Visualization of Learning Processes for Back Propagation Neural Network Clustering

Kohei Arai 1
Graduate School of Science and Engineering
Saga University
Saga City, Japan

***Abstract*—Method for visualization of learning processes for back propagation neural network is proposed. The proposed method allows monitor spatial correlations among the nodes as an image and also check a convergence status. The proposed method is attempted to monitor the correlation and check the status for spatially correlated satellite imagery data of AVHRR derived sea surface temperature data. It is found that the proposed method is useful to check the convergence status and also effective to monitor the spatial correlations among the nodes in hidden layer.**

Keywords-neural network; error back propagation; convergence process; spatial correlation

I. INTRODUCTION

Back Propagation Neural Network: BPNN is widely used method for machine learning and optimization method. One of the problems of BPNN is that it cannot ensure to find global optimum solution and can find one of local minima. Also it is difficult to check convergence status; residual error can be monitored though. Method for visualization of convergence processes and spatial correlation of nodes in hidden layer of BPNN is proposed.

II. PROPOSED METHOD

A. Back Propagation Neural Network

The activation is differentiable function of total input, given by equation (1) and (2),

$$y_k^p = \mathcal{F}(s_k^p) \tag{1}$$

$$s_k^p = \sum_j w_{jk} y_j^p + \theta_k \tag{2}$$

where W_{jk} can be known as the weight of the connection from unit *j* to unit *k*. It is convenient to represent the pattern of connectivity in the network by a weight matrix whose elements are the weights W_{ik}. In addition, the unit calculates the activity by using some function of the total weighted input. Typically we use the sigmoid function:

$$y^p = \mathcal{F}(s^p) = \frac{1}{1+e^{-s^p}} \tag{3}$$

The error measure E^p is defined as the total quadratic error for pattern "*p*" at the output units:

$$E^p = \frac{1}{2}\sum_{o=1}^{N_o}(d_o^p - y_o^p)^2 \tag{4}$$

where d_o^p is the desired output for unit "*o*" when pattern "*p*" is clamped. We further set as the sum square error.

$$E = \sum_p E^p \tag{5}$$

The error derivative of the weights is computed to recognize that how the error changes as each weight is increased or decreased slightly. We can write as equation (6),

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial s_k^p}\frac{\partial s_k^p}{\partial w_{jk}} \tag{6}$$

Thus, the error signal for an output unit can be written as equation (7),

$$\delta_o^p = (d_o^p - y_o^p)\,\mathcal{F}_o{}'(s_o^p) \tag{7}$$

Also the error signal for a hidden unit is determined recursively in term of error signals of the units to which it directly connects and the weights of those connections which is shown in equation (8),

$$\delta_h^p = \mathcal{F}'(s_h^p)\sum_{o=1}^{N_o}\delta_o^p w_{ho} \tag{8}$$

Therefore, the weights could be modified by using the past weights as follows,

$$\Delta w_{jk}(t+1) = \gamma \delta_k^p y_j^p + \alpha \Delta w_{jk}(t) \tag{9}$$

The back propagation process can be explained as: when a set of desired input data and desired output data are ready, spare weights matrix which represent for the connection between “input layer and hidden layer” or “hidden layer and output layer”, will be created in random real numbers. While these matrixes are creating, the desired weights are also given define values.

When the preparation is finished, we will start learning process. In the beginning, the all actual input data will be given to equation (2), these results can be called the *nets*. The *nets* now can be used to figure out the actual output of the input nodes or actual input of hidden nodes through sigmoid equation (3) which is mentioned before. Equation (4) is useful for computing the error E^p which is the different between the actual output and the desired output. Follow the function is presented before; the error signal is able to calculate with equation (7) and (8). Thus then we have enough data to move to the next step which the weights can be changed by using equation (9).

The total error is able to compute by equation (5) above during the learning process. After the weights in neural network are changed, the total error will be compared with the value which is decided before. Therefore, if the total error is acceptable, the training process can be stopped. However, almost all the neural network can not finish its learning process after only several times. Thus, maybe it requires that we should train the network with a big enough number of times, and then the total error might be small enough to be acceptable.

