Tuning weight values by resolving imbalances at nodes in an augmented artificial neural network

Gou Shinkai (goushinkai@icloud.com)
Hitachi, Ltd

Research Article

Keywords: artificial neural network, computational function, artificial intelligence, tuning weight values

Posted Date: May 17th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-370478/v2

Tuning weight values by resolving imbalances at nodes in an augmented artificial neural network

Author names

Gou Shinkai

Affiliations

Centre for Exploratory Research, R&D Department, Hitachi, Ltd., Kokubunji, Tokyo, 185-8601, Japan.

Corresponding author

Gou Shinkai

Centre for Exploratory Research, R&D Department, Hitachi, Ltd., 1-280, Higashi Koigakubo, Kokubunji, Tokyo, 185-8601, Japan.

+81 (80)-8719-3598

go.shinkai.xa@hitachi.com

Abstract

Tuning the weight values in an artificial neural network for a computational function is essential for artificial intelligence. This letter proposes tuning based on a mathematical model, which conceptually extends the artificial neural network by allowing an imbalance between the sum of the incoming values to a node and the outgoing values from the node in the network. In a different way from the gradient-descent-based tuning that uses multiple updates to minimise the difference between the required output values and the computed output values of an artificial neural network, the proposed tuning resolves the imbalance at each node in the network. The proposed tuning exhibits similar performance to the existing stochastic-gradient-descent-based tuning. In contrast, the proposed tuning does not need to explore the optimal parameters to obtain the optimal weight values. These benefits of the proposed tuning method could accelerate the advancement of artificial intelligence.

Introduction

Artificial intelligence (AI) utilises a mathematical model called an artificial neural network (ANN)[1-3] based on the study of biological neural network behaviours[4-6]. Like the neuron's dynamics in a biological neural network, the weight values (WVs) of an artificial neuron (AN) in an ANN determine the ANN's output values against its input values[4-6]. To design appropriate computational functions[7-11] with an ANN, tuning the WVs is essential.

In previous works, tuning ANNs was based on repeated updates of the WVs to minimise the difference between the target output values and the ANN's computed output values[12-15], whose difference is defined as the error in the tuning. This tuning requires careful consideration of the tuning rate of the WVs against the error[16] and the momentum of the WV's change between updates[13] to prevent the vanishing gradient problem[17,18] and local optima[3]. This complexity comes from schemes where the tuning depends on the feedback from the error[3,17,18]. Therefore, tuning being free from feedback from the outputs is worth investigating.

This letter reports an investigation of tuning under an augmented artificial neural network (AANN), which conceptually extends the ANN by allowing an imbalance between the sum of the incoming values and the outgoing value at each node in the ANN. Because of these imbalances, the values can accumulate at each node. An AANN is equivalent to the corresponding ANN if the imbalances are resolved at all nodes. Equivalence is achieved if the value's accumulation at each node converges with time. Once convergence is achieved, the WVs provided by the AANN can be used in the corresponding ANN, which does not allow imbalance at the nodes. Resolving the imbalance at the nodes can be performed independently and simultaneously.

An electronic circuit is a good way to study AANNs. The incoming and outgoing values of each node in the AANN can correspond to an electric current. The electric charge carried by the current can be accumulated at each circuit node capacitively coupled to the ground, which can be sensed according to the voltage of the node. A voltage difference between neighbouring nodes causes a current between them. In the circuit, balancing the incoming and outgoing currents at each node can be carried out by analogue signal processing[19]. The circuit can tune the WVs without avoiding a small gradient in the ANN's output against the WVs, while this avoidance is important for gradient-descent-based tunings[3,16-18]. AANN-based tuning could advance artificial intelligence from the viewpoint of addressing the issues[17,18] that existing tuning methods[3,16-18] have.

