Originally published as:

Bogatenko, T., Sergeev, K., Slepnev, A., Kurths, J., Nadezhda Semenova, N. (2023): Symbiosis of an artificial neural network and models of biological neurons: Training and testing. - Chaos, 33, 7, 07312.

DOI: https://doi.org/10.1063/5.0152703

Chaos
An Interdisciplinary Journal of Nonlinear Science

RESEARCH ARTICLE | JULY 11 2023

Symbiosis of an artificial neural network and models of biological neurons: Training and testing

Special Collection: Nonlinear dynamics, synchronization and networks: Dedicated to Jürgen Kurths' 70th birthday

Tatyana Bogatenko ; Konstantin Sergeev ; Andrei Slepnev ; Jürgen Kurths ; Nadezhda Semenova

Check for updates

Chaos 33, 073122 (2023)
https://doi.org/10.1063/5.0152703

View Online
Export Citation
CrossMark

16 November 2023 09:23:46

AIP Publishing

Symbiosis of an artificial neural network and models of biological neurons: Training and testing

Cite as: Chaos 33, 073122 (2023); doi: 10.1063/5.0152703
Submitted: 31 March 2023 · Accepted: 20 June 2023 ·
Published Online: 11 July 2023

View Online Export Citation CrossMark

Tatyana Bogatenko,[1] Konstantin Sergeev,[1] Andrei Slepnev,[1] Jürgen Kurths,[2,3] and Nadezhda Semenova[1,a)]

AFFILIATIONS

[1]Institute of Physics, Saratov State University, 83 Astrakhanskaya str., Saratov 410012, Russia
[2]Physics Department, Humboldt University, 15 Newtonstrasse, Berlin 12489, Germany
[3]Potsdam Institute for Climate Impact Research, A31 Telegrafenberg, Potsdam 14473, Germany

Note: This paper is part of the Focus Issue on Nonlinear dynamics, synchronization and networks: Dedicated to Juergen Kurths' 70th birthday.

[a)]**Author to whom correspondence should be addressed:** semenovani@sgu.ru

ABSTRACT

In this paper, we show the possibility of creating and identifying the features of an artificial neural network (ANN), which consists of mathematical models of biological neurons. The FitzHugh–Nagumo (FHN) system is used as a paradigmatic model demonstrating basic neuron activities. First, in order to reveal how biological neurons can be embedded within an ANN, we train the ANN with nonlinear neurons to solve a basic image recognition problem with an MNIST database; next, we describe how FHN systems can be introduced into this trained ANN. After all, we show that an ANN with FHN systems inside can be successfully trained with improved accuracy comparing with first trained ANN and then with inserted FHN systems. This approach opens up great opportunities in terms of the direction of analog neural networks, in which artificial neurons can be replaced by more appropriate biological ones.

Published under an exclusive license by AIP Publishing. https://doi.org/10.1063/5.0152703

16 November 2023 09:23:46

Over the recent years, artificial neural networks (ANNs) have found applications in solving many problems from pattern recognition to predicting climate phenomena. From a computational perspective, ANN modeling is a very resource-intensive task. Despite the existence of high-power computing clusters with the ability to parallel computations, neural network modeling on digital equipment is a bottleneck in network scaling and the speed of collecting and processing information. This makes more and more researchers in the field of neural networks engaged in creating hardware ANN implementations. In such a concept, neurons and the connections between them represent a real device that can learn and solve problems. There has been an exponential growth in the number of works concerning hardware ANNs based on lasers, memristors, spin-transfer oscillators, etc. The fundamental possibility of making such devices "compatible" with the nervous system of animals and humans is of particular interest. In modern developed countries, the observed increase in the life expectancy inevitably correlates with an increase in the number of cases of neurodegenerative diseases, as well as cognitive and motor problems that accompany the so-called healthy aging. Today, we already see experimental implementation of hardware systems that solve the problems of partially restoring vision, restoring the functions of lost limbs, and even restoring the sensitivity and capabilities of paralyzed limbs. In this paper, we propose a completely new concept, which suggests building the neurons of a network utilizing biological principles, but not physical ones. Biological neuron models, such as the FitzHugh–Nagumo model, are used as the components of the ANN. Here, we show how the topology of an already trained ANN can be used to implement FitzHugh–Nagumo models into it, as well as how the resulting neural network can be trained.

