International JOURNAL OF CONTENTS Vol.21, No.1 March 2025

[2025. 3]

Integrating Error Back-Propagation and Independent Component Analysis Algorithms to Enhance Neural Network Performance

Sang-Hoon Oh [1,*]

[1]Mokwon University; Professor; ohsanghoon16@gmail.com

The Korea Contents Association

ISSN:1738-6764, eISSN:2093-7504

Integrating Error Back-Propagation and Independent Component Analysis Algorithms to Enhance Neural Network Performance

Sang-Hoon Oh [1,*]

[1] Mokwon University; Professor; ohsanghoon16@gmail.com
* Correspondence

https://doi.org/10.5392/IJoC.2025.21.1.147

Manuscript Received 4 December 2024; Received 14 January 2025; Accepted 14 January 2025

Abstract: *The EBP (Error Back-Propagation) Algorithm was initially proposed for training MLP's (Multi-Layer Perceptrons) and is now widely used for training deep neural networks. This supervised learning algorithm minimizes the error function between the actual output values of MLP's and their desired values. However, ICA (Independent Component Analysis) is an unsupervised learning algorithm that aims to maximize the independence among the outputs of neural networks. ICA has been shown to realize visual features in the V1 layer of the human brain by learning from natural scenes and cochlear features of the human ear by learning from auditory signals. In this paper, we propose merging the supervised EBP algorithm with the unsupervised ICA algorithm to enhance the performance of neural networks by training independent features in the initial learning stage. This approach mirrors the feature-learning process observed in mammals during the early stages of life. Furthermore, the proposed approach is verified through simulations on isolated-word recognition tasks, achieving improved classification performance with faster learning convergence. In detail, when the number of hidden nodes is 100, EBP with ICA reaches a misclassification ratio of 2.78% on the test data at 160 epochs, while EBP achieves 3.28% at 300 epochs.*

Keywords: Merge of Supervised and Unsupervised Learning; Error Back-propagation; Independent Component Analysis; Neural Networks

1. Introduction

The MLP (Multi-Layer perceptron) was a breakthrough in neural networks for training nonlinearly separable problems, and the EBP (Error Back-Propagation) algorithm is used to train MLP's [1]. MLP's are trained to minimize the error function between the actual output values of MLP's and their desired values. EBP algorithm is now widely used for training deep neural networks [2]. An MLP consists of an input layer, one or more hidden layers, and an output layer [1]. Input values are presented to the input layer, while desired values are presented to the output layer. Each hidden node, particularly that with sigmoidal activation function, acts as a soft threshold function, creating a hyperplane in the input space. These hyperplanes are determined by the EBP algorithm during the process of minimizing the error function and can be interpreted as a form of feature extraction. Thus, the hidden layers play a crucial role in feature extraction [3, 4].

Although there is theoretical proof that MLPs with enough hidden nodes are universal approximators of any function [5], significant challenges persist in achieving satisfactory training performance when applying MLP's to real-world problems. Numerous approaches have been proposed to improve the performance of MLP's. One such approach involves modifying the error or objective functions [6-8]. Liano proposed the MLS (mean-log square) error function to suppress the excessive weight updates caused by outliers of training data [9]. The binary CE (cross-entropy) error function can accelerate the learning convergence of neural network classifiers by mitigating the incorrect saturation of output nodes [10]. Additionally, the n-th order extension of the binary CE error achieves better performance by preventing overspecialization to training samples [11-13].

Another notable attempt is the CFM (classification figure of merit) objective function, proposed by Hampshire and Waibel, which is less prone to overlearning compared to its CE counterpart [14].

Another approach involves modifying the architecture of MLP's, including hidden node pruning or increasing output nodes of MLP's. Having too many hidden nodes can lead to better performance on training samples but degrades generalization performance on test samples. To address this, many attempts have been made to prune the number of hidden nodes, aiming to improve generalization performance on unseen samples [15-18]. In classification applications, one output node is typically allocated to each class. However, increasing the number of output nodes per class has also been proposed as a method to improve the performance of MLP's [19]. This strategy acts as an ensemble of many MLP classifiers.

