신경망 학습 과정중 불필요한 입력 정보 및 파라미터들의 제거

원 용 관† · 박 광 규††

요 약

형태 인식에서 유익한 특징정보의 선정 및 추출이 대단히 중요한 역할을 한다. 본 논문은 유익한 특징정보의 선정과 신경망의 학습을 동시에 수행할 수 있는 알고리즘을 기술한다. 알고리즘은 근본적으로 반복적으로 수행되는 세 단계로 구성 되어있는데, 이들은 학습, 연결자 제거, 그리고 입력 신경세포 제거이다. 초기 학습을 실행한후, 먼저 적은 절대값을 갖는 연결자들이 제거 된다. 그런 후, 내부 계층 신경세포들과 적은 숫자의 연결자들을 갖는 입력 신경세포들이 제거된다. 이 과정은 제거된 입력 신경세포들에 상응하는 특징정보들을 제외시키는 것과 동일하다. 만약, 에러값이 증가하면, 연결자 제거 및 입력 신경세포 제거 과정의 반복으로 구성된 신경망의 재학습을 실행한다. 그 결과, 알고리즘은 다른 공간계로의 변환없이 특징정보 추출 공간내에서 중요한 특징들을 선정하게 된다. 또한, 특징정보 선정이 형태 분류 관점에서의 성능과 긴밀하게 연결되어 수행되므로, 선정된 특징정보들은 형태 분류에 가장 좋은 정보를 제공한다. 이 알고리즘은 불필요 또는 그다지 중요하지 않은 정보의 추출로 인한 경제적 손실을 피할 수 있게한다. 더구나, 마지막에 얻어진 신경망은 인식 성능에 저해 요인이될 수 있는 불필요한 파라미터들, 즉, 가중 연결자 및 바이어스를 포함하지 않는다. 응용 결과, 본 알고리즘은 가장 좋은 정보를 갖는 특징들만을 남기며, 성능 저하를 일으키지 않으면서도 특징 벡터의 차원을 현저하게 줄였다.

Elimination of Redundant Input Information and Parameters during Neural Network Training

Yonggwan Won† · Kwang-Kyu Park††

ABSTRACT

Extraction and selection of the informative features play a central role in pattern recognition. This paper describes a modified back-propagation algorithm that performs selection of the informative features and trains a neural network simultaneously. The algorithm is mainly composed of three repetitive steps: training, connection pruning, and input unit elimination. After initial training, the connections that have small magnitude are first

† 정 회 원: 한국전자통신 연구소, 인공지능 연구실
†† 비 회 원: 한국전자통신 연구소, 인공지능 연구실
논문접수: 1996년 2월 10일, 심사완료: 1996년 3월 28일

pruned. Any input unit that has a small number of connections to the hidden units is deleted, which is equivalent to excluding the feature corresponding to that unit. If the error increases, the network is retrained, again followed by connection pruning and input unit elimination. As a result, the algorithm selects the most important features in the measurement space without a transformation to another space. Also, the selected features are the most informative ones for the classification, because feature selection is tightly coupled with the classification performance. This algorithm helps avoid measurement of redundant or less informative features, which may be expensive. Furthermore, the final network does not include redundant parameters, i.e., weights and biases, that may cause degradation of classification performance. In applications, the algorithm preserves the most informative features and significantly reduces the dimension of the feature vectors without performance degradation.

1. Introduction

The first step in designing a pattern recognition system is to measure features and to determine the best set of features. Selection of the best set helps realize a more efficient and accurate classification system. Previous works such as clustering transformation, entropy methods, K-L transform, and functional approximation focused on dimensional reduction of the patterns by means of a linear transformation [1, 2]. Since the dimension of patterns is reduced in the new space, it is still required that all features in the measurement space have to be measured when the system is in testing or operating mode. It should be inhibited specially when the features are not easily measurable and measurement cost is high. Furthermore, there is a trade-off between feature reduction and classification performance. In general, the classification performance is degraded with the dimensionally reduced pattern vectors. Therefore, selection of the important features in the measurement space is desirable, while retaining the classification performance as possible.