I. INTRODUCTION

There are two different approaches to implementing neural networks and two different definitions of neural networks. From a nonlinear dynamics' perspective, a neural network is of interest for describing biological phenomena and features of interaction between neurons within a neural circuit in response to an internal or

Published under an exclusive license by AIP Publishing

external impact. The temporal dynamics of biological neurons and the connections between them are extremely complex; therefore, there are a large number of studies describing conceptual models of different levels of complexity.[1–4]

On the other hand, there are artificial neural networks (ANNs). Despite having a similar name, these networks are totally different from biological neural networks in their purpose and design. ANNs are a prospective and widely spread tool for solving many computational tasks in a variety of scientific and engineering areas branching from climate research and medicine to sociology and economy.[5,6] An ANN consists of artificial neurons whose role is to generate an output signal based on a linear or nonlinear transformation of the input signal. Training an artificial neural network lies in fitting and altering the connection matrices between its neurons. In the learning process, the connection matrices are built in such a way that the network outputs the result required from it. The idea of implementing such neural networks came from biology.[7] Already in the 1940s, scientists were inspired by the idea of how a neural circuit is arranged and tried to implement its simplified model in order to solve non-trivial problems that do not obtain a strictly formulated solution algorithm.[8]

Having been first developed in the 1940s and 1950s, ANNs have undergone numerous substantial enhancements. Alternatively, simple threshold neurons of the first generation, which produced binary-valued outputs, evolved into systems, which use smooth activation functions, thus making it possible for the output to be real-valued. The most novel kind of an ANN is based on spiking neurons[9,10] and has received the name of a spiking neural network (SNN).

In contrast to neural networks of the previous generations, an SNN considers temporal characteristics of the information on the input. In this regard, an SNN architecture makes a step closer to a plausible model of a biological neural network, although still being highly simplified.[11] Within such a network, information transmission between artificial neurons resembles that of biological neurons. The neuron models in SNNs communicate by sequences of spikes. Typically, these spiking models are very simplified, and they have nothing in common with traditional models of biological neurons in nonlinear dynamics, such as a FitzHugh–Nagumo system, a Hodgkin–Huxley model, a leaky-integrate-and-fire neuron, the Izhikevich model, etc.

Similar to traditional ANNs, SNNs are arranged in layers, and a signal propagates from an input to the output layer traversing one or more hidden layers. However, in hidden layers, SNNs use spiking neurons, which are described by a phenomenological model representing a spike generation process. In a biological neuron, the activity of pre-synaptic neurons affects the membrane potential of post-synaptic neurons, which results in a generation of a spike when the membrane potential crosses a threshold.[9,11] This complex process has been described with the use of many mathematical models, with the Hodgkin–Huxley model being the first and the most famous one.[12] In order to find a balance between computational expenses and biological reality, several other models have been proposed, e.g., the leaky-integrate-and-fire model[13] or the Izhikevich model.[14]

Based on the available knowledge from neuroscience, several methods of information encoding have been developed, e.g., rate coding or latency coding. For rate coding, the rate (or the frequency) of spikes is used for information interpretation, while latency coding uses the timing of spikes. Both of these methods are special cases of a fully temporal code, where a timing correlates with a certain event, for instance, a spike of a reference neuron.

Having artificial neural networks and their tasks become more and more sophisticated, we may soon verge on some kind of a crisis;[15,16] i.e., the tasks become so complex that the capacities of modern computers will soon not be enough to meet the growing needs. Here, the bleeding-edge direction of hardware neural networks comes to the rescue.[17] According to this approach, neural networks are not created with a computer but are a real device that can learn and solve tasks. The neurons themselves and the connections between them exist at the physical level; i.e., the model is not simulated on a computer but is implemented in hardware according to its physical principles.

The main purpose of this work is to show the possibility of creating and identifying the features of a trained neural network, which consists of mathematical models of biological neurons. In this research, the FitzHugh–Nagumo system (FHN)[18,19] is used as an example. The FHN system is a well-known conceptual model of a biological neuron that demonstrates spike dynamics under certain conditions. It is often used to model basic neural activity.

This task helps us bring artificial neural networks closer to biological ones and reveal how biological neurons can be embedded within an artificial neural network; that is, we can combine the topology (the connection) from an ANN with the features of dynamics and interactions from a biological system. Thus, it enables us to approximate SNNs to biological ones. This will allow not only to obtain a working prototype of the ANN, but also to form a range of tasks in which the features of the temporal dynamics of the biological neurons themselves will be used. In the future, it is planned not only to develop an ANN with models of biological neurons, but also to study a range of neurophysiological problems in which the application of the principles obtained will allow modeling and structural processing of biomedical data. This direction is fundamentally new and relevant.

The work consists of several stages. First, there is a simple neural network with artificial linear and nonlinear neurons. It is trained to solve a basic image recognition problem with handwritten digits of the MNIST database (Sec. II). Then, a certain number of FHN systems is introduced into the existing neural network, and the task is to identify the conditions under which the network will still function (Sec. III). The next step is to make the task more difficult. From the beginning of this stage, we use a network in which the FHN systems are implemented and attempt to train the neural network (Sec. IV).