In contrast to the approaches mentioned above, this paper focuses on the role of hidden nodes during the training of MLP's [3]. Mammals learn visual or audio features in the early stages of life. Notably, in the first few weeks after birth, neurons in the visual cortex of kittens become properly connected and develop through exposure to visual stimuli [20]. Inspired by this biological process, we adopt a similar strategy to train the hidden layers of MLP's using unsupervised learning. Based on the extracted features, the MLP's are further trained using supervised learning to minimize the error function between actual output values and their desired values. Among unsupervised learning algorithms [21], the ICA (Independent Component Analysis) algorithm is known to realize visual features of simple cells in the V1 layer of the human brain by learning from natural scenes [22, 23] and cochlear features of the human ear by learning from auditory signals [24]. We adopt the ICA algorithm to train hidden nodes of MLP's for feature extraction. In this paper, we propose integrating the supervised EBP algorithm, which minimizes error functions, with the unsupervised ICA algorithm, which extracts independent features, to enhance the performance of MLP's. This approach mirrors the stages of learning in mammals.

In section 2, we propose integrating the supervised EBP algorithm for minimizing error functions with the ICA algorithm for extracting independent features. In section 3, the effectiveness of the proposed method is demonstrated through simulations on an isolated-word recognition problem. Finally, section 4 concludes this paper.

2. Integration of Error Back-Propagation and Independent Component Analysis Algorithms

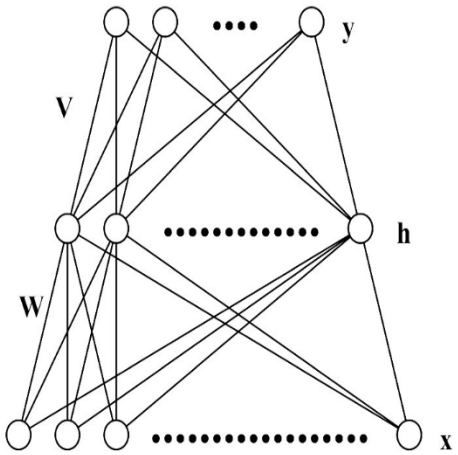

Fig 1. The architecture of a multilayer perceptron

Let's assume an MLP has N inputs, H hidden nodes, and M output nodes, denoted as an "N-H-M MLP". Figure 1 illustrates the architecture of MLP. When an input data $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^T$ is fed into the MLP, the jth hidden node is given by

$$h_j = \tanh(\hat{h}_j) = \tanh((w_{j0} + \sum_{i=1}^{N} w_{ji} x_i) / 2), \quad j = 1, 2, \ldots, H. \tag{1}$$

Here, w_{ji} denotes the weight connecting x_i to h_j, w_{j0} is a bias of h_j, and $\hat{h}_j$ is the net-input or the weighted sum to h_j. The kth output node is

$$y_k = \tanh(\hat{y}_k / 2), \quad k = 1, 2, \ldots, M, \tag{2}$$

where

$$\hat{y}_k = v_{k0} + \sum_{j=1}^{H} v_{kj} h_j \ . \tag{3}$$

Also, v_{k0} is a bias of y_k and v_{kj} denotes the weight connecting h_j to y_k.

