Discriminative Training of Predictive Neural Network Models

예측신경회로망 모델의 변별력 있는 학습

Kyung Min Na*, Jae Yeol Rheem*, Souguil Ann*

나 경 민*, 임 재 열*, 안 수 길*

ABSTRACT

Predictive neural network models are powerful speech recognition models based on a nonlinear pattern prediction. But those models suffer from poor discrimination between acoustically similar words. In this paper, we propose a discriminative training algorithm for predictive neural network models. This algorithm is derived from GPD (Generalized Probabilistic Descent) algorithm coupled with MCEF(Minimum Classification Error Formulation). It allows direct minimization of a recognition error rate. Evaluation of our training algorithm on ten Korean digits shows its effectiveness by 30% reduction of recognition error.

요 약

예측신경회로망 모델은 패턴 예측에 의한 매우 효과적인 음성인식 모델이다. 그러나, 그러한 모델은 유사한 어휘간에서 변별력이 떨어지는 단점이 있다. 이 논문에서는 그러한 단점을 극복하기 위한 변별력있는 학습 알고리즘을 제안한다. 이 알고리즘은 최소 분류 오차 수식화와 GPD 알고리즘으로부터 유도되며 그에 따라서 인식 오차의 수를 직접 최소화하는 것이 가능하다. 한국어 숫자음에 대한 인식 실험결과, 기존의 알고리즘에서 발생하는 오인식의 30%를 줄일 수 있었다.

Ⅰ. Introduction

Recently, predictive neural network models and their effective training algorithms have been proposed for speech recognition [1-3]. Those models are superior to their conventional neural rivals for speech recognition in that 1) they can efficiently normalize the nonstationary time-variability of speech signal, 2) they are easily applicable to continuous speech recognition, 3) they need not to be entirely retrained when new word classes are added and 4) required amount of training data is relatively small.

In predictive neural network models, an MLP (Multi-layer Pereptron) is used as a nonlinear predictor of adjacent speech feature vectors and DP(Dynamic Programming) algorithm [1-2] or Viterbi algorithm [3] is jointly used for time alignment process. A single word is modeled by a sequence of such MLP predictors and the switching between MLP's is determined along the opti-

*서울대 전자공학과

접수일자 : 1994년 1월 4일

mal path from time alignment algorithm. As a recognition cue, the resulting minimum accumulated prediction residual along that path is used.

Generally, predictive neural network models can be divided into several categories such as NPM (Neural Prediction Model) by K. Iso [1], LPNN (Linked Predictive Neural Network) by J. Tebelskis [2], and HCNN(Hidden Control Neural Network) by E. Levin [3]. NPM and LPNN are basically similar models in their training and recognition algorithms. But HCNN is different from those two models roughly in three points. First, HCNN is trained by the joint combination of EBP(Error Backpropagation) algorithm and Viterbi algorithm while NPM and LPNN are trained by the joint combination of EBP algorithm and DP algorithm. Second, HCNN modulated network weight parameters by introducing hidden control signals in input layer, not by directly switching MLP's as NPM and LPNN. Last, recognition of a word is performed by finding the best state sequence (hidden control signals) for minimum accumulated prediction residual using the Viterbi algorithm, not by directly comparing minimum accumulated prediction residuals of each word model.

However, in spite of the above-mentioned superiorities, predictive neural network models suffer from poor discrimination between acoustically similar speech data. It's because the conventional training algorithm trains each predictor with only one word class while not considering the training states of the other word classes. So, minimum accumulated prediction residual from class m and network weight parameter set of near-miss class n can be small enough to make class m and class n so confusable. That conventional training algorithm is EBP algorithm coupled with DP algorithm or Viterbi algorithm.

In this paper, we propose an effective discriminative training algorithm based on GPD(Generalized Probabilistic Descent) algorithm coupled with MCEF(Minimum Classification Error Formulation) [4-7]. GPD algorithm with MCEF has proven successful in improving the discrimination powers of the conventional recognizers such as DTW based recognizer [4] and HMM-based recognizer [5]. This algorithm directly minimizes an expected recognition error instead of minimizing the accumulated prediction residual. We apply this algorithm to the predictive neural network models and derive new discriminative training algorithm formulas. This new algorithm not only tries to minimize the accumulated prediction residual of correct class by the gradient descent method but also tries to maximize that of the other classes by the gradient ascent method, all along their optimal paths.