II. MNIST DATABASE AND NETWORK TOPOLOGY

At the first step, a simple deep neural network with one hidden layer is being trained. The neural network is schematically shown in Fig. 1.

Since a task of recognizing hand-written MNIST digits is being solved, an input signal of the ANN is an image of 28×28 pixels. Usually, such an image is transformed into a vector $\mathbf{X}$ of size 1×784 and is fed to the input layer of the ANN. This makes the first layer consisting of 784 simple linear neurons with the activation function

16 November 2023 09:23:46

Published under an exclusive license by AIP Publishing

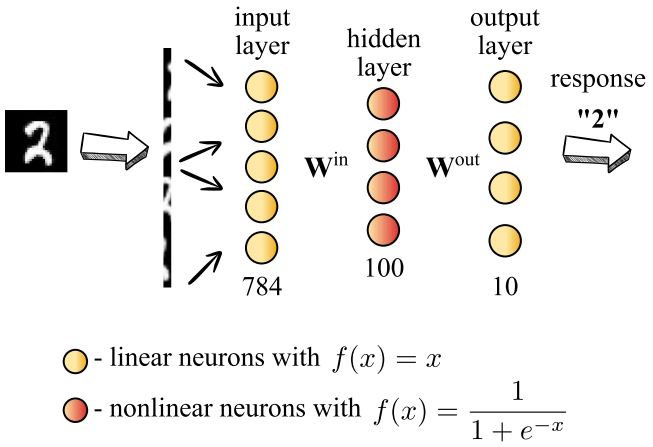

FIG. 1. Schematic illustration of a neural network under study. The artificial neurons with a linear activation function are colored in yellow, while the neurons with a nonlinear activation function, which will later be replaced by FHN systems, are marked in red.

$f(x) = x$. These neurons do not transform the signal, but merely pass it to the next layer. In order to do this, the vector $\mathbf{X}$ is multiplied by a corresponding matrix $\mathbf{W}^{in}$ of size 784×100. Thus, the input image is transformed into an input signal and is passed to the 100 neurons of the hidden layer. Within this layer, the neurons have a sigmoid activation function: $f(x) = 1/(1 + e^{-x})$. However, the choice of the function does not affect the subsequent results, and the use of a hyperbolic tangent function would lead to a similar outcome. Next, the signal from the 100 hidden neurons is fed to the output layer using the connection matrix $\mathbf{W}^{out}$. The output layer consists of ten neurons, which also use a linear activation function $f(x) = x$.

The response of the neural network is the index number of the output neuron producing the maximum output signal. This operation is also called softmax(). Specifically, if an image with number 2 is fed to the input of the ANN, as in Fig. 1, then the ANN response "2" will correspond to the situation when the neuron numbered $i = 2$ (where $i \in [0; 9]$) has the maximum output.

The MNIST database[20] was used to train the ANN. The database includes a training set (60 000 images of numbers 0–9) and a test set (10 000 images). To train the neural network, we used the Keras[21] library. This is a freely distributed API. The accuracy of the trained ANN on the training set was 99.5%, while the accuracy of training on the test set was 97.7%. The result of training is the connection matrices $\mathbf{W}^{in}$ and $\mathbf{W}^{out}$.

III. IMPLEMENTING THE FHN SYSTEMS INTO THE TRAINED ANN

In order to implement FHN systems in place of 100 artificial neurons in the hidden layer, one needs to understand the dynamics of the FHN system itself and how it should be fed with the input signal. After the input signal $\mathbf{X}$ is multiplied by the connection matrix $\mathbf{W}^{in}$, a vector of 100 values is obtained. These values are fed to the input of 100 neurons. Now, FHN systems play the role of the hidden layer neurons, and each of them is described by the following equations:[18,19]

$$\begin{aligned} \varepsilon \dot{x} &= x - \frac{x^3}{3} - y, \\ \dot{y} &= x + a + I(t), \end{aligned} \tag{1}$$

where x is an activator variable, while y is an inhibitor variable. This is a widely used form of the FHN system, where the parameter ε is responsible for the time scale, $I(t)$ is the input signal, and a is the control parameter. The system can demonstrate spiking dynamics in an oscillatory regime ($|a| < 1$) and the lack of oscillations with a stable equilibrium state in an excitable mode ($|a| > 1$). More general information about the considered system and the impact of parameter a is given in Appendix A.

If the input signal does not change in time but introduces an additional constant component $I(t) = I = \text{const}$ into the second equation, the sum $a + I$ allows one to influence the position of the vertical nullcline of the system (see Fig. 6, green dashed line). Thus, due to the input signal I, the system can establish either an oscillatory or excitable mode. For $a = 0$, the values $|I| < 1$ will correspond to the oscillatory mode, and $|I| > 1$ will establish the excitable one.