Major concern in the use of the artificial neural networks is the generalization capability. Among many factors that affect the generalization capability [3], the size of the neural network is the most important one because its relationship to the number of training examples, training time and problem difficulty mainly affects the generalization capability [4]. In general, the smallest network that fits the given data produces good generalization. However, choosing an appropriate network size is still not an easy task. Some researchers used learning theory to estimate the appropriate size of the network [5, 6, 7]. On the other hands, there is redundancy in a fully connected feedforward network, and this degrades the generalization capability [8]. There are several methodologies to improve the generalization capability through constraints on the weights, such as weight elimination [9, 10, 11, 12, 13], complexity regularization [14, 15, 16, 17], and structural constraints [18, 19]. It was shown that a network without redundant parameters(connections such as weights and biases) produced better performance for the classification of simulated random signals than the fully connected one [20].

In this paper, we describe a methodology to perform selection of the informative features and train a neural network simultaneously. Unlikely to other approaches, the selection of informative features is a classification-performance-dependent method, i.e., the selection of features is directly related to the classification performance. A modified back-propagation algorithm [13] that iteratively trains the network and eliminates the redundant connections is employed to eliminate input units that correspond to less informative features for the classification. After the connection pruning stage, the algorithm deletes any input

unit that has no or small number of weight connections to hidden units and does not degrade the performance with its elimination. Therefore, the dimension of the pattern vectors are reduced in the measurement space, while retaining the classification performance. This property helps avoid expensive measurement of redundant or less informative features. Furthermore, the final network does not include redundant connections, which makes the network architecture simpler.

This paper is composed of four sections. In section 2, we first introduce the training algorithm that reduces the redundant features and connection parameters as well. The connection pruning algorithm and its motivations are also described. Experimental results of this algorithm is reported in section 3. Three data sets, *iris* data, handwritten digits, and simulated random signals were concerned. Finally, in section 4, conclusion and possible future works are described.

2. Redundancy Elimination

This section describes the modified back-propagation (BP) algorithm that can reduce the number of connections [20] as well as eliminates the input units, which is equivalent to dimensional reduction. We first briefly introduce the motivations for the connection pruning algorithm [13].

2.1 Motivations for weight elimination algorithm

An algorithm that eliminates redundant connections was developed based on the distribution of weight magnitudes for different network sizes, effects of eliminating the connections on the classification boundary, and the nonlinearity of the neural network units [13]. As the network size increases for a certain problem, the number of connections with small magnitude increases in the trained network. This finding suggested that connections with a small magnitude should be eliminated. A small tilt of boundaries, which is equivalent to a small change on the connection values, does not degrade the classification performance. It sometimes improved the performance by reducing the overfitting and pattern memorization problems [20]. Nonlinearity of the sigmoid activation function also allows us to eliminate the connections with a small magnitude. In a situation that the output of a unit is close to the extreme value, i.e., 0 or 1, a small change on its netinput makes an negligible effect on the output value.

2.2 Training Algorithm for Pruning Redundant Connections

Based on those speculations described in the previous section, the connection pruning was developed [13]. The algorithm first trains the network using the standard back-propagation (BP) learning scheme to adjust all weights (initial back-propagation training). When this initial training reaches either the specified error or number of maximum training epochs, all weights and biases with a magnitude below a specified threshold value are set to zero (i.e., pruning). If this pruning does not increase the error, the algorithm terminates; otherwise it resumes training (i.e., retraining) by again adjusting all the connections, also using the back-propagation learning scheme, until the network reaches either the error obtained by the initial back-propagation training or the specified number of epochs for each retraining cycle. This two-step process, retraining and pruning, continues until the error caused by pruning is less than that obtained by the initially trained network or until the total number of retraining epochs reaches its predefined value. The algorithm is summarized with the following pseudo code:

```
Initialize the Network.
DO      update the weights by BP learning.
UNTIL the number of epoch reaches a specified value
          OR the error become less than a specified
          value.
Prune   the weights.
```