Results

An augmented artificial neuron

First, a brief explanation of how an augmented artificial neuron (AAN) is a building block of the AANN is given. The mathematical equation of an AAN is written as y = σ(u) and $u = \sum_i w_i x_i - dQ/dt$, where σ is an activation function[20], x_i (u) is the incoming (outgoing) value of the node in the AAN, w_i is the WV for x_i, y is the outgoing value from the ANN, and dQ/dt is the increment of the value Q at the node according to the imbalance between $\sum_i w_i x_i$ and u. If dQ/dt converges to zero by tuning the WVs optimally, the equation of the AAN coincides with that of the AN[1-3], i.e., $y=\sigma(\sum_i w_i x_i)$. Figure 1a is the schematic representation of the AAN. The realisation of the AAN in an electronic circuit is shown in Fig. 1b, where the incoming and outgoing values of the node are represented by an electric current, i.e., $\tilde{x}_1 = x_i I_0$, $\tilde{u} = uI_0$, $\tilde{y} = yI_0$, where I_0 =1 μA here. In the circuit, $w_i\tilde{x}_i$ is decomposed into $\tilde{x}_i$ and $(w_i - 1)\tilde{x}_i$. The voltage-controlled current sources (VCCSs) inject a current, $(w_i - 1)\tilde{x}_i$, to the node. This decomposing configuration ensures that the current from a node to its neighbouring node will be caused by the voltage difference between the nodes in the network configuration. The AAN's node voltage, V_n, represents the accumulation of electronic charge, Q, induced by current. The voltage comes from the capacitive coupling, C, between the node and the ground. The time dependence of the voltage is written as $dV_n/dt = C^{-1}\,dQ/dt = C^{-1}(\sum_i w_i \tilde{x}_i - \tilde{u})$. If $dV_n/dt \neq (=) 0$, it means that $\tilde{u} \neq (=) \sum_i w_i\tilde{x}_i$, which is illustrated in Fig. 1c. The details of achieving $dV_n/dt = 0$ in the circuit are explained in the next section. The mapping between ũ and $\tilde{y}$ can be implemented by using a bipolar junction transistor (BJT)[21] and an operational amplifier (opamp)[19], as shown in the inset of Fig. 1b. At a constant current, I_B, to the base of the BJT (not shown), the mapping between them has an S-shaped curve, as shown in Fig. 1d. Here, the opamp has a so-called voltage follower configuration, which results in $V_y = V_{y\prime}$. Therefore, ũ is divided into $\tilde{y}$ and $\tilde{u} - \tilde{y}$ if $\tilde{u} \gg I_B$, depending on the resistivity of the resistance and that of the BJT. The nonlinearity in the *I-V* characteristics of the BJT (not shown) causes the S-shaped curve. Note that it is known that the S-shaped activation function is related to the vanishing gradient problem[17,18] in gradient-descent-based tunings. However, AANN-based tuning gives the optimal WVs even if the S-shaped activation function is adopted since the AANN does not depend on the gradient of the network's output against the WVs.

Tuning by using an augmented artificial neural network

Second, the details of the tuning weight values for the AAN circuit are explained. Figure 2a shows the detailed circuit diagram of the AAN. The diagram has an equivalent configuration to that of Fig. 1b. Here, VCCSs are prepared for each incoming current $\tilde{x}_{\mathrm{i}}$. A VCCS is controlled by an analogue signal processing unit (ASPU). The incoming currents and the node voltages drive its ASPU. In the ASPU, $\tilde{x}_{\mathrm{i}}$ is converted to a voltage. However, for simplicity, this voltage is also denoted as $\tilde{x}_{\mathrm{i}}$. The conversion is performed by sensing the voltage drop in the electronic resistivity, r, with a differential amplifier[19]. The amplifier also works as a buffer amplifier[19], in which the input (output) impedance is high (low). Because of the high impedance, $\tilde{x}_{\mathrm{i}}$ can be sensed by the amplifier. Simultaneously, the time differential of V_{n}, $\mathrm{d}V_{\mathrm{n}}/\mathrm{d}t$, is sensed by the opamp with the capacitor and resistivity. As shown in Fig. 1c, the time differential represents the imbalance of current in the node. To eliminate $\mathrm{d}V_{\mathrm{n}}/\mathrm{d}t$ at rising and falling periods of $\tilde{x}_{\mathrm{i}}$ and/or $\tilde{y}$, a field-effect transistor[21] that is switched on and off by a square periodic signal wave is prepared. The second opamp with the transistor and resistivity acts as the multiplier and yields $-\tilde{x}_{\mathrm{i}}\mathrm{d}V_{\mathrm{n}}/\mathrm{d}t$. The third opamp with the resistivity and capacitor gives the integral of the value, i.e., $V_{\mathrm{w}} = -\int(\tilde{x}_{\mathrm{i}}\, \mathrm{d}V_{\mathrm{n}}/\mathrm{d}t)\mathrm{d}t$. The last opamp yields $V_{\mathrm{w}}\tilde{x}_{\mathrm{i}}$, which produces a current, $(\mathrm{w}_{\mathrm{N}} - 1)\tilde{x}_{\mathrm{i}}$, by the VCCS. Hence, the above V_{w} corresponds to $\mathrm{w}_{\mathrm{i}} - 1$. Here, the coefficients for each analogue signal processing step are omitted for simplicity. It is not difficult to merge the VCCSs in the circuit, which makes the circuit simpler. The VCCS can also be realised by circuit technologies[21]. The opamps and the differential amplifier have a sufficiently high open-loop gain and gain-bandwidth product[19], which are compared to the time scale described later.