Since the neural network was initially trained in such a way that the hidden layer neurons have a "sigmoid" type activation function, they accept an input signal of the range $(-\infty; +\infty)$ and return an output signal of $(0; 1)$. In this case, the product of the input image vector $\mathbf{X}$ and the connection matrix $\mathbf{W}^{in}$ may contain such large numbers that they will not commensurate with the scale of the variables (x, y) of the FHN system. In order for the product $\mathbf{X} \cdot \mathbf{W}^{in}$ to be further used as an input signal of the FHN system, we propose to introduce the following normalization:

$$I = \gamma \cdot \tanh(\mathbf{X} \cdot \mathbf{W}^{in}). \tag{2}$$

Then, no matter how large the values of the matrix $\mathbf{W}^{in}$ are, after applying the hyperbolic tangent, the range of values is transformed into the interval $(-1; 1)$. The γ multiplier allows you to set the range of the values more precisely.

The output signal of the system is defined as follows:

$$Y = \text{softmax}(\vec{x} \cdot \mathbf{W}^{out}). \tag{3}$$

It is also computed with the use of the softmax() function as earlier, but now, it is a function of the product of the $\mathbf{W}^{out}$ matrix and the variable vector x_i of the 100 FHN systems. This makes the ANN response to be the index number of the output layer neuron, which has the maximum output signal.

Figure 2 shows temporal dependencies of the neural network response for three different input images containing the numbers 0, 3, and 5. Strictly speaking, Fig. 2 does not show an immediate output of the neural network. There is a transient time of 1000 dimensionless units $T^{trans} = 1000$, which is discarded in order not to consider the regime establishment process. Now, there is a problem of the result interpretation. Since FHN systems are spike systems and may show an oscillatory mode, the ANN response may also oscillate, and at some moments, "wrong" neurons can be activated. In order to interpret the response of the ANN correctly, we choose the answer that takes the most time of the entire control record of the ANN output signal $T = 100$; for example, in the Fig. 2, the longest response

16 November 2023 09:23:46

Published under an exclusive license by AIP Publishing

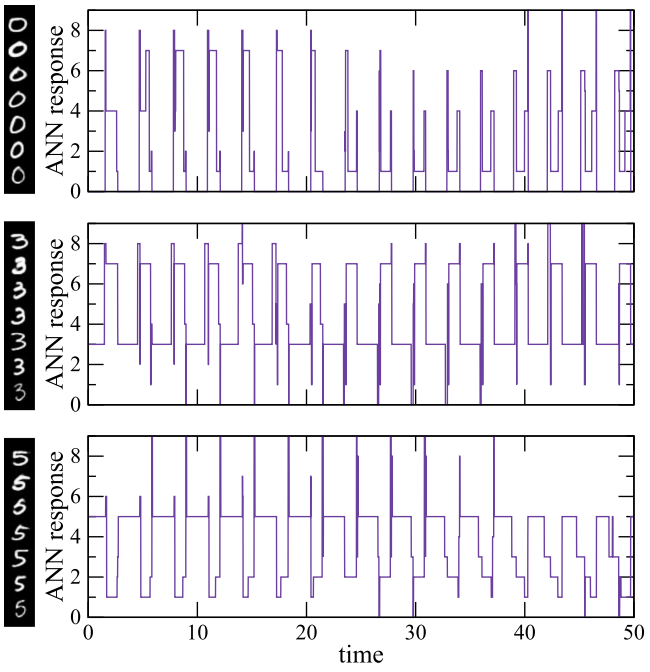

FIG. 2. Temporal evolution of the output of ANN with implemented FHN systems after transient time $T^{\text{trans}} = 1000$ for three different kinds of input image: digit "0" (top panel), "3" (middle), and "5" (bottom).

time is "0" for the top record, "3" for the middle one, and "5" for the bottom one.

Identical results were obtained for other digits. Table I shows the accuracy calculated for the digits 0–9 from the training and testing sets for the parameter $\gamma = -0.5$. We find that the average accuracy on the training set was 73.4% and the average accuracy on the test set was 73.3%. All accuracies hereinafter will be calculated after excluding the transient process during the time T^{trans}.