B. Proposed All Node Linked Neural Network: ANLNN

In back propagation algorithms, an important consideration is the learning rate. If the learning rate is too small, it will take a long time to converge. However, when the learning rate is too large, we may end up bouncing around the error surface out of control, the algorithms diverges. This often ends with an overflow error.

Beside the learning rate, momentum also plays an important role in back propagation process. Adding the momentum, is one of the ways to avoid oscillation at large learning process and when no momentum term is used, it properly takes a long time before the global minimum has been reached with a low learning rate. Moreover, the number of hidden units is one of the reasons which cause effect on the network.

A large number of hidden units lead to a small error on the training set. However, even if increasing the number of hidden units can help the neural network escape from a trap at local minimum but it will not guarantee that neural network again can find the global minimum, when number of hidden units is over the demand. As the network trains, the weights can be adjusted to very large values. The total input of a hidden unit or output unit can therefore reach very high (positive or negative either) values and because of the activation function which we use in this research, is sigmoid function, the unit will have an activation very close to zero or very close to one. Thus, in the back-ward stage the error signal will be close to zero and the learning process can come to a virtual standstill. The answer for this problem is that the suitable momentum maybe select in order to support the process becomes normal.

We also use the other neural network which has a structure different from a typical one, with the purpose is able to clearly recognize the trend following weight’s images. A new structure is described as: normally in typical structure neural network, all the nodes from one layer will completely link to all the other nodes of connected layer. The neural network which will be call All Nodes Linked Neural Network: ANLNN hereafter, does not have the same structure like a typical one, each node on a layer will only link to a specify node or specify nodes on connected layer. The nodes of a layer on the ANLNN is sorted like a matrix, from this matrix a smaller matrix will be picked up and linked to other specify nodes on the other layer whose nodes are also sorted in the same way.

The ANLNN and typical structure neural network will be tried to apply in recognizing integer numbers. Firstly, the desired input and desired output are selected from a set of integer numbers. The same problem happens like before, when the data is fed into neural network, nothing is done, neural network becomes virtual standstill. Thus, the data obviously has to normalize before it is given to a network. We know that all the integer numbers are 2 bytes digits, from this indication we can normalize the data by the same way which is mentioned above, like when we carried out the experiments with Multi Channel Sea Surface Temperature; MCSST data. All of the integer numbers will be divided into 1023 before they are brought to the neural network. The weight’s values are adjusted after each step is taken during the training process. Using these results, an image which displays how different the new weight’s values are, can be drawn. The initial values are absolutely free to choose between [-1, +1].

As we know, the highly correlated data which can be referred to AVHRR MCSST data or a set of integer numbers in turn from “0” until “9”, might be used in order to lead the weight’s images to the same trend if a global minimum is found when we keep changing the initial weight’s values for several times. Therefore, we can work out whether neural network converges at the local minimum or not by comparing the characteristic of the image.

We also use the other neural network which has a structure different from a typical one, with the purpose is able to clearly recognize the trend following weight’s images. A new structure is described as: normally in typical structure neural network, all the nodes from one layer will completely link to all the other nodes of connected layer. The neural network which will be call ANLNN neural network from now, does not have the same structure like a typical one, each node on a layer will only link to a specify node or specify nodes on connected layer. The nodes of a layer on the SL neural network are sorted like a matrix, from this matrix a smaller matrix will be picked up and linked to other specify nodes on the other layer whose nodes are also sorted in the same way.

The ANLNN neural network and typical structure neural network will be tried to apply in recognizing integer numbers. Firstly, the desired input and desired output are selected from a set of integer numbers. The same problem happens like before, when the data is fed into neural network, nothing is done, neural network becomes virtual standstill. Thus, the data obviously has to normalize before it is given to a network. We know that all the integer numbers are 2 bytes digits, from this indication we can normalize the data by the same way which is mentioned above, like when we carried out the experiments with MCSST data. All of the integer numbers will be divided into 1023 before they are brought to the neural network.

C. Visualization of Weighting Coefficients as an Image for ANLNN

Layered structure of the proposed ANLNN is shown in Figure 1. Because all the input nodes are linked to the hidden layer nodes and the hidden layer nodes are linked to the output layer nodes, the number of weighting coefficients is n^2 where n is the number of input layer nodes as well as the number of output layer nodes. Then weighting coefficients between input and hidden layers and those between hidden and output layers can be displayed as an image. AVHRR band 4 imagery data is assumed to be input data of the ANLNN while MCSST of imagery data is also assumed to be desired output for the output layer. Then learning process begins.

