, J.; Nie, H.; Wang, W.; Li, Z.; Li, Q.; Liu, H. Memristor-based binary neural network architecture with configurable neurons. Front. Neurosci. 2021, 15, 639526. [CrossRef]

17. Gao, L.; Chen, P.Y.; Yu, S. Demonstration of convolution kernel operation on resistive cross-point array. IEEE Electron Device Lett. 2016, 37, 870–873. [CrossRef]

18. Zeng, X.; Wen, S.; Zeng, Z.; Huang, T. Design of memristor-based image convolution calculation in convolutional neural network. Neural Comput. Appl. 2018, 30, 503–508. [CrossRef]

19. Chen, J.; Chang, T.C.; Sze, S.M.; Miao, X.S.; Pan, W.Q.; Li, Y.; Kuang, R.; He, Y.H.; Lin, C.Y.; Duan, N.; et al. High-precision symmetric weight update of memristor by gate voltage ramping method for convolutional neural network accelerator. IEEE Electron Device Lett. 2020, 41, 353–356. [CrossRef]

20. Zhang, W.; Pan, L.; Yan, X.; Zhao, G.; Chen, H.; Wang, X.; Tay, B.K.; Zhong, G.; Li, J.; Huang, M. Hardware-Friendly Stochastic and Adaptive Learning in Memristor Convolutional Neural Networks. Adv. Intell. Syst. 2021, 3, 2100041. [CrossRef]

21. Chen, J.; Wu, Y.; Yang, Y.; Wen, S.; Shi, K.; Bermak, A.; Huang, T. An efficient memristor-based circuit implementation of squeeze-and-excitation fully convolutional neural networks. IEEE Trans. Neural Netw. Learn. Syst. 2022, 33, 1779–1790. [CrossRef]

22. Yao, P.; Wu, H.; Gao, B.; Tang, J.; Zhang, Q.; Zhang, W.; Yang, J.J.; Qian, H. Fully hardware-implemented memristor convolutional neural network. Nature 2020, 579, 641–646. [CrossRef] [PubMed]

23. Li, C.; Ignovskii, J.; Sheng, X.; Vessel, R.; Jaffe, B.; Ingemi, J.; Graves, C.; Strachan, J.P. CMOS-integrated nanoscale memristive crossbars for CNN and optimization acceleration. In Proceedings of the 2020 IEEE International Memory Workshop IMW 2020-Proceedings, Dresden, Germany, 17–20 May 2020; Volume 2, p. 10.

24. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 13–17 August 2016; pp. 1135–1144.

25. Štrumbelj, E.; Kononenko, I. Explaining prediction models and individual predictions with feature contributions. Knowl. Inf. Syst. 2014, 41, 647–665. [CrossRef]

26. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. 2017. Available online: http://arxiv.org/abs/1708.07747 (accessed on 19 September 2022).

27. Ilyasov, A.I.; Nikiruy, K.E.; Emelyanov, A.V.; Chernoglozov, K.Y.; Sitnikov, A.V.; Rylikov, V.V.; Demin, V.A. Arrays of Nanocomposite Crossbar Memristors for the Formalization of Spiking and Neuromorphic Systems. Nanobiotechnol. Rep. 2022, 17, 118–125. [CrossRef]

28. Wang, S.; Wang, W.; Yakopcic, C.; Shin, E.; Subramanyam, G.; Taha, T.M. Experimental study of LiNbO3 memristors for use in neuromorphic computing. Microelectron. Eng. 2017, 168, 37–40. [CrossRef]

29. Huang, S.; Luo, W.; Pan, X.; Zhao, J.; Qiao, S.; Shuai, Y.; Zhang, K.; Bai, X.; Niu, G.; Wu, C.; et al. Resistive Switching Effects of Crystal-Ion-Slicing Fabricated LiNbO3 Single Crystalline Thin Film on Flexible Polyimide Substrate. Adv. Electron. Mater. 2021, 7, 2100301. [CrossRef]

30. Wang, J.; Pan, X.; Wang, Q.; Luo, W.; Shuai, Y.; Xie, Q.; Zeng, H.; Niu, G.; Wu, C.; Zhang, W. Reliable Resistive Switching and Synaptic Plasticity in Ar+-irradiated Single-crystalline LiNbO3 Memristor. Adv. Electron. Mater. 2022, 596, 153653. [CrossRef]
31. Rylkov, V.V.; Emelyanov, A.V.; Nikolaev, S.N.; Nikiruy, K.E.; Sitnikov, A.V.; Fadeev, E.A.; Demin, V.A.; Granovsky, A.B. Transport Properties of Magnetic Nanogranular Composites with Dispersed Ions in an Insulating Matrix. *J. Exp. Theor. Phys.* 2020, 131, 160–176. [CrossRef]

32. Martyshov, M.N.; Emelyanov, A.V.; Demin, V.A.; Nikiruy, K.E.; Minnekhanov, A.A.; Nikolaev, S.N.; Taldenkov, A.N.; Ovcharov, A.V.; Presnyakov, M.Y.; Sitnikov, A.V.; et al. Multifilamentary Character of Anticorrelated Capacitive and Resistive Switching in Memristive Structures Based on (Co – Fe – B)x(LiNbO3)100–x Nanocomposite. *Phys. Rev. Appl.* 2020, 14, 034016. [CrossRef]

33. Nikiruy, K.E.; Emelyanov, A.V.; Demin, V.A.; Sitnikov, A.V.; Minnekhanov, A.A.; Rylkov, V.V.; Kashkarov, P.K.; Kovalchuk, M.V. Dopamine-like STDP modulation in nanocomposite memristors. *AIP Adv.* 2019, 9, 065116. [CrossRef]

34. Nikiruy, K.E.; Surazhevsky, I.A.; Demin, V.A.; Emelyanov, A.V. Spike-Timing-Dependent and Spike-Shape-Independent Plasticities with Dopamine-Like Modulation in Nanocomposite Memristive Synapses. *Phys. Status Solidi Appl. Mater. Sci.* 2020, 217, 1900938. [CrossRef]

35. Emelyanov, A.V.; Nikiruy, K.E.; Serenko, A.V.; Sitnikov, A.V.; Presnyakov, M.Y.; Rybka, R.B.; Sboev, A.G.; Rylkov, V.V.; Kashkarov, P.K.; Kovalchuk, M.V.; et al. Self-adaptive STDP-based learning of a spiking neuron with nanocomposite memristive weights. *Nanotechnology* 2020, 31, 045201. [CrossRef] [PubMed]

36. Gong, Y.; Zhang, P. Research on Mnist Handwritten Numbers Recognition based on CNN. *J. Phys. Conf. Ser.* 2021, 2138, 012002. [CrossRef]

37. Sun, Y.; Zhang, L.; Schaeffer, H. NeuPDE: Neural Network Based Ordinary and Partial Differential Equations for Modeling Time-Dependent Data. *Proc. Mach. Learn. Res.* 2020, 107, 352–372.

38. Sun, S.-Y.Y.; Xu, H.; Li, J.; Li, Q.; Liu, H. Cascaded architecture for memristor crossbar array based larger-scale neuromorphic computing. *IEEE Access* 2019, 7, 61679–61688. [CrossRef]

39. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; p. 2623.