Average accuracies were also calculated for other values of γ. Figure 3 shows the dependence of the average accuracy on γ for the test set. As can be seen from the picture, the lowest accuracy can be obtained if $\gamma > 0$ and if γ is close to zero. Also, for $-1 < \gamma < 0$, the accuracy increases with the decrease of γ and saturates when $\gamma < -1$. In Eq. (2), the multiplier with hyperbolic tangent can only take values between -1 and 1. Therefore, the parameter γ controls both the scale of the parameter I and the mode of the FHN system. $|\gamma| < 1$ leads to the pure oscillatory regime when $|\gamma| > 1$ can lead to both excitable and oscillatory regimes depending on the value of $(\mathbf{X} \cdot \mathbf{W}^{\text{in}})$. The possibility of both regimes is accompanied with the largest accuracy for $\gamma < 1$. At the same time, a completely logical question arises why for the symmetrical case with $\gamma > 0$, the opposite effect occurs, and the recognition accuracy tends to 0. This is caused by the form of the output connection matrix $\mathbf{W}^{\text{out}}$ in (3). The already trained network was trained according to the sigmoid activation function in the hidden layer, and this imposes certain conditions on connection matrices. Therefore, by changing all the signs of the values inside the matrix $\mathbf{W}^{\text{out}}$ to the opposite, it is possible to achieve an inverse relationship between accuracy and γ, and then the maximum accuracy will be obtained for positive γ values.

TABLE I. Accuracies of ANN with implemented FHN systems applied to training (Tr.) and testing (Test.) datasets of each digit type. There are three ANNs with $\gamma = -0.5$, $\gamma = -1$, and $\gamma = 0.5$.

	$\gamma = -0.5$		$\gamma = -1$		$\gamma = 0.5$	
Digit	Tr.	Test.	Tr.	Test.	Tr.	Test.
0	99.0	98.6	88.9	99.2	6.7	6.4
1	1.4	1.5	16.3	16.4	0.0	0
2	83.3	82.6	91.8	90.3	5.6	5.6
3	84.6	85.1	92.7	91.9	3.2	3.7
4	90.4	88.0	98.0	94.3	1.0	0.6
5	29.6	31.9	52.0	54.9	0.3	1.1
6	96.9	96.2	99.9	98.0	8.6	9.4
7	56.5	57.5	80.4	78.6	0.9	0.8
8	99.8	99.5	100.0	99.0	6.5	6.0
9	92.3	91.9	98.9	96.6	1.36	1.98
Average	73.4	73.3	83.0	82.9	3.4	3.6

IV. ANN TRAINING

In order to speed up the processes of training and testing, not all the images from the training and test sets were used: 10,000 examples from the training set and 1000 examples of the test set with an equal amount of examples of the same digit were used.

The parameters of the FHN systems remained the same. Since ANN training is associated with a large number of runs of input images and connection matrices, in order to speed up this process, the settling time was reduced to $T^{\text{trans}} = 200$, but the control time $T = 100$ remained the same. This did not affect the accuracy when repeating the previously described steps.

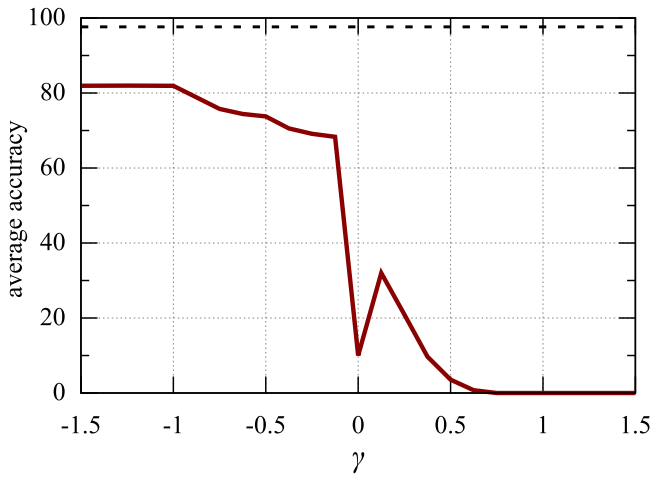

FIG. 3. Average testing accuracy of ANN with implemented FHN systems depending on the parameter γ.

16 November 2023 09:23:46

Published under an exclusive license by AIP Publishing

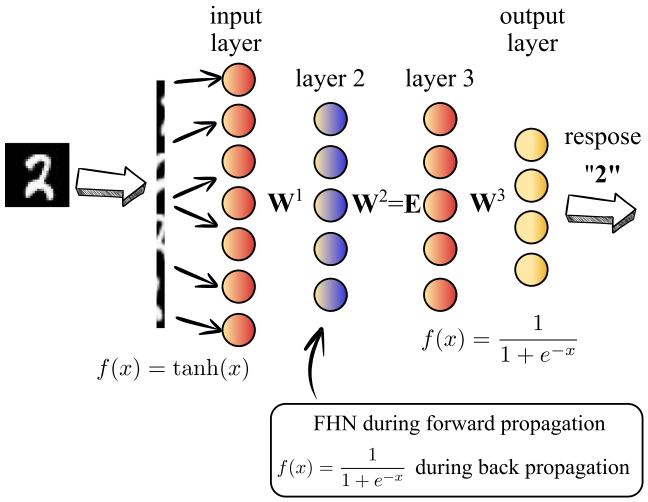

FIG. 4. Schematic illustration of a trainable ANN with FHN systems. The artificial neurons with a linear activation function are colored in yellow, while the neurons with a nonlinear activation function are marked in red. The violet color corresponds to FHN systems.