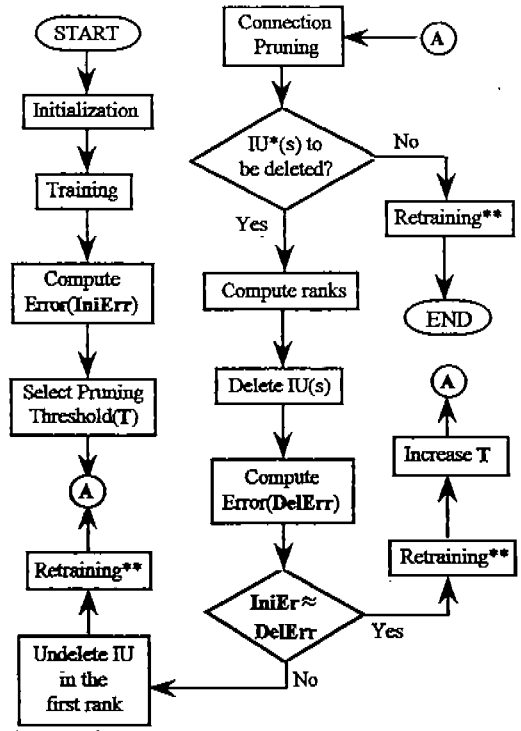
WHILE the total training epoch does not reach a specified number AND the error with pruned network is larger than the error at the end of the initial BP learning.
    DO update the weights by BP learning.
    UNTIL the specified number of retraining epochs per retraining cycle is reached OR the error becomes less than or equal to the error at the end of the initial BP learning.
    Prune the weights.
ENDWHILE

In applications, this algorithm produced the minimum network for the XOR problem and pruned unnecessary connections to learn a Boolean function, Rule-and-Exception [13]. Furthermore, it pruned some redundant connections from the minimum-sized network, producing better performances for the classification of the simulated random signals [20].

2.3 Training Algorithm for Feature Reduction

We modified the connection pruning algorithm in order to meet the feature reduction purpose. The algorithm is mainly composed of three repetitive steps: training, connection pruning, and input unit elimination. This three-step procedure is repeated until no input unit is eliminated and the degree of training is acceptable.

After pruning the connections, the algorithm eliminates any input unit which has no or a small number of weight connections to the first hidden layer. The input units to be eliminated are ranked by the summation of the magnitudes of the unpruned weights associated to the units. The unit with a larger value is ranked first. The hidden unit, if any, that does not have any weight connections to the input units is also eliminated. It is possible that, with the dimensionally reduced pattern set, retraining may not achieve reasonable degree of training. This possibility is measured by the difference between the error obtained at the end of the initial training and that obtained after input unit elimination. In this situation, the feature associated with the input unit which is in the first rank is included in the next retraining stage (i.e., undeleting the input unit). For the retraining process, all connections to remaining units are involved. The algorithm is summarized by the flow chart in Figure 1.



(Fig. 1) Flow chart of the feature reduction algorithm.