Figure 2b shows the AANN constructed from the N-layered AANs. In each layer, there are nine AANs. In the network, the blue dots in the figure represent the AANs. In the first layer, $\tilde{x}_{\mathrm{i}}$ is copied and inputted to each AAN in the first layer. The copying can be done by current mirror circuits[21]. In each following layer, the current produced by the AAN in the previous layer is copied and supplied to each following AAN except for the AAN in the same line, represented by a magenta line. In each magenta line, the current from an AAN to the following AAN is driven by the voltage difference between their nodes. In contrast, a change in the voltage on the node of an AAN in the network affects not only the current from the AAN in the previous layer but also the current to the following AAN. In this manner, tuning the WVs of an AAN in the network affects the tuning of the other AANs. This mutual effect caused by the voltage difference between the nodes enables the AANN to obtain the optimal WVs by locally and independently performed tuning. For the tuning demonstration, AANNs with one to five layers were prepared.

Tuning the WVs in these AANNs is demonstrated by the circuit simulator SPICE[22], which is widely used in electronics. In the simulation, ideal opamps and VCCSs are used to reduce the simulation time. As a demonstration, five black-and-white images of 3×3 pixels, which correspond to “H”, “I”, “T”, “A”, and “C”, were prepared (Fig. 2c). Image recognition is a popular application in AI[23,24]. The pixels in each image, x_1, x_2, …, x_9, are assigned from the top left to the bottom right, as shown in the image. If a pixel is black (white), $x_i = 2$ (0). For example, image-1 corresponds to $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\} = \{2, 0, 2, 2, 2, 2, 2, 0, 2\}$. On the other hand, the target output for image-1 is $\{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9\} = \{3, 0, 0, 0, 0, 0, 0, 0, 0\}$. Similarly, the target output values for image-j are $y_j = 3$ and $y_{k\neq j} = 0$. The optimal WVs for the prepared AANNs under the given images and their target outputs can be obtained by the following procedure: The current, $\tilde{x}_\mathrm{i}$, is injected on the left side of the AANNs, and $\tilde{y}_\mathrm{i}$ is drained from the right side of the AANNs simultaneously. The values of $\tilde{x}_\mathrm{i}$ and $\tilde{y}_\mathrm{i}$ depend on the images. To obtain the optimal WVs for all images, $\tilde{x}_\mathrm{i}$ and $\tilde{y}_\mathrm{i}$ change periodically, as shown in Fig. 2d. In the first duration of Δt, $\tilde{x}_\mathrm{i}$ and $\tilde{y}_\mathrm{i}$ correspond to image-1 and its target output, respectively. Similarly, in the subsequent durations, $\tilde{x}_\mathrm{i}$ and $\tilde{y}_\mathrm{i}$ correspond to image-2, image-3, image-4, image-5 and their target outputs, respectively. One round of tuning consists of five durations, $5\Delta t$. The rounds are repeated. In the simulation, $\Delta t = 1$ second to remove frequency-related complexities from the demonstration. The rise and fall time of $\tilde{x}_\mathrm{i}$ and $\tilde{y}_\mathrm{i}$ is 10 msec.

Image recognition under the tuned WVs

Third, image recognition by the ANN in which the WVs are tuned by the AANN is described. Figure 3a and b show the time evolution of the voltage on the nodes in the first and second layers of the two-layered AANN, respectively. A voltage fluctuation can be seen before the 10th round, and it disappears before the 100th round. Figure 3c shows the V_w of the AAN in the first line of the first layer of the AANN. Figure 3d shows the V_w of the AAN in the first line of the second layer of the AANN. The rest of the V_w values are shown in the Supplementary material. Each V_w also converges before the 100th round. By sensing these V_w at the 100th round, the WVs for the corresponding two-layered ANN can be obtained. The output of the ANN with the given WVs obtained by the AANN is calculated for the 2^9=512 black-and-white images, including the images in Fig. 2c. Figure 3e shows the black-and-white images recognised as “H”, “I”, “T”, “A” or “C” at a threshold of $y_i > 2.95$. Thirty-nine images, including image-1 (highlighted by the magenta frame), are recognised as “H”. They include eight (twenty-one) images where one (two) pixel(s) is (are) flipped compared with the true image of “H”, which are highlighted by a blue (grey) frame. The rest of the images have three flipped pixels compared with the true image of “H”. Similarly, one true image, one one-pixel-flipped image and three two-pixel-flipped images are recognised as “I”. One true image, two one-pixel-flipped images and one two-pixel-flipped image are recognised as “T”. One true image and four one-pixel-flipped images are recognised as “A”. One true image and two two-pixel-flipped images are recognised as “C”. The numbers of images recognised as “H”, “I”, “T”, “A” or “C” against the threshold of y_i are shown in the Supplementary material.