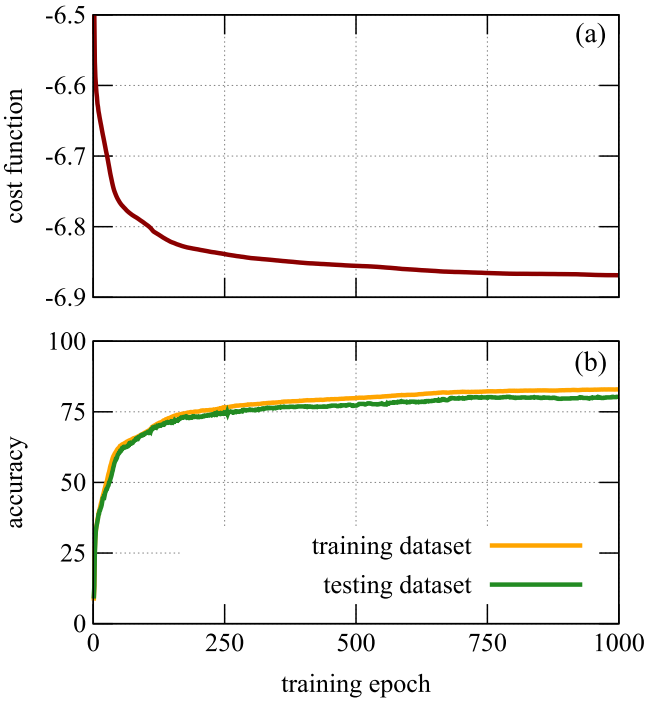

FIG. 5. Training process of ANN with FHN systems inside illustrated by cost function (a) and accuracies (b) on training and testing datasets depending on the training epoch.

There were some difficulties in the process of training the ANN. A large number of network topologies and several nonlinear activation functions were considered, and the number of layers was also being changed. In the end, we came to the conclusion that the optimal network topology looks like the one presented in Fig. 4. 784 pixels are fed to the ANN input; therefore, the input layer still consists of 784 neurons. In Sec. III, it was shown that before applying a signal to the FHN input, it must first be renormalized using a hyperbolic tangent; therefore, we added this processing step to the first layer in the new ANN. As a result, the number of neurons in the first layer remains the same, but their activation function becomes a hyperbolic tangent. The second layer consists of 100 FHN systems. The input layer is connected to the second layer by a $\mathbf{W}^1$ matrix 784×100.

The third layer was necessary to simplify the processes of learning and calculating derivatives. It contains 100 artificial neurons with a "sigmoid" activation function. Layers 2 and 3 are interconnected using the identity connection matrix $\mathbf{W}^2 = E$, which is fixed and does not change during the learning process; i.e., neurons of layers 2 and 3 are connected one-to-one throughout the training.

The output layer contains ten linear neurons with the function softmax(). The output layer is connected with the previous one with the connection matrix $\mathbf{W}^3$.

The interpretation of the output signal of the so-constructed ANN was the same as in Sec. III. An ANN's response over time $T = 100$ was the index number of the neuron, which produced the largest output signal for the longest time.

The training was carried out using the backpropagation method of logistic regression. Here, the following trick is applied. During the forward propagation, the FHN systems were used as layer 2 neurons, and during backpropagation, they were replaced by conventional artificial neurons with a "sigmoid" type activation function $f(x) = 1/(1 + e^{-x})$ for the correct calculation of the derivative. In Fig. 4, these neurons are represented in blue. In Fig. 5, the cost function (a) and accuracy on the training and test sets depending on the training epoch (b) illustrate the training process. The cost function was calculated according to

$$J = \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \left[-y_i^{\text{tr}} y_i^{\text{res}} - (1 - y_i^{\text{tr}})(1 - y_i^{\text{res}}) \right], \tag{4}$$

where $\mathbf{y}^{\text{tr}}$ is the set of N_{tr} correct answers and $\mathbf{y}^{\text{res}}$ is the set of corresponding ANN's responses.

The corresponding accuracies on the training and test sets by digits are given in Table II for both the truncated sets (for which Fig. 5 was built) and the full MNIST sets (for which we have already applied the trained network). The above results were obtained for the value $\gamma = -0.5$. The average accuracy of the resulting network is about 80%. In comparison of this value with the accuracy of the trained ANN with FHN systems embedded in it, this γ value corresponds to an accuracy of about 73%. Thus, after training with the proposed technique, it was possible to increase the accuracy of the neural network.