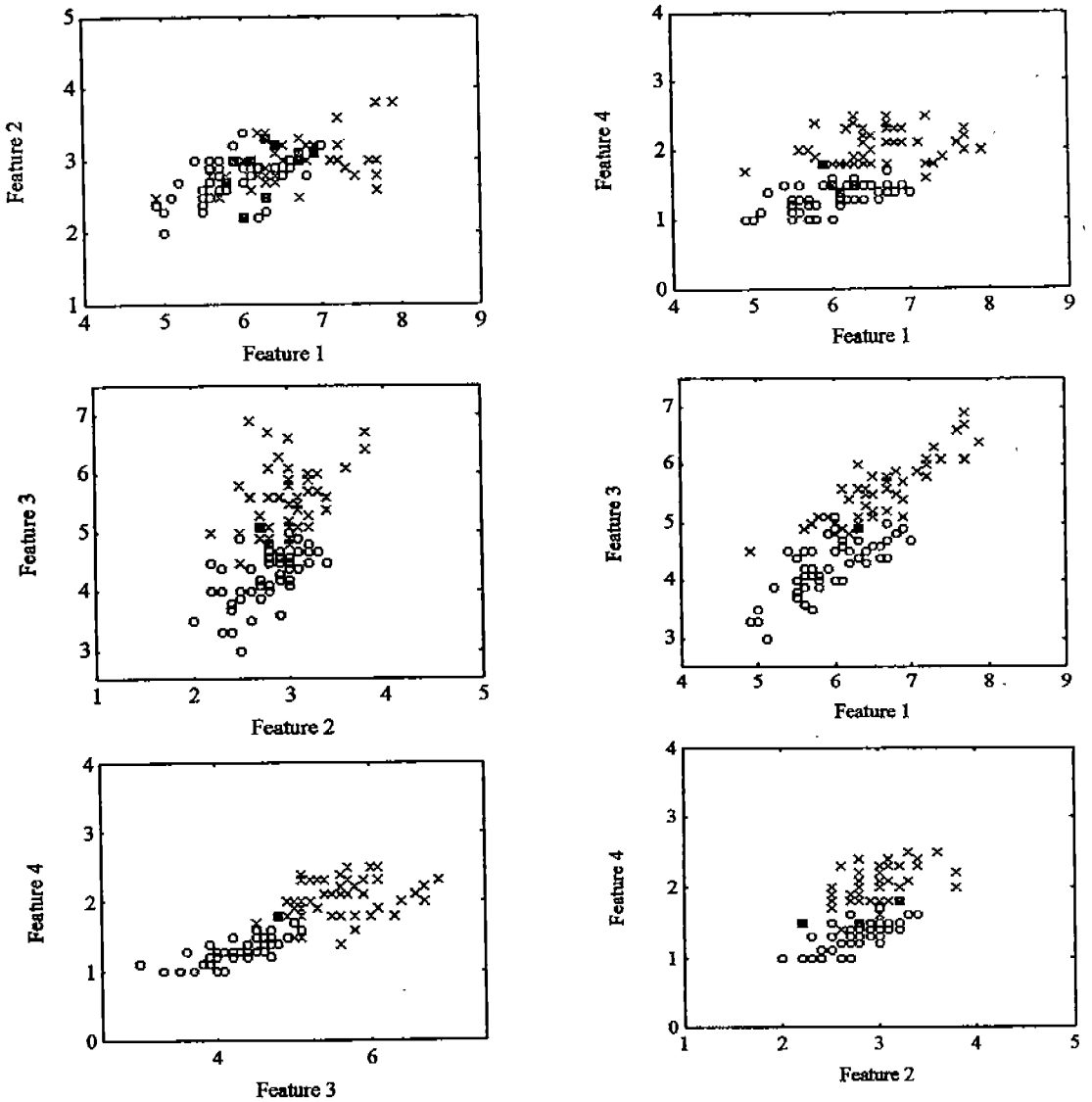
3. Experimental Results

This section provides the application results of our feature reduction algorithm. We concerned three classification problems: classifications of the iris data set, the handwritten digits and the simulated random signals.

3.1 *Iris* data set

The iris data is originally composed of three classes of Iris, namely, *setosa*, *virginica*, and *versicolar*, 50 patterns from each class with four-dimensional features. For our study, we only considered the class 2 and the class 3. Figure 2 shows the 2-dimensional plots of the patterns in these classes with different pairs of features. Patterns in class 2 are indicated by o and those in class 3 by x. It is obvious that the pair of feature 1 and feature 2 makes the classification the most difficult. Therefore, it is adequate that a classifier uses features other than only this pair of features.