Let the desired output vector corresponding to the training sample $\boldsymbol{x}$ be $\boldsymbol{t} = [t_1, t_2, \dots, t_M]^T$, which is coded as follows:

$$t_k = \begin{cases} +1 & \text{if } \mathbf{x} \text{ originates from class } k, \\ -1 & \text{otherwise.} \end{cases} \tag{4}$$

As a distance measure between the actual and desired outputs, we usually use the mean-squared error (MSE) function defined by [1]

$$E = \sum_{k=1}^{M} \frac{1}{2} \left(t_k - y_k \right)^2 . \tag{5}$$

To minimize E for all training data, weights v_{kj}'s are iteratively updated by

$$\Delta v_{kj} = -\eta \frac{\partial E}{\partial v_{kj}} = \eta \delta_k^{(\text{output})} h_j , \tag{6}$$

where

$$\delta_k^{(\text{output})} = -\frac{\partial E}{\partial \hat{y}_k} = \left(t_k - y_k \right) \frac{\left(1 - y_k\right)\left(1 + y_k\right)}{2} \tag{7}$$

is the error signal and η is the learning rate. Also, weights w_{ji}'s are updated by

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} = \eta x_i \, \delta_j^{(\text{hidden})} = \eta x_i \, \frac{(1 - h_j)(1 + h_j)}{2} \sum_{k=1}^{M} v_{kj} \delta_k^{(\text{output})} \ . \tag{8}$$

The weight-update procedure described above corresponds to the EBP algorithm [1]. However, when minimizing the MSE, the EBP algorithm often performs poorly due to the incorrection saturation of output nodes [25] and the overspecialization to test data [9]. To address these issues, we adopt the n-th order extension of the binary CE (nCE) error as an alternative to MSE for the training criterion for MLP's [11]. The nCE error function is defined by

$$E_{nCE} = -\sum_{k=1}^{M} \int \frac{t_k^{n+1} (t_k - y_k)^n}{2^{n-2} (1 - y_k)(1 + y_k)} dy_k \tag{9}$$

and the error signal is

$$\delta_k^{(\text{output})} = -\frac{\partial E_{nCE}}{\partial \hat{y}_k} = \frac{t_k^{n+1} \left(t_k - y_k \right)^n}{2^{n-1}} . \tag{10}$$

When MLP's are trained to minimize the error function, each hidden node with a sigmoidal activation function forms a hyperplane in the input space, acting as a soft threshold function. This behavior can be interpreted as a form of feature extraction [3, 4]. We focus on the role of hidden nodes during the training of MLP's [3]. Mammals learn visual or audio features in the early stages of life [20]. Inspired by this biological process, we adopt a similar strategy to train the hidden layers of MLP's in the initial learning stage. It is well known that the visual features of simple cells in the V1 layer of the human brain are realized by the ICA learning from natural scenes [22, 23] and cochlear features of the human ear by the ICA learning from auditory signals [24]. Drawing from these insights, we employ the ICA algorithm to train hidden nodes of MLP's for feature extraction.

The ICA algorithm aims to learn independent signals through the linear transformation of sensor signals. Let us assume that there are source signal vectors $\boldsymbol{s} = [s_1, s_2, \dots, s_N]^T$ where $s_i (i = 1,2, \dots, N)$ are statistically independent signals with non-Gaussian, or at most, one Gaussian signal. We measure a sensor signal vector $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$, where $\boldsymbol{A}$ is the $n \times n$ mixing matrix, and aim to recover the source signals from the learning a linear transformation $\boldsymbol{u} = \boldsymbol{W}\boldsymbol{x}$ where $\boldsymbol{W}$ is the $n \times n$ unmixing matrix. Importantly, there is no prior information about the mixing matrix or source signals.

To find the independent source signals, we do the element-wise transform $y_i = g(u_i)$ where $g(.)$ is the cumulative distribution function of source signal. The entropy $\boldsymbol{H}(\boldsymbol{y})$ is defined by

$$H(\mathbf{y}) = -E\left[\log p(\mathbf{y})\right] \tag{11}$$

where $p(\boldsymbol{y})$ is the joint probability density function (p.d.f.) of $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T$ and $E[.]$ is the expectation operator. To maximize $\boldsymbol{H}(\boldsymbol{y})$ for extracting independent components, the unmixing matrix is updated by