Experiments on ten Korean digits have been carried out, and the proposed training algorithm has achieved totally 30% reduction of recognition error compared with the conventional EBP algorithm. Among the several predictive neural network models, NPM(Neural Prediction Model) is chosen for experiment is this paper. But it does not matter because the proposed training algorithm can be considered as a modified EBP algorithm and can be easily applied to other models.

Ⅱ. Npm and its Conventional Training Algorithm

2.1 NPM(Neural Prdiction Model)

NPM(Neural Prediction Model) uses a sequence of MLP's as a separate nonlinear predictor for each word class and effectively normalizes temporal distortion of speech signal using dynamic programming technique. Particularly, temporal correlations between successive speech feature vectors are efficiently modeled by the MLP approximators.

Fig. 1 represents the structure of an MLP predictor. This MLP predictor outputs a predicted speech feature vector $\hat{S}_t^m = (\hat{s}_{1,t}^m, \hat{s}_{2,t}^m, \ldots, \hat{s}_{K,t}^m)$ using the preceding input speech feature vectors $S_{t-\tau}$, $S_{t-\tau+1}, \cdots, S_{t-1}$, where $S_t = (s_{1,t}, s_{2,t}, \cdots, s_{K,t})$, if a word is included in C^m(among M word classes C^i,

$t = 1, 2, \cdots, M$). The symbol τ represents the number of input speech feature vectors for prediction. Let $W^m_{n(t)} = (w^m_{jk,n(t)})$ be a weight matrix between hidden layer and output layer of $n(t)$-th predictor for a word model, m, $V^m_{n(t)} = (v^m_{ij,n(t)})$ be a weight matrix between input layer and hidden layer of $n(t)$-th predictor for a word m, $H^m_t = (h^m_{j,t})$ be an output from hidden unit at time t, and $f(A)$ be a sigmoid function, which operates on each element of a matrix A. Given an optimal path $(t, n(t))$ and an input vector $\bar{S}_t = (s_{1,t-\tau}, \cdots, s_{K,t-\tau}, \cdots, s_{K,t-1})$, the input-output relation for the MLP predictor is as follows :

$$H^m_t = f\left(V^m_{n(t)} \cdot \bar{S}_t\right), \tag{1}$$

$$\hat{S}^m_t = W^m_{n(t)} \cdot H^m_t \tag{2}$$

From the predicted speech feature vector $\hat{S}^m_t$, a prediction residual $\|\hat{S}^m_t - \bar{S}_t\|^2$ is calculated.

A word model is represented as a sequence of such MLP predictors. Fig. 2 illustrates such a model for word m, where each circle denotes an MLP predictor and N_m is its total number.

In training phase, the optimal segmentation of input speech feature vectors is done on the resulting prediction error matrix by DP algorithm to minimize the accumlated prediction residual $D(m)$.

$$D(m) = \min_{n(t)} \sum_{t=1}^{T} \left\|\hat{S}^m_t(t, n(t)) - \bar{S}_t\right\|^2 \tag{3}$$

Along that optimal path $(t, n(t))$, the conventional EBP algorithm is carried out.

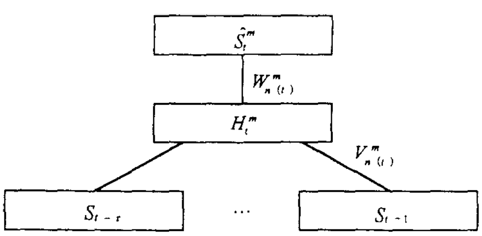


Fig 1. MLP predictor.

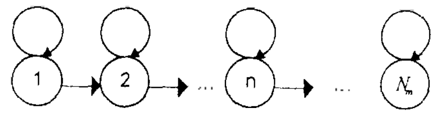


Fig 2. NPM for word in C^m.

2.2 Conventional EBP algorithm

If a training word is included in C^m, the conventional EBP algorithm formulas are as follows :

$$(w^m_{jk,n(t)})_{q+1} = (w^m_{jk,n(t)})_q + \eta(s_{k,t} - \hat{s}^m_{k,t})h^m_{j,t}, \tag{4.a}$$

$$(v^m_{ij,n(t)})_{q+1} = (v^m_{ij,n(t)})_q + \eta\delta^m_{j,t}h^m_{j,t}(1-h^m_{j,t})\bar{s}_{i,t},$$

$$\text{where} \quad \hat{s}^m_{k,t} = \sum_{j=1}^{J} h^m_{j,t} \cdot w^m_{jk,n(t)}, \quad h^m_{j,t} = f(\sum_{i=1}^{I} \bar{s}_i \cdot v^m_{ij,n(t)}),$$

$$\text{and} \quad \delta^m_{j,t} = \sum_{k=1}^{K}\left(s_{k,t} - \hat{s}^m_{k,t}\right) \cdot w^m_{jk,n(t)}. \tag{4.b}$$