Although it is possible to take a look at residual error, it is still difficult to decide convergence situations. Because BPNN utilizes steepest descent method for learning process, it is not possible to reach a global optimum for weighting coefficients. Because the aforementioned reason, input data of neighboring nodes are highly correlated and output data of neighboring nodes are also highly correlated. That is same thing for weighting coefficients image. The weighting coefficients between input and hidden layers of neighboring nodes are highly correlated while the weighting coefficients between hidden and output layers of neighboring nodes are also highly correlated. Therefore, if we take a look at correlation among weighting coefficients, then it is possible to decide the convergence status.

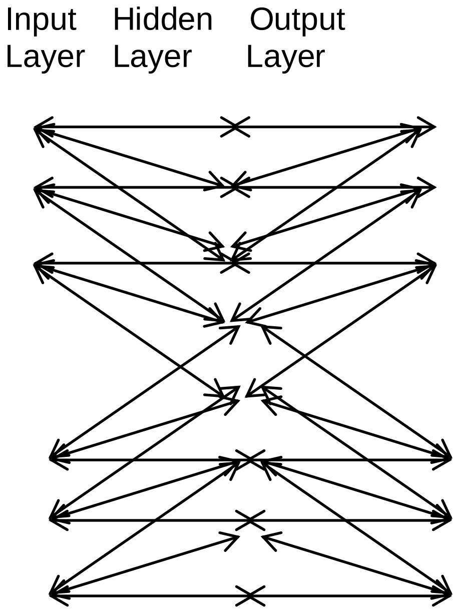


Fig. 1. Layered structure of the proposed ANLNN

III. EXPERIMENTS

A. Data Used

The AVHRR MCSST data is given to the process in order to figure out the essence of neural network which is acquired the back propagation algorithms. The data from band 4 and corresponding MCSST data are used this time.

MCSST can be estimated through regressive analysis with Advanced Very High Resolution Radiometer: AVHRR of Band 4 and 5 of thermal infrared images.

$$MCSST=a\ DN4 + b\ DN5 + c \qquad (1)$$

where MCSST, DN4 and DN5 denotes SST, Digital Number: DN of Band 4 and 5. Also a, b and c are regressive coefficients. If training samples, sets of MCSST, DN4 and DN5 are available, then a, b and c can be estimated. After that SST can be estimated with the regressive equation for another DN4 and DN5. Figure 2 and 3 show an example of AVHRR Band 4 and 5 and estimated MCSST images.

(a)AVHRR Band

(b) Band 5

Fig. 2. An example of AVHRR Band 4 and 5 images of Kyushu which are acquired on 20 May 2002

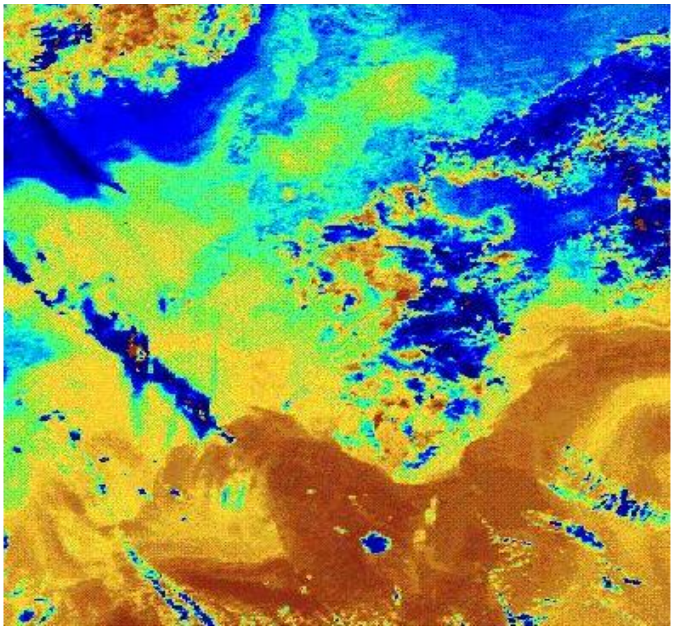


Fig. 3. SST estimated through a regressive analysis with sets of SST, DN4 and DN5 of training data