$$\Delta\mathbf{W} \propto \frac{\partial H(\mathbf{y})}{\partial \mathbf{W}} = \left[\left(\mathbf{W}^T\right)^{-1} - \varphi(\mathbf{u})\mathbf{x}^T\right], \tag{12}$$

where

$$\varphi(\mathbf{u}) = -\frac{\frac{\partial p(\mathbf{u})}{\partial \mathbf{u}}}{p(\mathbf{u})} \tag{13}$$

is the score function and $p(\boldsymbol{u})$ is the joint p.d.f. of $\boldsymbol{u} = [u_1, u_2, \ldots, u_N]^T$ [19]. Furthermore, Amari proposed the more efficient "Natural Gradient [26]" defined by

$$\Delta\mathbf{W} \propto \frac{\partial H(\mathbf{y})}{\partial \mathbf{W}}\mathbf{W}^T\mathbf{W} = \left[\mathbf{I} - \varphi(\mathbf{u})\mathbf{u}^T\right]\mathbf{W}. \tag{14}$$

However, the ICA algorithm described in equations (12) and (14) assumes that the number of source signals equals the number of sensor signals and $\boldsymbol{W}$ is the square matrix. Typically, MLPs have fewer hidden nodes than input nodes, meaning that the weight matrix $\boldsymbol{W} = [w_{ji}]_{H\times N}$ of the hidden layer is not a square matrix. In contrast, Girolami et al. [27] proposed a method involving the linear projection of data onto a lower-dimensional subspace to uncover the underlying structure of observations independent latent causes. This method provides a generalized neural framework for Independent Component Analysis. Accordingly, the ICA algorithm with the non-square unmixing matrix is given by

$$\Delta\mathbf{W} \propto \left[\mathbf{I} + f(\mathbf{u})\mathbf{u}^T\right]\mathbf{W}, \tag{15}$$

where

$$f(u_k) = -u_k - K_k \tanh(u_k) \tag{16}$$

and

$$K_k = \text{sign}\left(E\left[\text{sech}^2(u_k)\right]E\left[u_k^2\right] - E\left[\tanh(u_k)u_k\right]\right). \tag{17}$$

During the initial weeks after birth, neurons in the visual cortex of kittens establish proper connections and undergo development through exposure to visual stimuli [20]. Inspired by the biological process of learning features in the early stages of mammalian life, we adopt the ICA algorithm to learn independent features in the initial stage of EBP learning. Consequently, we integrate the ICA unsupervised learning with the EBP supervised learning for hidden nodes, while output weights are updated only by the EBP learning. Using matrix notation, the hidden weight matrix is updated by

$$\Delta\mathbf{W} = \eta_{EBP}\boldsymbol{\delta}^{(\text{hidden})}\mathbf{x}^T + \eta_{ICA}\left[\mathbf{I} + f\left(\hat{\mathbf{h}}\right)\hat{\mathbf{h}}^T\right]\mathbf{W} \tag{18}$$

where $\boldsymbol{\delta}^{(hidden)} = [\delta_1^{(hidden)}, \delta_2^{(hidden)}, \ldots, \delta_H^{(hidden)}]^T$ and $\hat{\boldsymbol{h}}=[\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_H]^T$. η_{EBP} and η_{ICA} are learning rates for EBP and ICA algorithms, respectively. We can control the initial stage of learning independent features using η_{ICA}.

For better understanding of the proposed algorithm of MLPs, we summarize the on-line learning procedure as follows:

1. Initialize MLPs with *N* inputs, *H* hidden nodes and *M* output nodes.
2. Present an input sample to an MLP and calculate output node values according to (1) and (2).
3. Calculate the error signal of output node according to (10)
4. Estimate the updating amounts of output weights according to (6)
5. Calculate the error signal of hidden nodes $\delta_j^{(\text{hidden})} = \frac{(1-h_j)(1+h_j)}{2}\sum_{k=1}^{M} v_{kj}\delta_k^{(\text{output})}$
6. Estimate the updating amounts of hidden weights according to (18)
7. Update output and hidden weights