η is a learning coefficient. I, J, and K are the number of input units, hidden units, and output units, respectively. $\bar{S}_t = (s_{1,t-\tau}, \cdots, s_{K,t-\tau}, \cdots, s_{K,t-1}) = (\bar{s}_{1,t}, \cdots, \bar{s}_{i,t}, \cdots, \bar{s}_{I,t})$ is an input vector.

Usually, a sigmoid function is used in output units as a nonlinear activation function. But note that a linear activation function is adopted in NPM because the MLP predictor is used as a universal approximator here.

The whole training procedure for NPM is shortly summarized below.

Step 1. Initialize all MLP predictor weight parameter sets.

Step 2. Compute the minimum accumulated prediction residual using the DP algorithm and find the optimal path by its backtracking.

Step 3. Update weight parameters of each MLP predictor along that optimal path using the conventional EBP algorithm.

Step 4. Repeat the above step 2 and step 3 for all training data.

Ⅲ. Proposed Discriminative Training Algorithm

3.1 MCEF(Minimum Classification Error Formulation)

This approach embeds both classification error count function and decision rule into one smoothing function, and applies the gradient descent search method to the function. To derive that function, three step MCEF(Minimum Classification Error Formulation) is required[4-8].

Step 1. An appropriate discriminant function $g_m(x, V, W)$ has to be chosen. This function is used as the decision rule in classification.

$$g_m(x,V,W) = \ln\left\{\sum_{\theta=1}^{\Theta} e^{-\left[\sum_{t=1}^{T} D_m^{\theta}(t,n(t))\right]\rho}\right\}^{-\frac{1}{\rho}} \quad (5.a)$$

$\sum_{t=1}^{T} D_{\theta}^{m}(t, n(t))$ is an accumulated prediction residual along the θ-th best path among all the possible Θ paths. If $\rho \to \infty$, then eq. (5.a) becomes the minimum prediction residual along the best optimal path θ^*.

$$g_m(x,V,W) = \min_{n(t)} \sum_{t=1}^{T} D_m^{\theta^*}(t,n(t)),$$
$$\text{where } D_m^{\theta^*}(t,n(t)) = \sum_{k=1}^{K}\left(s_{k,t} - \hat{s}_{k,t}^{m}\right)^2 \quad (5.b)$$

Eq. (5.b) is adopted in this paper.

Step 2. A misclassification function $d_m(x, V, W)$ is properly chosen. The introduction of the misclassification function is a key difference from the conventional formulations. A larger $d_m(x, V, W)$ implies that the input x is misclassified more definitely. A general form of this function is shown in eq. (6.a). By controlling the value of ξ, the range of competing classes that can participate in the process of optimizing the recognizer is determined.

$$d_m(x,V,W) = g_m(x,V,W) - \ln\left[\frac{1}{M-1}\sum_{l,l\neq m} e^{-g_l(x,V,W)\xi}\right]^{-\frac{1}{\xi}} \quad (6.a)$$

In extreme case, if $\xi \to \infty$, the misclassification function of eq. (6.a) becomes eq. (6.b) as $(M-1)^{-\frac{1}{\xi}} \to 1$. In resulting eq. (6.b), $d_m(x, V, W) \geq 0$ means that a misclassification has been occurred. And a word class l' is the most confusable class to the correct word class m.

$$d_m(x,V,W) = g_m(x,V,W) - g_{l'}(x,V,W) \quad (6.b)$$

Step 3. A smoothed loss function is introduced. It is a kind of cost function in Bayesian estimation. We choose a sigmoid function. A general form of the loss function can be expressed as a function of the misclassification function. The loss function l_m and the misclassification function d_m can be defined individually for each class m for generality. But this loss function $l_m(d_m)$ represents an approximation to the error probability for the m-th class data, and is directly related to the classification error rate.

$$l_m(x,V,W) = l_m(d_m(x,V,W)) = \frac{1}{1+e^{-\alpha d_m}}, \quad (7)$$

where α is a positive constant for scaling.

