Hetero-Correlation-Associative Memory with Trigger Neurons: Accumulation of Memory through Additional Learning in Neural Networks

Hiroshi Inazawa
Center for Education in Information Systems
Kobe Shoin Women's University
1-2-1 Shinohara-Obanoyama, Nada Kobe 657-0015, Japan

In this paper, we present a hetero-correlation-associative memory model that allows additional learning without destroying existing stored data, where the model is a feed-forward neural network consisting of two layers. The number of neurons in the input layer (called "trigger neurons") increases as the number of stored images increases. One trigger neuron is linked to one image to be learned. Each time an image to be learned is added, a new trigger neuron is also added, thereby enabling the model to learn the additional image. Moreover, since the learning process simply adds new trigger neurons, it does not influence previously learned data. We can store images in the network as necessary, one after another. The stored images can be recalled through the firing of the corresponding trigger neuron. The recalled images are approximately perfect matches to the teacher images. We also show that using the proposed learning procedure greatly improves the memory rate compared with the conventional one.

Keywords: additional learning; trigger neuron; hetero-correlation-associative memory; memory rate

1. Introduction

In this paper, we present a hetero-correlation-associative memory model involving neural networks in which the number of neurons changes dynamically. The model can learn images at any time, and the learning process does not influence previously learned data. Consequently, we can store images in the networks as necessary, one after another. The proposed learning and recall mechanism allows for the storage of many images in a neural network. This greatly improves the memory rate as compared with the conventional mechanism, where we define the percentage of the number of images that can be stored in the neural network as the memory rate. For example, the rate approaches 100% as the number of neurons in a network increases.

https://doi.org/10.25088/ComplexSystems.27.2.187

We use feed-forward neural networks with two layers, where the first layer consists of neurons, called "trigger neurons," whose number increases dynamically. The second layer consists of a fixed number of neurons and expands the layer to two dimensions. The stored images are recalled and displayed on the second layer. A trigger neuron is linked to an image it is to store, and the network learns this relation using the gradient descent method (GDM) [1–3]. In addition, the trigger neuron gives a cue to recall a stored image; in other words, stored images can be recalled by firing the corresponding trigger neurons.

Learning methods using neural networks have been very successful [4–6]. Various ideas on learning have been actively reported since around 2000, especially in regard to image recognition [7–9]. The topic of such a study is generally referred to as deep learning, which has been highly successful in the field of artificial intelligence. Many studies of deep learning are based on work that had been conducted on multilayered neural networks from 1980 to the beginning of 1990, a period that has been called the second neural network boom (2nd NN) [3]. It can almost be said that the structure and the framework of current neural networks originate from the studies in the 2nd NN [3, 10–12]. However, various problems arose in the learning processes for multilayered neural networks in the 2nd NN, and the research had therefore subsided [13]. However, steady research on how to make multilayered neural networks learn has continued [14], resulting in promising solutions for problems that first arose in the 2nd NN [15, 16]. Today's successes are the result of studies conducted during the 2nd NN and of the research that was conducted steadily after that.

In addition, many studies on associative memory have been reported, proposing important models with excellent concepts and fundamental structures [17–21]. The energy localization model presented by Hopfield made substantial progress on NP problems, such as the traveling salesman problem [22, 23]. A problem with these associative memory models has been the poor memory rate of memory data. There have been many excellent research reports regarding the rate [20, 22, 24], according to which the rate is approximately 15% of the number of neurons in an output layer, though deep learning might improve the rate. However, until now, no breakthrough in improving the rate has been made. Considering these problems together, it is very important to achieve a large memory rate and to find procedures of additional learning that do not destroy previously stored data. In this paper, we propose a hetero-correlation-associative memory model for achieving additional learning without destroying previous data and for improving the memory rate. In Section 2, we describe the specification of the model. The method of simulation and

the results are presented in Section 3. We summarize our findings and discuss future work in Section 4.