The distribution of the accuracy for different digits is of particular interest (see Table II and Fig. 7 in Appendix B). The resulting ANN does not work well with the number 8. If it is not included, the overall accuracy is about 90%.

16 November 2023 09:23:46

Published under an exclusive license by AIP Publishing

TABLE II. Accuracies of trained ANN with FHN systems applied to training (Tr.) and testing (Test.) datasets of each digit type. Parameter γ is set to -0.5.

Digit	Tr. set (10 000)	Test. set (1000)	Tr. set (60 000)	Test. set (10 000)
0	97.1	96.5	95.9	95.6
1	97.5	97.6	96.6	97.8
2	89.2	87.1	86.4	86.6
3	87.1	85.1	84.2	86.8
4	91.9	86.4	88.6	89.1
5	85.9	86.2	81.0	81.7
6	95.2	86.2	92.2	91.1
7	94.4	86.9	92.2	90.5
8	0.1	0	0.1	0.1
9	83.7	80.9	79.5	79.9
Average	82.2	79.3	79.7	79.9

V. CONCLUSION

We have managed to find well-working ways to introduce FitzHugh–Nagumo systems into artificial neural networks. This enables us to combine the neural network topology with the peculiarities of the FitzHugh–Nagumo spike dynamics. We have also been able to find conditions under which the resulting neural network demonstrates good accuracy about 90%.

The signal from the first layer is multiplied by the corresponding connection matrix, and then, it is sent to each of the 100 FHN systems in an ANN's hidden layer according to Eq. (2). The multiplier γ allows one to control the amplitude of the FHN input signal more precisely. We show here that only negative γ values lead to appropriate accuracy.

In addition, we proposed a method for training the ANN with introduced FHN systems, as schematically shown in Fig. 5. The resulting neural network increases the accuracy by ≈12% when compared to initially trained ANN, into which the FHN systems were subsequently inserted.

ACKNOWLEDGMENTS

This work was partially supported by the Russian President scholarship (No. SP-749.2022.5).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Tatyana Bogatenko: Writing – original draft (equal); Writing – review & editing (equal). **Konstantin Sergeev:** Investigation (equal); Visualization (equal). **Andrei Slepnev:** Investigation (equal); Methodology (equal). **Jürgen Kurths:** Conceptualization (equal); Supervision (equal). **Nadezhda Semenova:** Methodology (equal); Supervision (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: THE FITZHUGH–NAGUMO SYSTEM

The considered FHN system is given by the system of Eq. (1). Depending on the value of a, the system demonstrates the Andronov–Hopf bifurcation: for $|a| > 1$, a stable equilibrium state of the "focus" type is observed (excitable mode); if $|a| < 1$, the mode is called oscillatory, and the system exhibits periodic spike dynamics.

A description of the system (1) from the perspective of current and voltage is also common. Then, the variable x is a voltage-like membrane potential with cubic nonlinearity that allows regenerative self-excitation via a positive feedback. The variable y is called the recovery variable with linear dynamics that provides a slower negative feedback. The parameter I corresponds to a stimulus current. A positive current corresponds to a current directed from the outside of the cell membrane to the inside.

Figure 6 shows the phase plane of the system (1). Also, the corresponding activator $\dot{x} = 0$ and inhibitor $\dot{y} = 0$ nullclines are depicted. The activator nullcine corresponds to the $y = x - x^3/3$ line (Fig. 6, the orange line), while the inhibitor nullcline corresponds to the $x = -a$ line when there is no input signal (Fig. 6, the green line). For $a = 1$, the nullclines intersect at $x_0 = -1$, $y_0 = -2/3$. This point is a stable equilibrium state for $a > 1$.

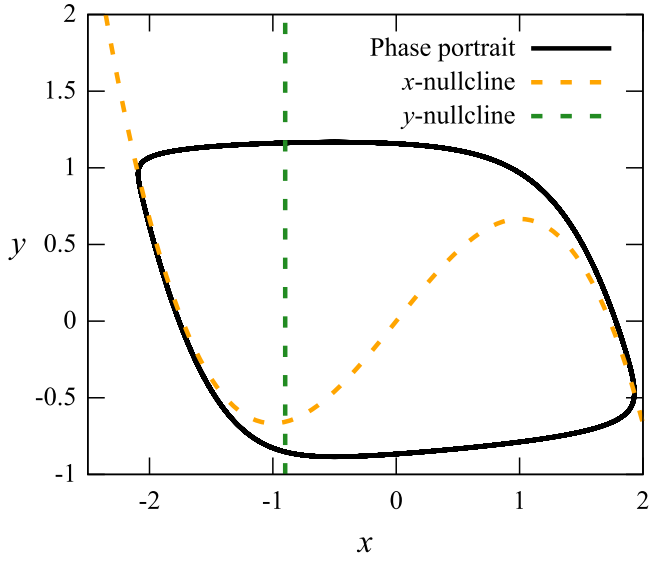

FIG. 6. Phase portrait of the FHN system (1) with corresponding $\dot{x} = 0$ (orange) and $\dot{y} = 0$ nullclines (green).