The above three functions are chosen as continuous functions with respect to the network weight parameter sets in order that the gradient descent search method can be easily applied. The above formulation allows direct minimization of the expected recognition error by the gradient descent search method, instead of minimzing the accumulated prediction residual.

3.2 New Discriminative Training Formulas by GPD Algorithm

Now, the proposed discriminative training algorithm formulas will be derived below. The expected recognition error as an objective criterion and probabilistic descent methods are defined as follows :

$$L(V,W) = \sum_m l_m(x,V,W),$$
$$V_{t+1}^m = V_t^m + \delta V_t^m, \quad \text{where } \delta V_t^m = -\eta U \nabla l_m \quad (9)$$

in a matrix form.

$$W_{t+1}^m = W_t^m + \delta W_t^m, \quad \text{where } \delta W_t^m = -\eta U \nabla l_m \qquad (10)$$

in a matrix form.

U is a positive-definite matrix(identity matrix in this paper), η is a positive real number for learning step size, and ∇ is a notation for gradient.

The final goal is to derive a new adaptation formula such that $E[\delta L(V, W)] < 0$, where $E[\cdot]$ is a notation for expectation, and $\delta L(V, W) = -\eta U \nabla l_m(x, V, W)$, and such that the weight parameter converges to an at least locally optimum solution. By the probabilistic descent theorem [4], this goal can be satisfied if a step size sequence

$\eta = \{\eta_i\}$ satisfies 1) $\sum_{i=1}^{\infty} \eta_i \rightarrow \infty$ and 2) $\sum_{i=1}^{\infty} \eta_i^2 < \infty$.

By applying the gradient descent search method and combining eq.s (5.b), (6.a) and (7) with (8), (9) and (10), new discriminative training algorithm formula eq. (12) is derived. Let a current training word be S.

For $S \in C^m$

$$\frac{\partial L(V,W)}{\partial w_{jk,n(t)}^m} = \frac{\partial L(V,W)}{\partial l_m(d_m)} \cdot \frac{\partial l_m(d_m)}{\partial d_m(x,V,W)} \cdot \frac{\partial d_m(x,V,W)}{\partial g_m(x,V,W)} \cdot \frac{\partial g_m(x,V,W)}{\partial w_{jk,n(t)}^m}$$
$$= -2\alpha l_m(1-l_m)(s_{k,t} - \hat{s}_{k,t}^m) h_{j,t}^m, \qquad (11.a)$$

$$\frac{\partial L(V,W)}{\partial v_{ij,n(t)}^m} = \frac{\partial L(V,W)}{\partial l_m(d_m)} \cdot \frac{\partial l_m(d_m)}{\partial d_m(x,V,W)} \cdot \frac{\partial d_m(x,V,W)}{\partial g_m(x,V,W)} \cdot \frac{\partial g_m(x,V,W)}{\partial v_{ij,n(t)}^m}$$
$$= -2\alpha l_m(1-l_m) \delta_{j,t}^m h_{j,t}^m (1-h_{j,t}^m) \bar{s}_{i,t},$$

where $\delta_{j,t}^m = \sum_{k=1}^{k=K} (s_{k,t} - \hat{s}_{k,t}^m) w_{jk,n(t)}^m$. (11.b)

For $l \neq m$

$$\frac{\partial L(V,W)}{\partial w_{k,n^*(t)}^l} = \frac{\partial L(V,W)}{\partial l_m(d_m)} \cdot \frac{\partial l_m(d_m)}{\partial d_m(x,V,W)} \cdot \frac{\partial d_m(x,V,W)}{\partial g_l(x,V,W)} \cdot \frac{\partial g_l(x,V,W)}{\partial w_{jk,n^*(t)}^l}$$
$$= 2\alpha l_m(1-l_m) v_l (s_{k,t} - \hat{s}_{k,t}^l) h_{j,t}^l, \qquad (11.c)$$

$$\frac{\partial L(V,W)}{\partial v_{ij,n^*(t)}^l} = \frac{\partial L(V,W)}{\partial l_m(d_m)} \cdot \frac{\partial l_m(d_m)}{\partial d_m(x,V,W)} \cdot \frac{\partial d_m(x,V,W)}{\partial g_l(x,V,W)} \cdot \frac{\partial g_l(x,V,W)}{\partial v_{ij,n^*(t)}^l}$$
$$= 2\alpha l_m(1-l_m) v_l \delta_{j,t}^l h_{j,t}^l (1-h_{j,t}^l) \bar{s}_{i,t}, \qquad (11.d)$$

where $v_l = \dfrac{e^{-g_l(x,V,W)\xi}}{\sum_{s,s\neq m} e^{-g_s(x,V,W)\xi}}$ and $\delta_{j,t}^l = \sum_{k=1}^{k=K} (s_{k,t} - \hat{s}_{k,t}^l) w_{jk,n^*(t)}^l$.