We performed several experiments with this data set. All patterns were used only for training. The



(Fig. 2) Plot for the class 2 and 3 of the iris data

feedforward neural network had a single hidden layer with three hidden units. The network was trained initially until the RMSE was less than 0.1 or the training epochs reached 500 with the learning rate 0.05 and momentum 0.9. The network was trained well through the initial training by producing a very low error such as the correction rate 94% and the RMSE 0.15. Initial threshold for connection pruning (T) was 0.7 and increased by 0.1, if necessary. Any input unit that had weight connections less than three was deleted at the input unit elimination stage. The input unit in the highest rank was undeleted if **DelErr** was 1.7 times larger than **IniErr.**

Maximum retraining epoch was set to 50 and entire training took about 650 epochs. The RMSE and the correction rate produced by the final networks were similar to those at the end of the initial training. Our algorithm mostly preserved two features, favorably feature 3 and feature 4. In very few cases, only one feature was eliminated. The pair of feature 1 and feature 2 was never solely remained. The results justify that our algorithm selects the most informative features. Furthermore, the training results were similar to those obtained with all four features.

3.2 Handwritten digit data set

From the handwritten digit data base which were extracted from the USPS mail pieces [21], 500 digits for each class were collected. The digit images were then normalized to the fixed size of 24×18 using moment normalization [22]. Some samples of normalized handwritten digits are shown in Fig. 3. Among the collected digits, 400 per class were used for training and 100 for testing the network.

We first extracted numerical features from the normalized digit images. Note that goodness of the feature extraction method we used was not evaluated. Development of a good feature extraction algorithm is outside the scope of this study. A 5×5 window scanned the images, downsampling both horizontally and vertically by two, and the number of black pixels

(Fig. 3) Some examples of the size-normalized handwritten digits.

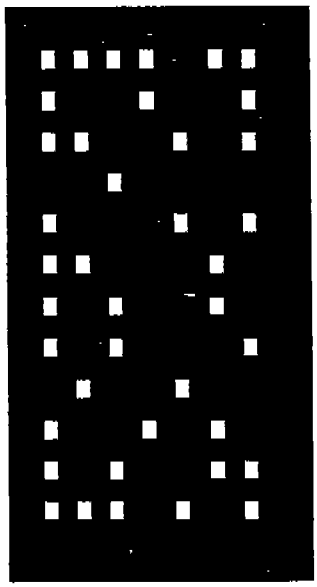
inside the window was counted. The result was normalized between 0 and 1. Finally, a row-ordered pattern vector in 84 dimensions was obtained from each binary digit image. This pattern vector was used as the input pattern for the neural network.

A feedforward neural network that had a single hidden layer with 30 hidden units was used. The network was initially trained with learning rate 0.02 and momentum 0.8 until the RMSE was less than 0.05 or the training epoch reached 20,000. The maximum retraining epoch for each retraining cycle was 2,000. The initial connection pruning threshold T and its increment were the same as those for the iris data set. The input unit in the highest rank was also undeleted if **DelErr** was 1.7 times larger than **IniErr.**

We conducted several experiments with different initial weights. Total training cycles were about 30,000. The initial training generally produced very low RMSE and about 99% classification for the training set and 93% for the test set. Our feature reduction algorithm significantly reduced the number of features without performance degradation in both training and testing. The reduction ratios were between 42% and 51%.

Figure 4 shows a 24×18 grid map in which each

cell indicates a pixel in the images. A shaded cell indicates the pixel that was not involved in numerical feature extraction. The cell in black indicates that the feature extracted from the corresponding pixel was left after training with our feature reduction algorithm. Note that most of the features extracted from the pixels around the edge of the images is not useful. In general, the edge of the most images does not have strokes. This result also justifies that our feature reduction algorithm preserves the most informative features. Furthermore, it suggests that, in testing stage, feature extraction be performed at the location where the extracted feature is left after feature reduction training.