B. Node Images Derived from the Proposed Method

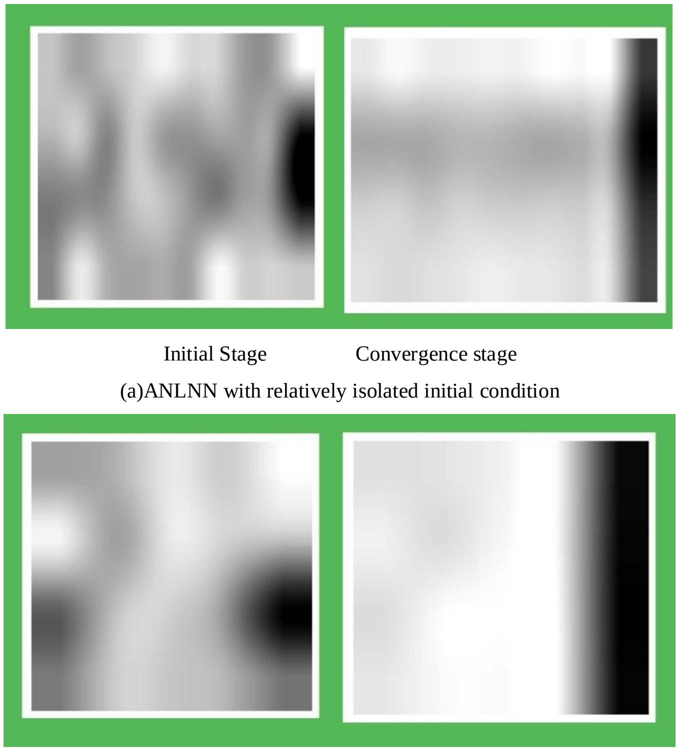
In the beginning, the experiment showed that there is no remarkable change in average error's value or weight's values either. The neural network becomes virtual standstill. After checking carefully all the steps, we realize that because the value of AVHRR MCSST data is quite big, thus it is necessary to normalize the data otherwise nothing will be done because almost all the results which are an output of hidden nodes or output nodes, error signal, will close to 0 or close to 1. As we know MCSST data is 10 bit data, therefore when we prepare a set of desired input and desired output, MCSST data can be normalized by dividing each MCSST data into 1023 which is means that totally there are 1024 elements, from 0 until 1023

We start to display weight images in order to confirm whether the initial image is different from the result image after learning process or not. As the images are shown below, we see that the initial image does not have anything special, it does not show any characteristic but the result image does. It is not difficult to recognize that through the training process, weight's values are adjusted and they are modified with a remarkable amount because of shape and also the dark, bright color on the result image.

C. Experimental Resuts

Figure 4 shows weight coefficients image for between input and hidden layers at initial stage and convergence stages. Because the number of input nodes is 3 by 3 of AVHRR band 4 of data, the number of weighting coefficients is 81 by 81. Therefore, the weighting coefficients image of Figure 4 consists of 81 by 81, accordingly.

At the initial stage, all the weighting coefficients are determined with the random numbers derived from Mersenne Twister random number generator. This is referred as comparatively isolated weighting coefficients. By using averaging filter, relatively correlated initial conditions of weighting coefficients are also determined. There is an initial condition dependency for convergence processes. That is because of preparation of two sets of initial weighting coefficients, relatively isolated and comparatively correlated initial conditions. The weight's initial values of this network are in turn given -1, 0, 1 and random numbers while a set of desired input, output data also is tried with integer numbers and MCSST data. We suppose that if the network converges at the global minimum, all the weight's images will show the same trend even the initial images or initial weight's values are different. In addition, the momentum plays an important role in effecting weight's values while desired input and desired output also decide the value of average error, the speed of neural network when it converges. These doubtful questions are proved when an experiment above is conducted.