From eq. (11), new discriminative formulas are derived as below.

$$\delta w_{jk,n(t)}^m = \eta \alpha l_m(1-l_m)(s_{k,t} - \hat{s}_{k,t}^m) h_{j,t}^m \quad \text{for } S \in C^m, \quad (12.a)$$

$$\delta v_{ij,n(t)}^m = \eta \alpha l_m(1-l_m) \delta_{j,t}^m h_{j,t}^m (1-h_{j,t}^m) \bar{s}_{i,t} \quad \text{for } S \in C^m, \quad (12.b)$$

$$\delta w_{jk,n^*(t)}^l = -\eta \alpha l_m(1-l_m) v_l (s_{k,t} - \hat{s}_{k,t}^l) h_{j,t}^l \quad \text{for } l \neq m, \quad (12.c)$$

$$\delta v_{ij,n^*(t)}^l = -\eta \alpha l_m(1-l_m) v_l \delta_{j,t}^l h_{j,t}^l (1-h_{j,t}^l) \bar{s}_{i,t} \quad \text{for } l \neq m. \quad (12.d)$$

Training is performed simultaneously all along their optimal paths $(t, n(t))$ and $(t, n^*(t))$.

In eq. (6.a), if $\xi \rightarrow \infty$, then the training formulas of the extreme case are derived. If $x \in C^m$, and $C^{l'}$ is a near-miss class,

$$\delta w_{jk,n(t)}^m = \eta \alpha l_m(1-l_m)(s_{k,t} - \hat{s}_{k,t}^m) h_{j,t}^m \quad \text{for } S \in C^m, \quad (13.a)$$

$$\delta v_{ij,n(t)}^m = \eta \alpha l_m(1-l_m) \delta_{j,t}^m h_{j,t}^m (1-h_{j,t}^m) \bar{s}_{i,t} \quad \text{for } S \in C^m, \quad (13.b)$$

$$\delta w_{jk,n^*(t)}^{l'} = -\eta \alpha l_m(1-l_m)(s_{k,t} - \hat{s}_{k,t}^{l'}) h_{j,t}^{l'} \quad \text{for } l' \neq m, \quad (13.c)$$

$$\delta v_{ij,n^*(t)}^{l'} = -\eta \alpha l_m(1-l_m) \delta_{j,t}^{l'} h_{j,t}^{l'} (1-h_{j,t}^{l'}) \bar{s}_{i,t} \quad \text{for } l' \neq m, \quad (13.d)$$

where $\delta_{j,t}^{l'} = \sum_{k=1}^{k=K} (s_{k,t} - \hat{s}_{k,t}^{l'}) w_{jk,n^*(t)}^{l'}$.

Comparing eq. (13) with eq. (4), we can easily find important differences. Eq. (13.a) and eq. (13.b) represent the cost-weighted gradient descent method for correct class m while eq. (13.c) and eq. (13.d) represent the cost weighted gradient ascent method for near-miss class l' both along the optimal path. The cost $\alpha l_m(1-l_m)=l'_m(d_m)$, a differential form of the sigmoid function. A positive constant α is its steepness factor. We can find easily that the cost reaches a maximum value of $\frac{\alpha}{4}$ at $d_m=0$. So, weights are adjusted in proportion to the value of $l'_m(d_m)$ and the maximum changes in the weights are happened when $d_m=0$, which means that the decision criterion is exactly in the boundary of word class m and word class l'. So, the more confusable those two models are, the more gradient descent training is carried out on the correct model weight parameter set along its optimal path, while the more gradient ascent training is carried out on the near-miss model weight parameter set along its optimal path. Consequently, the discrimination between the correct word model and the near-miss word model will be increased.

Ⅳ. Experimental Results

We have evaluated our new discriminative training algorithm on a data base of ten isolated Korean digits with three versions of each digit pronounced by seven male speakers. Only 50 speech data of five speakers have participated in training and other 160 speech data have been used for test. The speech data were sampled at 10 kHz and analyzed by 25.6 ms frame periods with pre-emphasis and Hamming window. And 12 LPC cepstral coefficients(excluding 0th order) were derived as an input feature vector for each frame. Among ten Korean digits, 3("sam") and 4("sa") are frequently confusable each other.