(Fig. 4) A grid map that shows the pixels from which the most informative features were extracted.

3.3 Simulated Random Signals

A random signal generation process produces a set of class patterns using a mathematical model which generates pulse patterns with exponentially damped oscillatory edges [23]. This model is described by

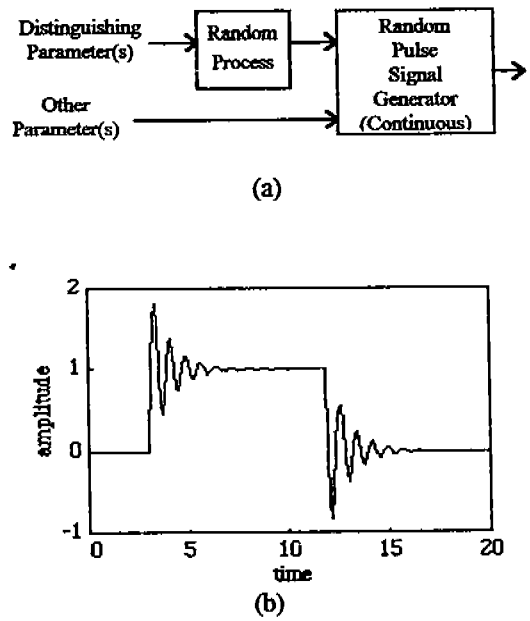
$$y(t) = A[x(t-T_o) - x(t-T_o-T_w)] \quad (1)$$

where

$$x(t) = [1 - e^{-Ct} \sin(2Ft)]\, u(t). \quad (2)$$

There are five parameters: amplitude(A), starting time (T_o), pulse width(T_w), exponential coefficient(C) and frequency of oscillations(F).

For each pattern, the process first selects the parameters with which the classes are distinguished. The values for the distinguishing parameters are obtained from a Gaussian distribution with a specified mean and variance. In multiple-class problems, separated processes represent each class, and at least one of the parameters has to have a different distribution for each class. In other words, the classes are distinguished by the mean vectors of the parameters among which the distinguishing ones have random values. Figure 5 illustrates the process and the continuous pulse signal. The pulse signal is then sampled to obtain the pattern vector.



(Fig. 5) (a) Random signal generation process and (b) the pulse signal.

The problem difficulty is determined by the statistical difference between processes, which is described by the Mahalanobis distance (R) between the parameter distributions. Let $\mathbf{m}_1$ and $\mathbf{m}_2$ be the mean vector of the parameters and C be the covariance matrix. Then, the problem difficulty is measured by

$$R=[(\mathbf{m}_1-\mathbf{m}_2)^t\, C^{-1}(\mathbf{m}_1-\mathbf{m}_2)]^{1/2}. \quad (3)$$

Therefore, the larger value defines an easy problem. We assumed that random variables are independent and variances are identical for each distribution. Under this assumption, the measurement is simplified to

$$R=\sigma^{-2}[(\mathbf{m}_1-\mathbf{m}_2)^t\,(\mathbf{m}_1-\mathbf{m}_2)]^{1/2}. \quad (4)$$

where σ is the identical variance.

We considered two-class problems. The statistical difference between two classes was 5. 300 signals were collected from each process with the exponential coefficient and the frequency of oscillation as the distinguishing parameters. We selected the values for those parameters in order for the head and the tail of the signals to be identical. The continuous signal was then uniformly sampled to have 200 dimensional pattern vectors.

A network that had a single hidden layer with four units was used. 200 patterns from each class was used for training the network. The network was initially trained with learning rate 0.02 and momentum 0.8 until the training epoch reached 500 or the RMSE was below 0.05. The maximum retraining epoch for each retraining cycle was 50. All the other training parameters were the same as those for the other data set.