3. Simulations

To verify the effectiveness of the proposed learning strategy, which integrates the ICA algorithm for training hidden nodes to extract independent features with the EBP algorithm for minimizing the error function in input data classification, we conducted experiments using MLP's on an isolated-word recognition problem. The vocabulary consists of 50 words, each spoken twice by nine speakers. A total of 900 speech samples were used to train the MLPs after extracting the 1024-dimensional ZCPA (zero-crossing peak amplitude) components [28]. An additional 1,050 speech samples, spoken three times by seven different speakers, were used as test data to measure the generalization performance of the MLP's on unseen data.

The MLP's consist of 1024 input nodes and 50 output nodes, with various configurations of hidden nodes (50, 60, 80, 100). Initial weights were randomly selected using a uniform distribution on $[-1 \times 10^{-4}, 1 \times 10^{-4}]$. To mitigate issues such as the incorrect saturation of output nodes and the overspecialization to test data, we employed the nCE error function with $n = 6$ as the learning objective function of the EBP algorithm [9]. The learning rate for the EBP algorithm was $\eta_{EBP} = 0.07$, and the learning rate for the ICA algorithm was $\eta_{ICA} = 0.0001$. As described above, we set η_{ICA} to zero after learning epoch surpassed 100 to regulate the ICA algorithm, ensuring the extraction of independent features occurred primarily during the initial stages of learning.

Nine simulations were conducted using different initializations of MLP's, and the average results are presented in Figure 2. From Figure 2(a), we observe that the proposed learning method demonstrates faster convergence and better performance on training samples compared to the EBP with nCE error function. More importantly, as shown in Figure 2(b), the proposed method achieves superior generalization performance on test data. Also, increasing the number of hidden nodes leads to significantly better performance on test data. As expected, this improvement occurs because the proposed method tries to extract independent components among hidden nodes, thereby reducing redundancy among hidden nodes. Let us compare EBP and EBP with ICA in more detail. When the number of hidden nodes is 100, EBP with ICA achieves perfect classification on the training data at 110 learning epochs, whereas EBP requires 200 epochs. Additionally, EBP with ICA reaches a misclassification ratio of 2.78% on the test data at 160 epochs, while EBP achieves 3.28% at 300 epochs.

Integrating supervised EBP learning with unsupervised ICA learning has successfully enhanced the performance of MLP's by generating more training samples [29, 30]. Accordingly, inspired by the biological process of early learning stages in life, this paper highlights the success of improving MLP performance through learning independent features and reducing redundancies among hidden nodes. Unsupervised learning can complement supervised learning in various ways. Integrating the ICA algorithm for training output nodes can be another approach to improve performances of neural networks.

4. Conclusions

Supervised learning is an algorithm designed to minimize the error functions between the actual output values of artificial neural networks and their desired values. In contrast, unsupervised learning is an algorithm that projects input samples onto another space without the need for desired values. Interestingly, mammals learn visual and auditory features during the early learning stages of development. Inspired by this biological process, we integrated the supervised EBP learning with the unsupervised ICA learning to enhance the classification performance of MLP's.

In the proposed method, MLP's were trained by the EBP algorithm with nCE objective function to prevent incorrect saturation of output nodes and overspecialization to training samples. Additionally, during the initial learning epochs, the hidden nodes were trained by the ICA algorithm to extract independent components among the nodes. Simulations conducted on an isolated-word recognition problem showed that the proposed method achieved superior performance with faster learning convergence compared to the EBP algorithm alone. Furthermore, the proposed method demonstrated significantly improved performance on test data when the number of hidden nodes was increased because of reducing redundancy among hidden nodes. These results suggest that integrating ICA with EBP enhances the role of hidden nodes by enabling them to extract independent features while reducing redundancy.