Comparison with stochastic-gradient-descent-based tuning

Finally, AANN-based tuning is compared with stochastic-gradient-descent-based tuning. The tuning error was evaluated by the root-mean-squared error (RMSE), defined as RMSE =

$\sqrt{(1/450)\sum_{j=1}^{50}\sum_{i=1}^{9}\left(y_{j,i}-\hat{y}_{j,i}\right)^2}$, where $y_{j,i}$ is the i-th output for the j-th image and $\hat{y}_{j,i}$ is the corresponding target value. Here, 50 images, including five true images and 45 images in which one pixel is flipped compared with the true images, are tested (see the Supplementary material). The RMSE under AANN-based tuning is compared with that under a stochastic-gradient-descent tuning method called ADAM[16]. ADAM does not work if all WVs are one before the tuning since there is no gradient descent against the WVs, while the AANN-based tuning works even in this condition. Therefore, distributed WVs are allowed before tuning only for ADAM. Therefore, at the beginning of the tuning, the RMSE for ADAM is smaller than that for the AANN-based tuning (see Fig. 4a and b). ADAM works under the four parameters β1, β2, ε, and the tuning rate (TR). Generally, β1 and β2 are close to one. ε is close to zero[16]. The minimum RMSE is explored among β1, β2 $\in$ {0.9, 0.92, 0.94, 0.96, 0.98, 0.9, 0.99, 0.999}, ε $\in$ {10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}}, and TR $\in$ {10^{6}, 10^{7}, 10^{8}, 10^{9}, 10^{10}, 10^{11}, 10^{12}, 10^{13}, 10^{14}, 10^{15}}. Hence, ADAM tuning was performed for $8\times8\times6\times10 = 3840$ combinations. The sigmoid function was used as the activation function in ADAM (see the Supplementary material). Figure 4a and b show the RMSE for one- to five-layered ANNs in which the WVs are given by AANN and ADAM, respectively. The dark blue, orange, olive, dark red, and purple lines correspond to the one-, two-, three-, four-, and five-layered ANNs, respectively. In Fig. 4a, these lines, except for the one-layered ANN, decrease monotonically and converge before the 100th round. On the other hand, Fig. 4b shows zig-zag lines since the minimum RMSE depends on β1, β2, ε, and TR in each round. Figure 4c shows the minimum RMSE among the rounds in Fig. 4a and b against the number of ANN layers. The AANN-based tuning and ADAM tuning show similar RMSEs. This means that AANN-based tuning can be competitive with ADAM tuning in terms of RMSE, while AANN-based tuning does not need to explore the parameters.

Data availability

The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

Code availability

The codes generated during the current study are available from the corresponding author on reasonable request.

Author contributions

GS designed the study, performed the experiments, analyzed the data, and wrote the manuscript.

Competing interests

The author has no conflicts of interest to declare.

References

1. McCulloch, W. S. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5,** 115-133 (1943).
2. Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65,** 386-408 (1958).
3. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436-444 (2015).
4. Hebb, D. O. *The Organization of Behavior: A Neuropsychological Theory* (Wiley and Sons, New York, NY, 1949).
5. Keysers, C. & Gazzola, V. Hebbian learning and predictive mirror neurons for actions, sensations and emotions. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **369,** 20130175 (2014).
6. Shackleton-Jones, N. *How People Learn: Designing Effective Training to Improve Employee Performance* (Kogan Page Ltd., London, UK, 2019).
7. Silver, D. et al. Mastering the game of Go without human knowledge. *Nature* **550,** 354-359 (2017).
8. Bressem, K. K. et al. Comparing different deep learning architectures for classification of chest radiographs. *Sci. Rep.* **10,** 13590 (2020).
9. Gómez, P. A. et al. Rapid three-dimensional multiparametric MRI with quantitative transient-state imaging. *Sci. Rep.* **10,** 13769 (2020).