In general, total training cycles were about 700. The network trained by the standard BP algorithm produced about 94% classification rates for the training set and 86% for the test set. Our feature reduction algorithm significantly reduced the number of features without performance degradation in both training and testing. Furthermore, all of the features corresponding to the head (before To) and the tail of the pulse was eliminated all the time. This is obvious since the head and the tail of the pulse signals were identical. This result again justifies that our algorithm eliminates redundant information.

4. Conclusion

We have introduced an algorithm that reduces redundant information during training a neural network. Our algorithm is a classification-performance-dependent method, i.e., the selection of features is directly related to the classification performance. In other words, training a network and performing feature reduction are tightly coupled. Therefore, the remaining features can be considered a better set in term of classification with the neural network. Furthermore, the final network does not include redundant connections (i.e., weights and biases), which makes the network architecture simpler.

Unlikely to other feature reduction approaches that use a transformation to another space [1, 2], thus loose meaning of the original feature, our algorithm preserves the description of the features, since it reduces the dimension of the pattern vectors in the measurement space. This property is very important because one can avoid expensive measurement of redundant or less informative features.

In applications, our training algorithm eliminated redundant information, while preserving the most informative features. With the iris data set, it mostly reduced two features out of four without degradation of performance. A pair of features that makes the classification problem the most difficult never remained. In the handwritten digit recognition, it significantly reduced the dimension of the feature vectors. With the simulated random signals that included useless information in the head and the tail, the algorithm always eliminated the corresponding features.

Our feature reduction algorithm seemed to be sensi-

tive to selection of connection pruning threshold T and its increment. More work is required to develop a systematic method to select these values. It is closely related to training time and reduction performance. In general, our algorithm took longer training time than the standard algorithm. Therefore, we need to employ a methodology for fast convergence [24, 25, 26].

REFERENCES

[1] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles,* Addison-Wesley Publishing Company, Inc., Massachusetts, 1974.

[2] S. Banks, *Signal Processing, Image Processing, and Pattern Recognition,* Prentice Hall, Englewood Cliffs, 1990.

[3] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation,* Addison-Wesley Publishing Company, Redwood City, CA, 1991.

[4] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," in D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* MIT Press, Cambridge, MA, 1986.

[5] A. Blumer, A. Ehrenfeucht, D. Haussler and M. Warmuth, "Learnability and the Vapnik-Chervonenkis dimension," *J. Ass. Comput. Mach.,* Vol. 36, No. 4, pp. 929-965, 1989.

[6] A. Ehrenfeucht, D. Haussler, M Kearns and L. Valiant, "A general lower bound on the number of examples needed for learning," in *Proc. 1988 Workshop Computational Learning Theory,* 1988.

[7] L. Valiant, "A Theory of the learnable," *Commun. Ass. Comput. Mach.,* Vol. 27, No. 11, pp. 1134-1142, 1984.

[8] M. Morgan and H. Bourlard, "Generalization and Parameter estimation in Feedforward Nets: Some experiments," in *Advances in Neural Information Processing Systems,* Vol. II, D. Touretzky, ed., Morgan Kaufmann Publisher, pp. 630-637, 1989.

[9] Y. le Cun, J.S. Denker, and S.A. Solla, "Optimal Brain Damage," in *Advances in Neural Information Processing Systems,* Vol. II, D. Touretzky, ed., Morgan Kaufmann Publisher, pp. 598-605, 1990.

[10] E. Karnin, "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks," *IEEE Trans. on Neural Networks,* Vol. 1, pp. 239-242, 1990.

[11] J. Sietsma and R. J. F Dow, "Creating Artificial Neural Networks that Generalize," *Neural Networks,* Vol. 4, pp. 67-79, 1991.

[12] M. C. Mozer and P. Smolensky, "Skeletonization : A Technique for Trimming the Fat from a Network via Relevance Assessment," in *Advances in Neural Information Processing Systems,* Vol. I, David S. Touretzky (Ed.), Morgan Kaufmann Publisher, pp. 107-115, 1989.