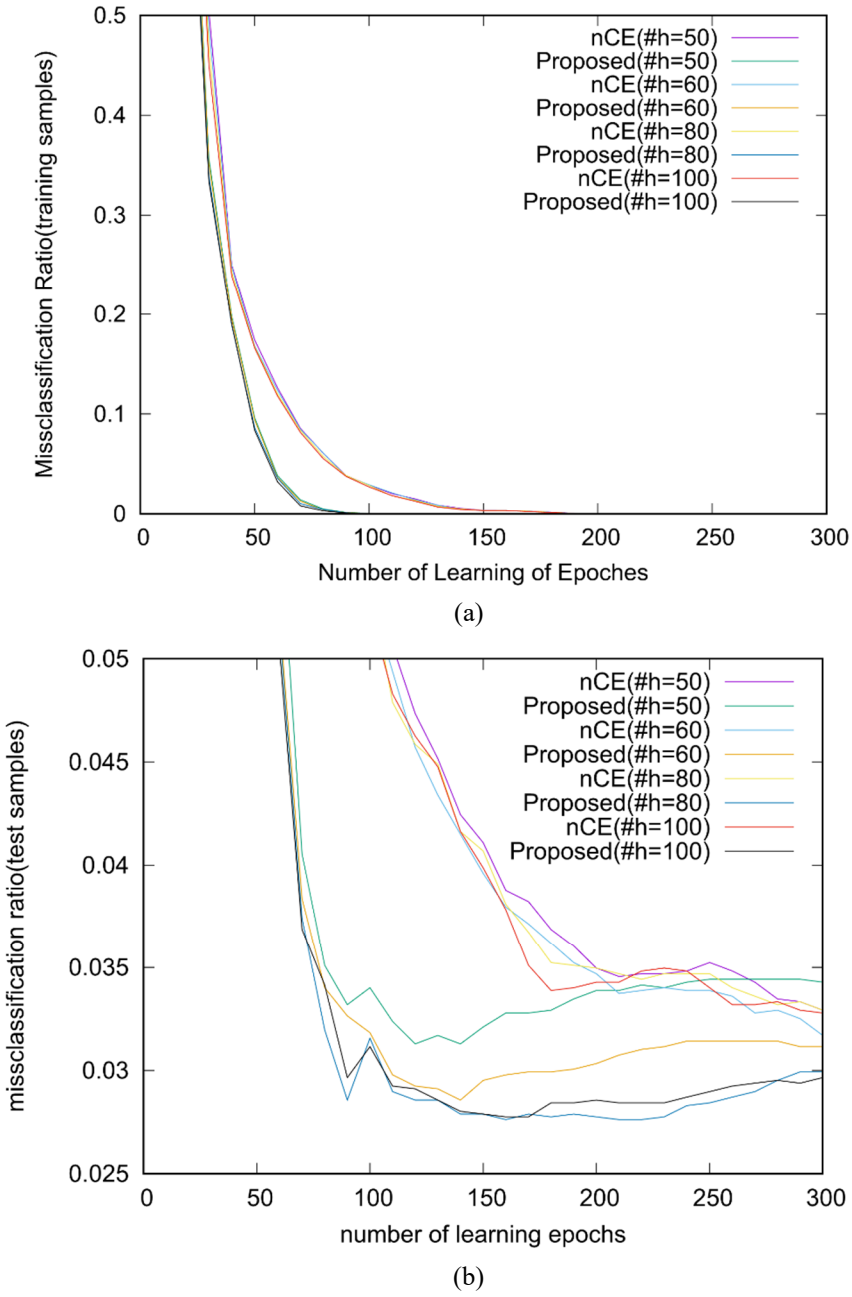

Figure 2. Simulation results of MLP's with isolated-word recognition task. "nCE" denotes the EBP learning with nCE(n=6) error function and "Proposed" denotes the proposed learning algorithm. Also, "#h" is the number of hidden nodes. (a) the misclassification ratio for the training samples; (b) the misclassification ratio for the test samples

Conflicts of Interest: The author declares no conflict of interest.