2. Specification of the Model

The proposed model is classified in neural networks as a hetero-correlation associative memory (cross-correlation model consisting of two layers). The first layer is called the trigger net (T-net). The number of neurons in the T-net increases according to the number of images stored. We call the neurons in the T-net trigger neurons. The number of trigger neurons is equal to the number of stored images. The second layer is called the memory net (M-net), in which the number of neurons is fixed to the pixel value of the stored image. The shape of the M-net spreads in two dimensions. On the other hand, the T-net can have any shape. The model is fully connected between the T- and M-nets, but has no connections within the same layer. The trigger neurons of the T-net send a signal to all the neurons of the M-net; that is, one trigger neuron has one input and multiple outputs. On the other hand, the neurons in the M-net have multiple inputs and one output. Regarding the storage and recall of one image, one trigger neuron in the T-net fires a signal. All the neurons in the M-net are connected to all the trigger neurons in the T-net, but always receive input from one trigger neuron regarding storage and recall of one image. Learning is executed by changing the value of the synaptic weight of the neurons in the M-net.

The fundamental feature of this model is that one trigger neuron in the T-net learns one image showing on the M-net (teacher image). The learning data is stored as values of the synapse weight w_{ji}^p, where p is the stored image number, i is the trigger neuron number and j is the neuron number in the M-net. Thus, the firing of the trigger neuron brings about the recall of the corresponding image on the M-net.

We now describe the detailed behavior of the model. The signal x_i^p of the trigger neuron has binary values (zero or one), but the output y_j^p from a neuron in the M-net has a real value. The neuron numbers j are numbered left to right and top to bottom in the two-dimensional plane of the M-net. The activation function of y_j^p uses a rectified linear function. We define the function as:

$$y_j^p = u_j^p \equiv \sum_{i=0}^{N} \left(w_{ji}^p + b_j\right) x_i^p = \begin{cases} 0 & \text{for } u_j^p < 0 \\ \displaystyle\sum_{i=0}^{N} \left(w_{ji}^p + b_j\right) x_i^p & \text{for } u_j^p \geq 0, \end{cases} \tag{1}$$

https://doi.org/10.25088/ComplexSystems.27.2.187

where N is the number of trigger neurons in the T-net. b_j indicates a bias that has the same value for all neurons in the M-net. The learning is executed using GDM, and we treat it as a regression task. Let us describe the learning formula of the weight w_{ji}^p. We first define the error function as:

$$E^p \equiv \frac{1}{2}\sum_{j=1}^{M}\left(d_j^p - y_j^p\right)^2, \tag{2}$$

where d_j^p is a teacher signal that is the element value of constructing the image on the M-net, and M is the number of neurons in the M-net. The update equations (the learning formulas) of the synapse weight are obtained as follows:

$$w_{ji}^p(t+1) = w_{ji}^p(t) + \Delta w_{ji}^p(t), \tag{3}$$

$$\Delta w_{ji}^p(t) \equiv -\varepsilon\frac{\partial E^p}{\partial w_{ji}^p(t)} = \begin{cases} 0 & \text{for } u_j^p < 0 \\ \varepsilon\left(d_j^p - y_j^p\right)x_i^p & \text{for } u_j^p \geq 0, \end{cases} \tag{4}$$

where t denotes the update count, and ϵ is a learning rate. Since there is only one firing trigger neuron for learning or recall for a stored image, no other neuron in the T-net fires. Assuming that the firing neuron is the a^{th} neuron, the input-output relation (1) at t is rewritten as:

$$y_j^p = \begin{cases} 0 & \text{for } u_j^p < 0 \text{ or } i \neq a \\ \left(w_{ja}^p(t) + b_j\right)x_a^p & \text{for } u_j^p \geq 0 \text{ and } i = a. \end{cases} \tag{5}$$