Initial Stage Convergence stage

(a)ANLNN with relatively isolated initial condition

Initial Stage Convergence stage

(b)ANLNN with comparatively correlated initial condition

Fig. 4. Weight coefficients image of ANLNN with the different initial conditions for weighting coefficients between input and hidden layers for 9 input nodes

Figure 5 shows example of convergence processes with residual error while Figure 6 shows the average correlation coefficient for all the weighting coefficients which situate in between input and hidden layers. In accordance with increasing of the iteration number, residual error goes down together with increasing of the averaged correlation coefficient. This is because there are high correlations among the input nodes. Therefore, it is possible to determine convergence of learning processes with referencing the averaged correlation coefficient rather than referring to the residual error. Furthermore, it is also possible to monitor convergence processes by looking at the weighting coefficients image.

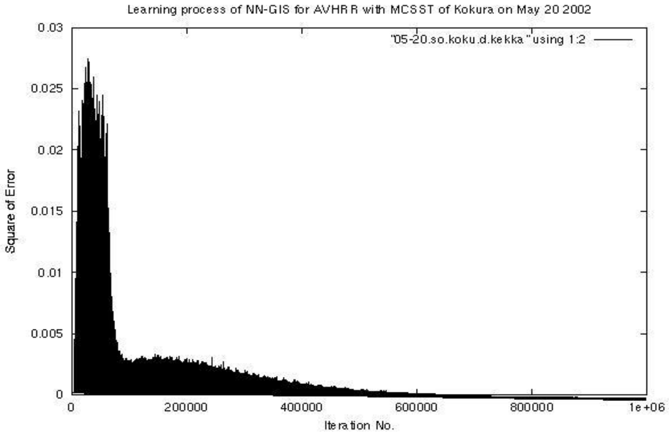


Fig. 5. Residual errors in convergence process of the ANLNN

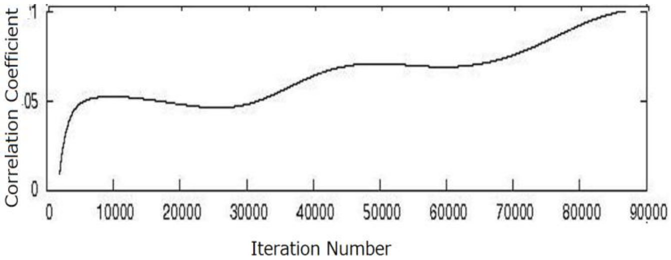


Fig. 6. Averaged correlation coefficient among the weighting coefficients between input and hidden layer.

IV. CONCLUSION

Method for visualization of learning processes for back propagation neural network is proposed. The proposed method allows monitor spatial correlations among the nodes as an image and also check a convergence status. The proposed method is attempted to monitor the correlation and check the status for spatially correlated satellite imagery data of AVHRR derived sea surface temperature data. It is found that the proposed method is useful to check the convergence status and also effective to monitor the spatial correlations among the nodes in hidden layer

ACKNOWLEDGMENT

The author would like to thank Mr. Son Nguyen Xuan for his effort to conduct experiments.

REFERENCES

[1] Ben Krose, Patric Van Der Smagt, An introduction to neural network — Eighth edition, November 1996.

[2] Colin Fyfe, Artificial neural network — Department of computing and information systems, the University of Paisley, Edition 1.1, 1996.

[3] Dave Anderson and George McNeill, Kaman, Artificial neural network technology – A Dacs state of the air report, August 20, 1992 - Sciences Corporation.

[4]

[5] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[6] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[7] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[8] K. Elissa, "Title of paper if known," unpublished.

[9] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[10] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[11] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

AUTHORS PROFILE

Kohei Arai, He received BS, MS and PhD degrees in 1972, 1974 and 1982, respectively. He was with The Institute for Industrial Science and Technology of the University of Tokyo from April 1974 to December 1978 also was with National Space Development Agency of Japan from January, 1979 to March, 1990. During from 1985 to 1987, he was with Canada Centre for Remote Sensing as a Post Doctoral Fellow of National Science and Engineering Research Council of Canada. He moved to Saga University as a Professor in Department of Information Science on April 1990. He was a councilor for the Aeronautics and Space related to the Technology Committee of the Ministry of Science and Technology during from 1998 to 2000. He was a councilor of Saga University for 2002 and 2003. He also was an executive councilor for the Remote Sensing Society of Japan for 2003 to 2005. He is an Adjunct Professor of University of Arizona, USA since 1998. He also is Vice Chairman of the Commission "A" of ICSU/COSPAR since 2008. He wrote 30 books and published 322 journal papers.