References

[1] D. E. Rumelhart and J. L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, MA, 1986, doi: https://doi.org/10.7551/mitpress/5236.001.0001.

[2] Y. LeCun, LeNet-5, Convolutional Neural Networks, 1998. [Online] Available: http://yann.lecun.com/exdb/lenet/

[3] J. V. Shah and C. S. Poon, "Linear Independence of Internal Representations in Multilayer Perceptrons," IEEE Tans. Neural Networks, vol. 10, no. 1, pp. 10-18, Jan. 1999, doi: https://doi.org/10.1109/72.737489

[4] S. H. Oh and Y. Lee, "Effect of Nonlinear Transformations on Correlation Between Weighted Sums in Multilayer Perceptrons," IEEE Trans. Neural Networks, vol. 5, no. 3, pp. 508-510, May. 1994, doi: https://doi.org/10.1109/72.286927.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," Neural Networks, vol. 2, pp. 359-366, 1989, doi: https://doi.org/10.1016/0893-6080(89)90020-8.
[6] S. H. Oh, "Statistical Analyses of Various Error Functions for Pattern Classifiers," Proc. Convergence on Hybrid Information Technology, Daejon, Korea, vol. 206, pp. 129-133, Sep. 22-24, 2011, doi: https://doi.org/10.1007/978-3-642-24106-2_17.
[7] S. H. Oh, "Contour Plots of Objective Functions for Feed-Forward Neural Networks," Int. J. of Contents, vol. 8, no. 4, pp. 30-35, Dec. 2012, doi: https://doi.org/10.5392/IJoC.2012.8.4.030.
[8] S. H. Oh and H. Wakuya, "Comparison of Objective Functions for Feed-Forward Neural Network Classifiers Using Receiver Operating Characteristics Graph," Int. J. of Contents, vol. 10, no. 1, pp. 23-28, Mar. 2014, doi: https://doi.org/10.5392/IJoC.2014.10.1.023.
[9] K. Liano, "Robust Error Measure for Supervised Neural Network Learning with Outliers," IEEE Trans. Neural Networks, vol. 7, pp. 246-250, Jan. 1996, doi: https://doi.org/10.1109/72.478411.
[10] A. van Ooyen and B. Nienhuis, "Improving the Convergence of the Backpropagation Algorithm," Neural Networks, vol. 4, pp. 465-471, 1992, doi: https://doi.org/10.1016/0893-6080(92)90008-7.
[11] S. H. Oh, "Improving the error back-propagation algorithm with a modified error function," IEEE Trans. Neural Networks, vol. 8, pp. 799-803, May. 1997, doi: https://doi.org/10.1109/72.572117.
[12] S. H. Oh, "Error Back-Propagation Algorithm for Classification of Imbalanced Data," Neurocomputing, vol. 74, pp. 1058-1061, 2011, doi: https://doi.org/10.1016/j.neucom.2010.11.024.
[13] S. H. Oh, "Improving the Error Back-Propagation Algorithm for Imbalanced Data Sets," Int. J. of Contents, vol. 8, no. 2, pp. 7-12, Jun. 2012, doi: https://dx.doi.org/10.5392/IJoC.2012.8.2.007.
[14] J. B. Hampshire and A. H. Waibel, "A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Networks," IEEE Trans. Neural Networks, vol. 1, pp. 216-218, Jun. 1990, doi: https://doi.org/10.1109/72.80233.
[15] S. H. Oh, "Hidden Node Pruning of Multilayer Perceptrons Based on Redundancy Reduction," Proc. Convergence on Hybrid Information Technology, Daejon, Korea, LNCS, vol. 6935, pp. 245-249, Sep. 22-24, 2011.
[16] J. Xu and D. W. C. Ho, "A New Training and Pruning Algorithm Based on Node Dependence and Jacobian Rank Deficiency," Neurocomputing, vol. 70, pp. 544-558, 2006, doi: https://doi.org/10.1016/j.neucom.2005.11.005.