Rewriting the learning equations (3) and (4) using equation (5) yields:

$$w_{ji}^p(t+1) = \begin{cases} w_{ji}^p(t) & \text{for } i \neq a \text{ or } u_j^p < 0 \\ w_{ja}^p(t) + \varepsilon\left(d_j^p - \left(w_{ja}^p(t) + b_j\right)x_a^p\right)x_a^p & \text{for } i = a \text{ and } u_j^p \geq 0. \end{cases} \tag{6}$$

Here, let us examine the recall process with the given equations. Since x_a^p is 1.0, if ϵ is 1.0, then equation (6) becomes:

$$w_{ji}^p(t+1) = \begin{cases} w_{ji}^p(t) & \text{for } i \neq a \text{ or } u_j^p < 0 \\ d_j^p - b_j & \text{for } i = a \text{ and } u_j^p \geq 0. \end{cases} \tag{7}$$

Using this $w_{ji}^p(t+1)$ for the equation (5) at $t+1$, the equation becomes:

$$y_j^p = \begin{cases} 0 & \text{for } u_j^p < 0 \text{ or } i \neq a \\ d_j^p & \text{for } u_j^p \geq 0 \text{ and } i = a. \end{cases} \tag{8}$$

Thus, we can obtain the complete recalled image from w_{ji}^p after the learning process.

According to this method, every time a new image to store is presented, the new synapse weights are added to the neurons in the M-net. Therefore, a new trigger neuron is added to the T-net to learn a new image, and the added new synapse weights are used to learn the new image in the same manner. With this mechanism, the additional learning processes have no effect on the previously learned data.

3. Simulation and Results

First of all, let us explain the specification of the simulation. We use the dataset "train-images-idx3-ubyte" from the MNIST database [25], which consists of 60 000 images of handwritten characters, as the teacher images for learning. In the simulation, we select the images one by one for learning, and the trigger neurons of the T-net are also added one by one. The number of learning instances for an image is set to one. Note that E^p becomes zero at $t = 1$, as can be seen from equation (8). The number of neurons in the M-net is fixed to be $28 \times 28 = 784$ pixels for each image. Since the image data is monochrome, we use decimal integers from 0 to 255 as the image data. Furthermore, these decimal numbers are normalized so that 255 is 1. We denote the resulting data as d_j^p.

We can understand from equation (8) that the initial values of the synapse weights and bias are not related to learning or recall. Thus, the initial values of w_{ji}^p can be set to arbitrary values in practice. The bias b_j also can be set to an arbitrary value. Therefore, we set all w_{ji}^p and b_j values to 1.0 for convenience. Let us summarize the algorithm in the simulation as follows:

1. Select an image and prepare a new trigger neuron in the T-net.
2. Normalize the decimal number of constructing the image to one.
3. Set the values of w_{ji}^p and b_j to 1.0.
4. Fire the trigger neuron and calculate the output of the neurons in the M-net.
5. Compare the output in the M-net and the teacher image data d_j^p, and learn w_{ji}^p using GDM.
6. Go to step 1.

   Repeat this procedure until the number of images reaches 60 000.

https://doi.org/10.25088/ComplexSystems.27.2.187

Note that we applied step 2 to all images prior to this simulation. Therefore, in effect, we have skipped step 2 here.

In the recalling process, we use equation (1) to fire sequentially from the first trigger neuron and calculate $y_j^p \left(\equiv r_j^p \right)$ using equation (1), where we use r_j^p to denote the output of the recalled image.