16 November 2023 09:23:46

Published under an exclusive license by AIP Publishing

APPENDIX B: TESTING ACCURACY OF THE TRAINED ANN ACCORDING TO THE DIGIT TYPE

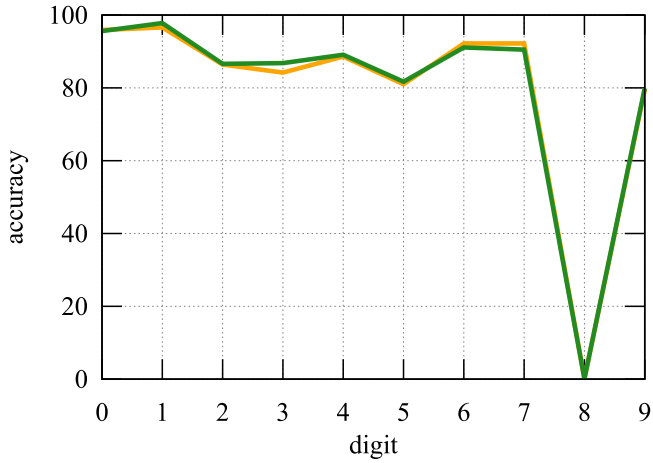

FIG. 7. Accuracy of ANN applied to training (60 000 examples, orange) and testing (10 000 examples, green) MNIST datasets grouped by the digit type. Corresponding data can be found in two right columns in Table II.

REFERENCES

[1]B. Hellwig, "A quantitative analysis of the local connectivity between pyramidal neurons in layers 2/3 of the rat visual cortex," Biol. Cybern. **82**, 111–121 (2000).
[2]O. Sporns, G. Tononi, and G. Edelman, "Connectivity and complexity: The relationship between neuroanatomy and brain dynamics," Neural Netw. **13**, 909–922 (2000).
[3]S. Druckmann, L. Feng, B. Lee, C. Yook, T. Zhao, J. Magee, and J. Kim, "Structured synaptic connectivity between hippocampal regions," Neuron **81**, 629–640 (2014).
[4]P. Tewarie, B. A. E. Hunt, G. C. O'Neill, A. Byrne, K. Aquino, M. Bauer, K. J. Mullinger, S. Coombes, and M. J. Brookes, "Relationships between neuronal oscillatory amplitude and dynamic functional connectivity," Cereb. Cortex **29**, 2668–2681 (2018).
[5]I. Basheer and M. Hajmeer, "Artificial neural networks: Fundamentals, computing, design, and application," J. Microbiol. Methods **43**, 3–31 (2000).
[6]O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," Heliyon **4**, e00938 (2018).
[7]A. Jain, J. Mao, and K. Mohiuddin, "Artificial neural networks: A tutorial," Computer **29**, 31–44 (1996).
[8]W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Bull. Math. Biophys. **5**, 115–133 (1943).
[9]F. Ponulak and A. Kasiński, "Introduction to spiking neural networks: Information processing, learning and applications," Acta Neurobiol. Exp. **71**, 409–433 (2011).
[10]A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," Neural Netw. **111**, 47–63 (2019).
[11]S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," Int. J. Neural Syst. **19**, 295–308 (2009).
[12]A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," J. Physiol. **117**, 500 (1952).
[13]R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," J. Neurophysiol. **94**, 3637–3642 (2005).
[14]E. Izhikevich, "Simple model of spiking neurons," IEEE Trans. Neural Netw. **14**, 1569–1572 (2003).
[15]J. Hasler and H. Marr, "Finding a roadmap to achieve large neuromorphic hardware systems," Front. Neurosci. **7**, 118 (2013).
[16]S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 37, edited by F. Bach and D. Blei (PMLR, Lille, France, 2015), pp. 1737–1746.
[17]G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nat. Rev. Phys. **3**, 422–440 (2021).
[18]R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," Biophys. J. **1**, 445–466 (1961).
[19]J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," Proc. IRE **50**, 2061–2070 (1962).
[20]Y. LeCun, "The MNIST database of handwritten digits" (1998), see http://yann.lecun.com/exdb/mnist/.
[21]F. Chollet *et al.*, "Keras," GitHub (2015), see https://github.com/fchollet/keras.

16 November 2023 09:23:46

Published under an exclusive license by AIP Publishing