10. Kong, X. et al. Artificial intelligence enhanced two-dimensional nanoscale nuclear magnetic resonance spectroscopy. *npj Quantum Inf.* **6,** 79 (2020).

11. Zhao, H. et al. Molecular imaging and deep learning analysis of uMUC1 expression in response to chemotherapy in an orthotopic model of ovarian cancer. *Sci. Rep.* **10,** 14942 (2020).

12. Nishimori, H. *Statistical Physics of Spin Glasses and Information Processing: An Introduction* (Oxford University Press, Oxford, UK, 2001).

13. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323,** 533-536 (1986).

14. Amari, S. A theory of adaptive pattern classifiers. *IEEE Trans. Electron. Comput.* **EC-16,** 299-307 (1967).

15. Alpaydın, E. *Introduction to Machine Learning* (The MIT Press, Cambridge, MA, 2010).

16. Kingma, D. P. & Ba, J. ADAM: a method for stochastic optimization in *The 3rd International Conference for Learning*. 13 (ICLR, San Diego, CA, 2015).

17. Hochreiter, S., Bengio, Y., Frasconi, P. & Schmidhuber, J. Gradient flow in recurrent nets: the difficulty of learning longterm dependencies in *A Field Guide to Dynamical Recurrent Networks* (eds Kolen, J. F. & Kremer, S. C.) 237-243 (Wiley-IEEE Press, New York, NY, 2001).

18. Goh, G. B., Hodas, N. O. & Vishnu, A. Deep learning for computational chemistry. *J. Comput. Chem.* **38,** 1291-1307 (2017).

19. Mancini, R. *Op Amps for Everyone* (Texas Instruments, Dallas, Texas, 2002).

20. Ramachandran, P., Zoph, B. & Le, Q. V. Searching for activation functions. *arXiv 1710.05941* (2017).

21. Sansen, W. M. *Analog Design Essentials* (Springer, New York, NY, 2006).

22. Brocard, G. *The LTspice IV Simulator: Manual, methods and applications* (Swiridoff Verlag, Künzelsau, Germany, 2013)

23. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 770-778 (IEEE, Las Vegas, NV, 2016).

24. Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. V. Learning transferable architectures for scalable image recognition in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 8697-8710 (IEEE, Salt Lake City, UT, 2018).

Competing interests

The author has no conflicts of interest to declare.

Figure legends

Fig. 1. An augmented artificial neuron.

a Schematic of an AAN with n inputs. The hexagons on the left and right sides indicate the junction points where values come in or go out. The blue dot is the node where the value accumulates because of the imbalance between the values that come into the node and those that go out of the node. The box represents a nonlinear mapping between u and y. **b** The corresponding circuit configuration of the ANN shown in **a**. The inset shows the circuit configuration for the nonlinear mapping between $\tilde{u}$ and $\tilde{y}$. **c** Schematic of the time dependence of V_n under imbalanced and balanced conditions. In the blue region, the current at the node is imbalanced, which appears as a rising V_n. The descent of V_n also indicates an imbalance in the node. In the magenta region, the current in the node is balanced, which appears as a converged V_n. **d** Mapping between $\tilde{u}$ and $\tilde{y}$ in the inset of **b**. Here, $V_H \approx 3.08$ µA and $V_L \approx -0.01$ µA.

Fig. 2. Weight tuning by the convergence of voltage under a constant current.

a The detailed circuit configuration of the AAN. The inset shows the operation of the ASPU in balancing the current at the node, in which the voltage is V_n. **b** AANN with N layers. Each blue dot represents an AAN. The magenta lines show unmediated connections between the AANs. The blue lines show connections mediated by controlled current sources. **c**, Images prepared for the demonstration of AANN-based tuning. **d** The time sequence of $\tilde{x}_1$ and y_1.

Fig. 3. Image recognition as a demonstration of AANN-based tuning by using a two-layered AANN.

a-b Time evolution of the node voltage in the first and second layers of the AANN. **c-d** Time evolution of each V_{w} of the AAN in the first line and the first layer and that of the ANN in the first and second layers in the AANN, respectively. Here, the notation $V_{\mathrm{wi\rightarrow j}}^{(\mathrm{N})}$ represents the V_{w} related to the coefficient of the current flowing from the i-th AAN in the (N−1)-th layer to the j-th AAN in the N-th layer if N $\geq$ 2. If N = 1, $V_{\mathrm{wi\rightarrow j}}^{(\mathrm{N})}$ represents the V_{w} related to the coefficient of the current flowing from the i-th current source in the 0th layer to the j-th AAN in the first layer (see Fig. 2b). The rest of the V_{w} values are shown in the Supplementary material. **e** Black-and-white images recognised as "H", "I", "T", "A" and "C" by the ANN in which the WVs are tuned by the AANN. They correspond to $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\} = \{2, 0, 2, 2, 2, 2, 2, 0, 2\}$, $\{0, 2, 0, 0, 2, 0, 0, 2, 0\}$, $\{2, 2, 2, 0, 2, 0, 0, 2, 0\}$, $\{0, 2, 0, 2, 0, 2, 2, 0, 2\}$ and $\{2, 2, 2, 2, 0, 0, 2, 2, 2\}$, respectively.