In the following we will describe the results. We calculate the following two quantities to check if the recalled image and the corresponding teacher image match. The first is the sum of absolute differences (SAD):

$$\mathrm{SAD}^p = \sum_{j=1}^{M} \left| R_j^p - D_j^p \right|. \tag{9}$$

Here R_j^p and D_j^p are the values obtained by returning r_j^p and d_j^p to a decimal number from 0 to 255, respectively, and

$$R_j^p = 255 \cdot r_j^p, \quad D_j^p = 255 \cdot d_j^p. \tag{10}$$

We can confirm that if SAD^p is close to zero, then there is almost no difference in contrast between a recalled image and the corresponding teacher image as a whole. However, we do not know whether the shape of the recalled handwritten character is consistent with the teacher one. Thus, we introduce a second quantity, the Hamming difference (H). To calculate H, we convert the values of R_j^p and D_j^p to digital data, where if the value is greater than zero, then we set it to one. H can be used to check if the shapes of the handwritten characters of a recalled image and a teacher image match. When the value of H is close to zero, it can be confirmed that the shapes of two handwritten character images are almost the same. We obtain H using the following equation:

$$\mathrm{H}^p = \sum_{j=1}^{n} \left| HR_j^p - HD_j^p \right|, \tag{11}$$

where HR_j^p and HD_j^p denote the converted digital data from R_j^p and D_j^p, respectively. Figure 1 shows averaged SAD^p ($\left\langle \mathrm{SAD}^p \right\rangle_{\Delta T}$) and averaged H^p ($\left\langle \mathrm{H}^p \right\rangle_{\Delta T}$), where these quantities are defined as follows:

$$\begin{aligned} \left\langle \mathrm{SAD}^p \right\rangle_{\Delta T} &= \frac{1}{\Delta T} \sum_{p=\mathrm{n}\Delta T}^{(n+1)\Delta T-1} \mathrm{SAD}^p, \\ \left\langle \mathrm{H}^p \right\rangle_{\Delta T} &= \frac{1}{\Delta T} \sum_{p=n\Delta T}^{(n+1)\Delta T-1} \mathrm{H}^p, \end{aligned} \tag{12}$$

where $n = 0, 1, 2, \ldots, 59$ and $\Delta T = 1000$. The values of $\langle \mathrm{H}^p \rangle_{\Delta T}$ are all zero, as shown in Figure 1, where the values of H^p also are zero for all images. Thus, we judge that the shapes of the handwritten characters of the recalled images and the teacher images are identical. On the other hand, the value of the averaged SAD^p is approximately 70 for every $\langle \mathrm{SAD}^p \rangle_{\Delta T}$. Specifically, the average of $\langle \mathrm{SAD}^p \rangle_{\Delta T}$ for all images is 69.73, where the average is defined as $\sum_{n=0}^{59} \langle \mathrm{SAD}^p \rangle_{\Delta T} / 60$. Therefore, we have checked all elements for differences in values between corresponding images. As a result, we have confirmed that the difference is one for all of these elements. Consequently, we can judge that the recall is almost completely performed, though there is a slight difference in contrast.

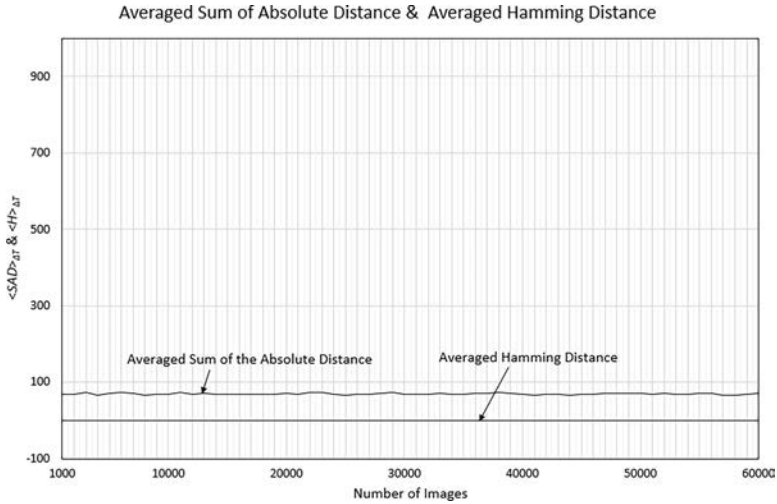

Figure 1. The relation among $\langle \mathrm{SAD}^p \rangle_{\Delta T}$, $\langle \mathrm{H}^p \rangle_{\Delta T}$ and the number of stored images $\langle \mathrm{SAD}^p \rangle_{\Delta T}$ is almost constant, and $\langle \mathrm{H}^p \rangle_{\Delta T}$ is all zeros.