[17] L. Zhang, et al., "Multivariate Nonlinear Modeling of Fluorescence Data by Neural Network with Hidden Node Pruning Algorithm," Analytica Chimica Acta, vol. 344, pp. 29-39, 1997, doi: https://doi.org/10.1016/S0003-2670(96)00628-9.
[18] A. P. Engelbrecht, "A New Pruning Heuristic Based on Variance Analysis of Sensitivity Information," IEEE Trans. Neural Networks, vol. 12, pp. 1386-1399, 2001, doi: https://doi.org/10.1109/72.963775.
[19] S. H. Oh, "Performance Improvement of Multilayer Perceptrons with Increased Output Nodes," Journal of the Korea Contents Association, vol. 9, no. 1, pp. 123-130, Jan. 2009, doi: https://doi.org/10.5392/JKCA.2009.9.1.123.
[20] D. A. Lienhard, David H. Hubel and Torsten N. Wiesel's Research on Optical Development in Kittens, Embryo Project Encyclopedia, Oct. 2017, 11, ISSN: 1940-5030, [Online] Available: https://hdl.handle.net/10776/12995
[21] S. H. Oh, "Comparisons of Linear Feature Extraction Methods," Journal of the Korea Contents Association, vol. 9, no. 4, pp. 121-130, Apr. 2009, doi: https://doi.org/10.5392/JKCA.2009.9.4.121.
[22] A. J. Bell and T. J. Sejnowski, "The Independent Components of Natural Scenes Are Edge Filters," Vision Res., vol. 37, pp. 3327-3338, 1997, doi: https://doi.org/10.1016/S0042-6989(97)00121-1.
[23] E. Doi, T. Inui, T. W. Lee, T. Wachtler, and T. J. Sejnowski, "Spatiochromatic Receptive Field Properties Derived from Information-Theoretic Analyses of Cone Mosaic Responses to Natural Scenes," Neural Computation, vol. 15, pp. 397-417, 2003, doi: https://doi.org/10.1162/089976603762552960.
[24] J. H. Lee, et al., "On the Efficient Speech Feature Extraction based on Independent Component Analysis," Neural Processing Letters, vol. 15, no. 3, pp. 235-245, 2002.
[25] Y. Lee, S. H. Oh, and M. W. Kim, "An Analysis of Premature Saturation in Back-Propagation Learning," Neural Networks, vol. 6, pp. 719-728, 1993, doi: https://doi.org/10.1016/S0893-6080(05)80116-9.
[26] S. Amari, "Natural Gradient Works Efficiently in Learning," Neural Computation, vol. 10, pp. 251-276, 1998, doi: https://doi.org/10.1162/089976698300017746.
[27] M. Girolami, A. Cichocki, and S. I. Amari, "A Common Neural-Network Model for Unsupervised Exploratory Data Analysis and Independent Component Analysis," IEEE Trans. Neural Networks, vol. 9, no. 6, pp. 1495-1501, 1998, doi: https://doi.org/10.1109/72.728398.
[28] D. S. Kim, S. Y. Lee, and R. M. Kil, "Auditory Processing of Speech Signals for Robust Speech Recognition in Real-World Noisy Environments," IEEE Trans. Speech and Autio Processing, vol. 7, pp. 56-69, 1999, doi: https://doi.org/10.1109/89.736331.
[29] S. H. Oh, "Development of Brain-Style Intelligent Information Processing Algorithm Through the Merge of Supervised and Unsupervised Learning: Generation of Exemplar Patterns for Training," Journal of the IEEK, vol. 41-CI, no. 6, pp. 61-67, Nov. 2004.
[30] S. H. Oh, "Improving the Subject Independent Classification of Implicit Intention by Generating Additional

Training Data with PCA and ICA," Int. J. of Contents, vol. 14, no. 4, pp. 24-29, Dec. 2018, doi: https://doi.org/10.539/IJoC.2018.14.4.024.