Fig. 4. Comparison of AANN-based tuning and ADAM tuning.

a RMSEs against the number of tuning rounds for the one-, two-, three-, four- and five-layered AANs. The WVs in the AANs are tuned by using the corresponding AANNs. **b** RMSEs against the number of tuning rounds for the one-, two-, three-, four- and five-layered AANs. The WVs in the AANs are tuned by ADAM. **c** Minimum RMSE values for the one-, two-, three-, four- and five-layered AANs, in which the WVs are tuned by ADAM or the corresponding AANNs. The tuning parameters for the one-, two-, three-, four- and five-layered AANs are {TR, β1, β2, ε} = {10^{11}, 0.9, 0.9, 10^{-3}}, {10^{13}, 0.9, 0.999, 10^{-1}}, {10^{11}, 0.9, 0.92, 10^{-2}}, {10^{12}, 0.92, 0.92, 10^{-1}}, {10^{8}, 0.9, 0.999, 10^{-6}} and {10^{11}, 0.92, 0.999, 10^{-3}}, respectively.

Figures

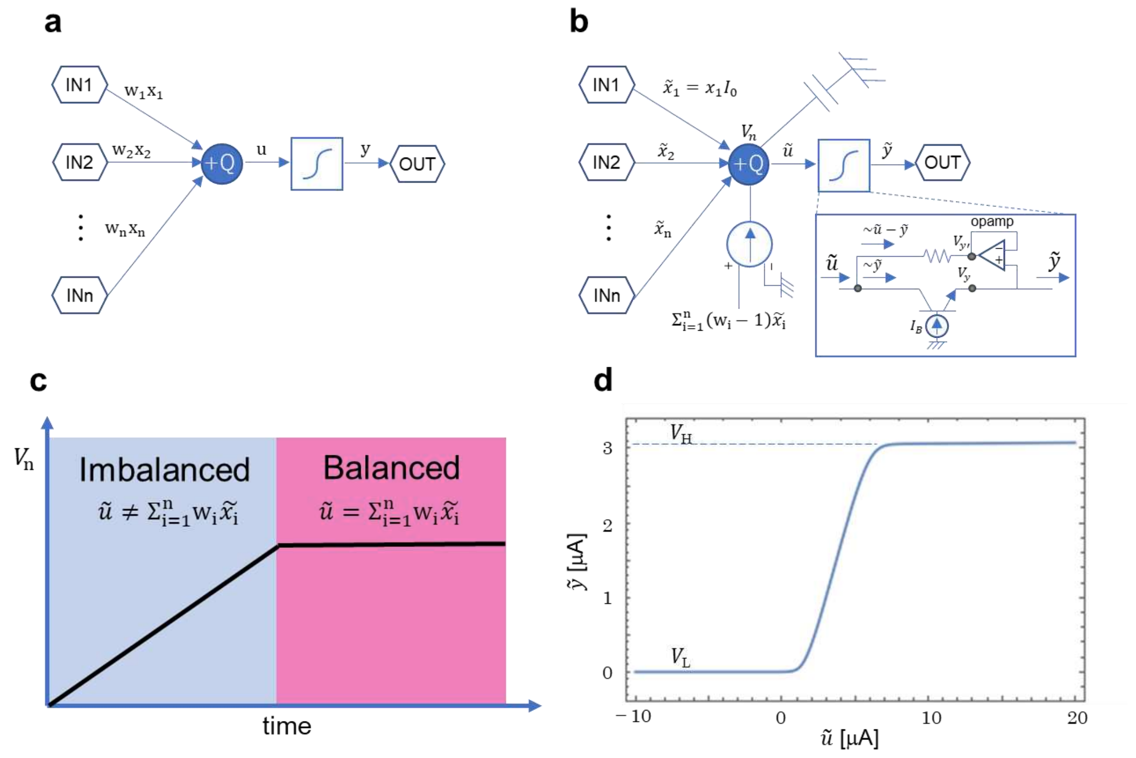