On the other hand, the recalled images and the teacher images are expected to be identical by equation (8). We consider that this results from calculation errors in the computer simulation.

4. Conclusion and Discussion

In this paper, we proposed a hetero-correlation-associative memory model containing two layers in neural networks. As a feature, the number of trigger neurons changes dynamically according to the

number of images to be stored, and one trigger neuron learns one corresponding teacher image. The stored images can be recalled by the firing of the corresponding trigger neurons. The recalled images match the teacher images almost completely. Moreover, this model enables additional learning without destroying previously learned data, because a new trigger neuron is added for each new image; in other words, since the synapse weights are also new, the new trigger neuron has no effect on synaptic weight values (learning) between existing trigger neurons and neurons in the memory net (M-net). Therefore, it can be used to learn images one after another by adding trigger neurons. As a result, the number of stored images becomes very large. Let us calculate the memory rate:

$$\alpha = \frac{\text{the number of stored images}}{\text{total number of neurons}}, \tag{13}$$

where the number of stored images reaches 60 000, and the total number of neurons in the trigger net (T-net) and M-net is 60 784. Thus, the memory rate a is 0.987. The rate approaches one as the number of stored images increases.

In this paper, although one image is linked to one trigger neuron, a trigger neuron can link to many images. For example, let us assume that L sets of M-nets have been prepared. Note that the T-net is still one, as previously. It is possible to learn teacher images displayed simultaneously on an M_1-net, M_2-net, …, M_L-net, for one trigger neuron in the T-net. We can obtain L images for firing one trigger neuron in the recalling process. Furthermore, the data on the M-nets is not limited to images only. For example, it could be that M_1-net learns images, M_2-net learns audio data and M_3-net learns character data—namely, all information that can be patterned can be processed. Thus, if there is mutually related information, the network can learn a group of related data and can recall it.

In this model, if we attempt to select a trigger neuron in the T-net randomly, then the recall of various information passes around on the M-net. Moreover, if the value of x_i^p is small, for example, $x_i^p = 0.1$, then images with low contrast are recalled sequentially. This might be similar to a situation in which we remember various memories one after another. Furthermore, if the stored data can be recalled in response to external data, it would be interesting in the sense that it resembles the recall of memories of experiences we are undergoing. This is currently under investigation.

References

[1] B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," in *1960 IRE Wescon Convention Record*, New York: IRE, 1960 pp. 96–104. Reprinted in *Neurocomputing: Foundations of Research* (J. A. Anderson and E. Rosenfeld, eds.), Cambridge, MA: MIT Press, 1988 pp. 126–134.

[2] R. A. Rescorla and A. R. Wagner, "A Theory of Pavlovian Conditioning: The Effectiveness of Reinforcement and Nonreinforcement," *Classical Conditioning II: Current Research and Theory* (A. H. Black and W. F. Prokasy, eds.), New York: Appleton-Century-Crofts, 1972 pp. 64–69.

[3] D. E. Rumelhart, J. L. McCleland and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: MIT Press, 1986.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, **313**(5786), 2006 pp. 504–507. doi:10.1126/science.1127647.

[5] G. E. Hinton, S. Osindero and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, **18**(7), 2006 pp. 1527–1554. doi:10.1162/neco.2006.18.7.1527.

[6] Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, "Greedy Layer-Wise Training of Deep Networks," in *Advances in Neural Information Processing Systems 19* (B. Schölkopf, J. C. Platt and T. Hofman, eds.), Cambridge, MA: MIT Press, 2006 pp. 153–160.