[13] Y. Won, Connection pruning algorithms and their comparison with the standard back-propagation algorithm, Master Thesis, University of Missouri, 1991.

[14] S.J. Hanson and L.Y. Pratt, "Comparing biases for minimal network construction with back-propagation," in *Advances in Neural Information Processing Systems,* Vol. I, D. Touretzky, ed., Morgan Kaufmann Publisher, pp. 117-185, 1989.

[15] G.E. Hinton, "Learning Distributed Representations of Concepts," *Proc. of the English Annual Conference of the Cognitive Science Society,* pp. 1-12, Amherst, 1986.

[16] S.J. Nowland and G.E. Hinton, "Simplifying Neural Network by Soft Weight Sharing," *Neural Computation,* Vol. 4, No. 4, pp. 473-493, 1992.

[17] A.S. Weigend, D.E. Rumelhart, and A. Huberman, "Back-propagation Weight Elimination and Time Series Prediction," *Proc. of the 1990 Connectionist Models Summer School,* Morgan

Kaufmann, pp. 65-80, 1990.

[18] Y. le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, "Handwritten Digit Recognition: Application of Neural Network Chips and Automatic Learning," *IEEE Communications Magazine*, November, pp. 41-64, 1989.

[19] A. Waibel, Y. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition using Time-Delayed Neural Network," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 37, No. 3, pp. 328-339, 1989.

[20] Y. Won and B. Min, "A simple Connection Pruning Algorithm and its Application to Simulated Random Signal Classification," *Trans. of Korea Information Processing Society*, Vol3, No. 2, 1996.

[21] P.D. Gader and Mohamed A. Khabou, "Automated Feature Generation for Handwritten Digit Recognition by Neural Networks," *Proc. of the Intl Workshop Frontiers Handwriting Recognition*, pp. 21-31, Buffalo, NY, 1993.

[22] R. Casey, "Moment Normalization of Handprinted Characters," in *IBM J. Res. Develop.*, pp. 548-557, September, 1970.

[23] D. S. Park, Relationship Between the Performance of Multilayer Neural Network Pattern Classifiers and the Statistics of Pattern Generating Processes, Ph. D. Dissertation, University of Missouri-Columbia, 1990.

[24] A. Jacobs, "Increased Rates of Convergence through Learning Rate Adaptation," *Neural Networks*, Vol. I, pp. 295-307, 1988.

[25] J. Sun, W. I. Grosky, and M Hassoun, "A Fast Algorithm for Finding Global Minima of Error Functions in Layered Neural Networks," *IJCNN International Joint Conference on Neural Networks*, Vol. I, pp. 715-720, 1990.

[26] S.-H. Oh and Y. Lee, "A Modified Error Function to Improve the Error Back-Propagation Algorithm for Multi-Layer Perceptrons," *ETRI Journal*, Vol. 17, No. 1, pp. 11-22, 1995.

원 용 관

1987년 한양대학교, 전자공학과(학사)
1991년 University of Missouri, Electrical and Computer Engineering(석사)
1995년 University of Missouri, Electrical and Computer Engineering(박사)
1987년~1988년 금성통신 근무
1992년~1993년 U.S. Postal Service 지원, 필기체 문자 인식
1994년~1995년 U.S. Air Force 지원, 자동 목표물 인식
1995년 11월~현재 한국전자통신연구소, 인공지능연구실
관심분야: Image/Signal Processing, Pattern Recognition(자동목표물, 문자, 얼구), Virtual Reality, Neural networks, Fuzzy logic

박 광 규

1985년 경희대학교 수학과(학사)
1987년 한국과학기술원 응용수학과(석사)
1991년 한국과학기술원 응용수학과(박사)
1991년~현재 한국전자통신연구소, 인공지능연구실 근무
관심분야: 멀티미디어 통신, 실시간 OS, 인공 지능, Wavelet