We have used eq. (12) and NPM among the several predictive neural network models, but this algorithm can be easily applied to the other models without loss of generality. The learning coefficient η is 0.001, and α is 0.1 with 500 iterations.

For the data(A) of the speakers who participated in training and those(B) of the speakers who didn't participate in training, the recognition rates of the conventional training algorithm have scored 97% and 88.3% respectively while those of the proposed training algorithm have scored 99% and 90%, respectively.

Especially, errors between 3("sam") and 4("sa") have been all corrected as we expected. So, our discriminative training algorithm has reduced 30 % of recognition error.

Table 1. Recognition Result.

Data	Conventional Algorithm	Proposed Algorithm
A	97 %	99 %
B	88.3 %	90 %

Ⅴ. Conclusion

In this paper, we proposed a new discriminative training algorithm for predictive neural network models using GPD(Generalized Probabilistic Descent) on the expected recognition error count function derived from MCEF(Minimum Classification Error Forumlation). As a result, we derive new training formulas eq. (12) and eq. (13). The physical meaning of eq. (13) was shortly described in the last part of section III and the property of the cost was analyzed.

As an experimental result, 30% reduction of recognition error has been achieved comparing with the conventional training algorithm. Particularly, the errors observed between two acoustically similiar words 3("sam") and 4("sa") have been all corrected as we expected.

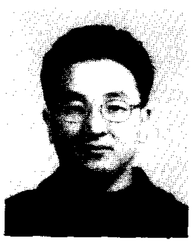
The proposed training algorithm need not to change the network structure at all. It takes roughly N times longer to train the recognizer with the proposed algorithm than with the conventional algorithm if there are N different classes.

There remains much room for the further improvement. Nonuniform weightings on the optimal path by DP can be considered. This technique has already been used in DTW-based recognizer. The weighting function can be adaptively obtained from GPD algorithm. Another possibility is choosing different loss functions. The loss function decides the degree of the cost value that directly participates in the training process.

References

1. K. Iso and T. Watanabe, "Large vocabulary speech recognition using neural prediction model," *Proc. ICASSP-91*, pp. 57-60, 1991.
2. J. Tebelskis, A. Waibel, B. Petek and O. Schmidbauer, "Continuous speech recognition using linked predictive neural network," *Proc. ICASSP-91*, pp. 61-64, 1991.
3. E. Levin, "Hidden control neural architecture modeling of nonlinear time varying systems and its applications," *IEEE Trans. Neural networks*, vol. 4, no. 1, pp. 109-116, 1993.
4. P. C. Chang and B. H. Juang, "Discriminative training of dynamic programming based speech recognizer," *IEEE Trans. Speech and Audio Processing*, pp. 135-143, 1993.
5. W. Chou, B. H. Juang and C. H. Lee, "Segmental GPD training of HMM based speech recognizer," *Proc. ICASSP-92*, pp. 473-476, 1992.
6. B. H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, pp. 3043-3054, 1992.
7. A. Beim and S. Katagiri, "Feature extraction based on minimum classification error/generalized probabilistic descent method," *Porc. ICASSP-93*, pp. 275-278, 1993.
8. P. C. Chang, S. H. Chen and B. H. Juang, "Discriminative analysis of distortion sequences in speech recognition," *Proc. ICASSP-91*, pp. 549-552, 1991.
9. A. Mellouk and P. Gallinari, "A discriminative neural prediction system for speech recognition," *Proc. ICASSP-93*, pp. 533-536, 1993.
10. H. Gish, "A minimum classification error, maximum likelihood, neural network," *Proc. ICASSP-92*, pp. 289-292, 1992.
11. L. R. Rabiner, J. G. Wilpon and F. K. Soong, "High performance connected digit recognition using hidden Markov models," *Proc. ICASSP-88*, pp. 119-122, 1988.
12. J. M. Zurada, *Introduction to artificial neural systems*, west publishing company, 1992.

▲나 경 민 1968년 3월 5일생



1990년 2월 : 서울대학교 전자공학과 졸업
1994년 2월 : 서울대학교 대학원 전자공학과 석사과정 졸업(공학석사)
※주관심분야 : 신경회로망, 음성인식, 패턴분류

▲임 재 열
11권4호(92년 8월 31일) 참조

▲안 수 길
11권4호(92년 8월 31일) 참조