[7] H. Lee, R. Grosse, R. Ranganath and A. Y. Ng, "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations," in *ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Quebec, Canada, 2009, New York: ACM, 2009 pp. 609–616. doi:10.1145/1553374.1553453.

[8] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS '12 Proceedings of the 25th International Conference on Neural Information Processing Systems, Vol. 1*, Lake Tahoe, NV, 2012, USA: Curran Associates, Inc. 2012 pp. 1097–1105.

[9] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean and A. Y. Ng, "Building High-Level Features Using Large Scale Unsupervised Learning," in *ICML '12 Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, 2012, USA: Omnipress, 2012 pp. 507–514.

[10] K. Fukushima and S. Miyake, "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position," *Pattern Recognition*, **15**(6), 1982 pp. 455–469. doi:10.1016/0031-3203(82)90024-3.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, **1**(4), 1989 pp. 541–551. doi:10.1162/neco.1989.1.4.541.

[12] K. Tsutsumi, "Cross-Coupled Hopfield Nets via Generalized-Delta-Rule-Based Internetworks," in *1990 IJCNN International Joint Conference on Neural Networks*, San Diego, CA, 1990 pp. 259–265. doi:10.1109/IJCNN.1990.137724.

[13] P. Y. Simard, D. Steinkraus and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," in *ICDAR '03 Proceedings of the Seventh International Conference on Document Analysis and Recognition, Vol. 2*, Washington, DC: IEEE Computer Society, 2003 pp. 958–963.

[14] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, **86**(11), 1998 pp. 2278–2324. doi:10.1109/5.726791.

[15] N. Srebro and A. Shraibman, "Rank, Trace-Norm and Max-Norm," *Learning Theory, COLT 2005, Lecture Notes in Computer Science* (P. Auer and R. Meir, eds.), Berlin, Heidelberg: Springer, 2005 pp. 545–560. doi:10.1007/11503415_ 37.

[16] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The Journal of Machine Learning Research*, **15**(1), 2014 pp. 1929–1958.

[17] K. Nakano, "Associatron: A Model of Associative Memory," *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-2**(3), 1972 pp. 380–388. doi:10.1109/TSMC.1972.4309133.

[18] J. A. Anderson, "A Simple Neural Network Generating an Interactive Memory," *Mathematical Biosciences*, **14**(3–4), 1972 pp. 197–220. doi:10.1016/0025-5564(72)90075-2.

[19] T. Kohonen, "Correlation Matrix Memories," *IEEE Transactions on Computers*, **C-21**(4), 1972 pp. 353–359. doi:10.1109/TC.1972.5008975.

[20] S.-I. Amari and K. Maginu, "Statistical Neurodynamics of Associative Memory," *Neural Networks*, **1**(1), 1988 pp. 63–73. doi:10.1016/0893-6080(88)90022-6.

[21] S. Yoshizawa, M. Morita and S.-I. Amari, "Capacity of Associative Memory Using a Nonmonotonic Neuron Model," *Neural Networks*, **6**(2), 1993 pp. 167–176. doi:10.1016/0893-6080(93)90014-N.

[22] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences*, **79**(8), 1982 pp. 2254–2258. doi:10.1073/pnas.79.8.2554.

[23] J. J. Hopfield and D. W. Tank, " 'Neural' Computation of Decisions in Optimization Problems," *Biological Cybernetics*, **52**(3), 1985 pp. 141–152.

[24] D. J. Amit, H. Gutfreund and H. Sompolinsky, "Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks," *Physical Review Letters*, **55**(14), 1985 pp. 1530–1533. doi:10.1103/PhysRevLett.55.1530.

[25] Y. LeCun, C. Cortes and C. J. C. Burges, "The MNIST Database of Handwritten Digits." (May 23, 2018) yann.lecun.com/exdb/mnist.