Fig. 1

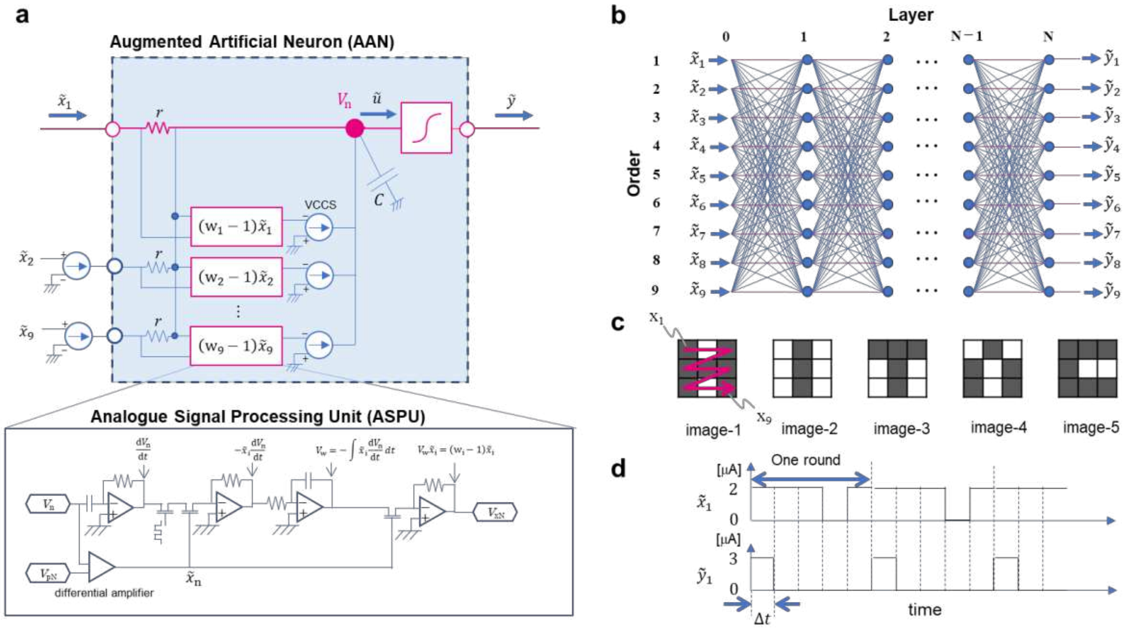
a
Augmented Artificial Neuron (AAN)
$\tilde{x}_1$
r
V_n
$\tilde{u}$
$\tilde{y}$
C
VCCS
$(w_1-1)\tilde{x}_1$
$(w_2-1)\tilde{x}_2$
$(w_9-1)\tilde{x}_9$
$\tilde{x}_2$
$\tilde{x}_9$
Analogue Signal Processing Unit (ASPU)
$\frac{dV_n}{dt}$
$-\tilde{x}_i\frac{dV_n}{dt}$
$V_w=-\int\tilde{x}_i\frac{dV_n}{dt}dt$
$V_w\tilde{x}_i=(w_i-1)\tilde{x}_i$
V_n
V_{pN}
$V_{\Delta N}$
$\tilde{x}_n$
differential amplifier
b
Layer
0
1
2
N − 1
N
Order
1 2 3 4 5 6 7 8 9
$\tilde{x}_1$ $\tilde{x}_2$ $\tilde{x}_3$ $\tilde{x}_4$ $\tilde{x}_5$ $\tilde{x}_6$ $\tilde{x}_7$ $\tilde{x}_8$ $\tilde{x}_9$
$\tilde{y}_1$ $\tilde{y}_2$ $\tilde{y}_3$ $\tilde{y}_4$ $\tilde{y}_5$ $\tilde{y}_6$ $\tilde{y}_7$ $\tilde{y}_8$ $\tilde{y}_9$
c
x_1
x_9
image-1
image-2
image-3
image-4
image-5
d
One round
[μA]
2
0
$\tilde{x}_1$
[μA]
3
0
$\tilde{y}_1$
Δt
time

Fig. 2

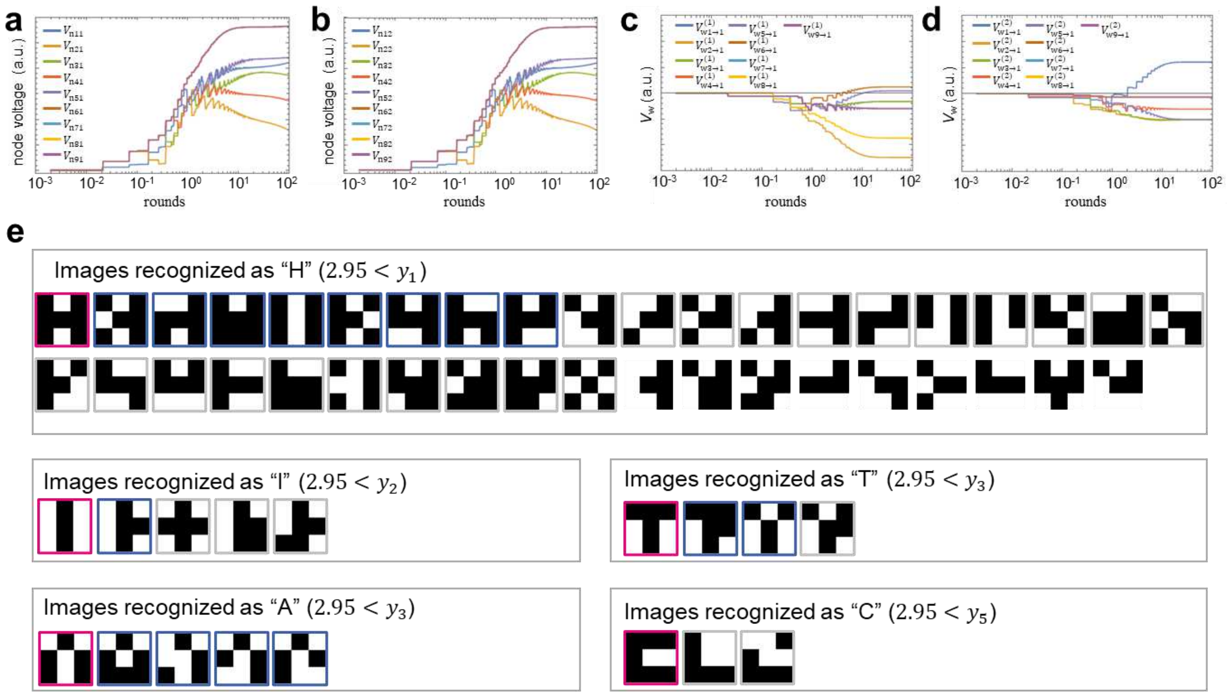
a
b
c
d
e
node voltage (a.u.)
V_w (a.u.)
rounds
V_{n11}
V_{n21}
V_{n31}
V_{n41}
V_{n51}
V_{n61}
V_{n71}
V_{n81}
V_{n91}
V_{n12}
V_{n22}
V_{n32}
V_{n42}
V_{n52}
V_{n62}
V_{n72}
V_{n82}
V_{n92}
$V^{(1)}_{w1\to1}$ … $V^{(1)}_{w9\to1}$
$V^{(2)}_{w1\to1}$ … $V^{(2)}_{w9\to1}$
Images recognized as "H" $(2.95 < y_1)$
Images recognized as "I" $(2.95 < y_2)$
Images recognized as "T" $(2.95 < y_3)$
Images recognized as "A" $(2.95 < y_3)$
Images recognized as "C" $(2.95 < y_5)$

Fig. 3

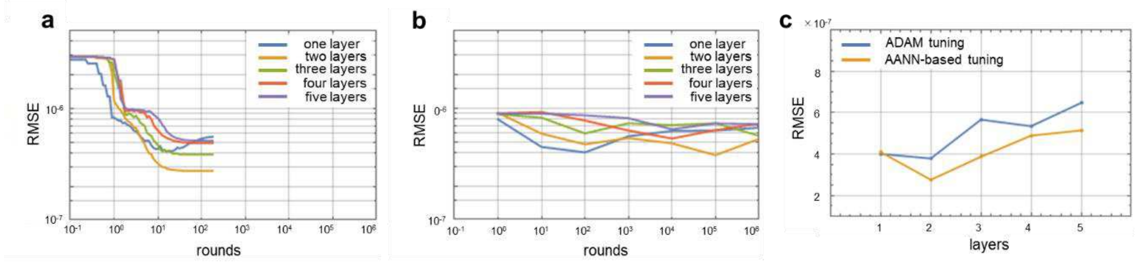
a
b
c
one layer
two layers
three layers
four layers
five layers
ADAM tuning
AANN-based tuning
RMSE
rounds
layers
×10⁻⁷

Fig. 4