A GPD-BASED DISCRIMINATIVE TRAINING ALGORITHM FOR PREDICTIVE NEURAL NETWORK MODELS

KyungMin NA, JaeYeol RHEEM and SouGuil ANN

Department of Electronics Engineering
Seoul National University
Seoul 151-742, KOREA
Telephone: +82-2-880-7279
Fax: +82-2-882-3906
E-mail: ctlab@krsnucc1.bitnet

ABSTRACT Predictive neural network models are powerful speech recognition models based on a nonlinear pattern prediction. Those models can effectively normalize the temporal and spatial variability of speech signals. 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 based on a generalized probabilistic descent (GPD) algorithm and minimum classification error formulation (MCEF). The Evaluation of our training algorithm on ten Korean digits shows its effectiveness by 40 % reduction of recognition error.

1. INTRODUCTION

Recently, predictive neural network models and their effective training algorithms have been proposed for various speech recognition tasks [1-3]. 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]. In predictive neural network models, an MLP (Multi-layer Perceptron) 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 optimal path from time alignment algorithm. 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) the required amount of training data is relatively small.

However, in spite of the above-mentioned superiority, predictive neural network models suffer from poor discrimination between acoustically similar words. It's because the conventional training algorithm trains each predictor in a word model only with training data of corresponding word while not considering the training states of predictors in other word models.

Instead of conventional error backpropagation (EBP) training algorithm, several discriminative training algorithms for predictive neural network models have been proposed [7-12]. We have proposed an extreme-case discriminative training algorithm based on a generalized probabilistic descent (GPD) algorithm coupled with minimum classification error formulation (MCEF) [12]. The GPD method has proven successful in improving the discrimination powers of the conventional recognizers such as DTW-based recognizer [4] and HMM-based recognizer [5]. The algorithm

directly minimizes an expected recognition error instead of minimizing the accumulated prediction residual. We applied the extreme-case GPD algorithm to the predictive neural network models and derived a new discriminative training algorithm. In this paper, we apply general-case GPD algorithm and derive a training algorithm which trains all models at the same time.

2. NPM AND ITS CONVENTIONAL TRAINING ALGORITHM

2.1 Neural Prediction Model

The MLP predictor outputs a predicted speech feature vector $\hat{S}_t^m = \left(\hat{s}_{1,t}^m, \hat{s}_{2,t}^m, \cdots, \hat{s}_{K,t}^m\right)$ using the preceding input speech feature vectors $S_{t-\tau}, S_{t-\tau-1}, \cdots, S_{t-1}$, where $S_t = \left(s_{1,t}, s_{2,t}, \cdots, s_{K,t}\right)$, if a word is included in C^m (among M word classes $C^i, i = 1,2,\cdots,M$). The symbol τ represents the number of input speech feature vectors for prediction. Let $W_{n(t)}^m = \left(w_{jk,n(t)}^m\right)$ be a weight matrix between the hidden layer and output layer of the $n(t)$-th predictor for a word model m, $V_{n(t)}^m = \left(v_{ij,n(t)}^m\right)$ be a weight matrix between the input layer and hidden layer of the $n(t)$-th predictor for a word m, $H_t^m = \left(h_{j,t}^m\right)$ be an output from the hidden unit at time t, and $f(\cdot)$ be a sigmoid function. Given an optimal path $\left(t, n(t)\right)$, the input-output relation for the MLP predictor is as follows:

$$H_t^m = f\left(\sum_{i=1}^{\tau} V_{n(t)}^m \cdot S_{t-i}\right), \tag{1}$$

$$\hat{S}_t = W_{n(t)}^m \cdot H_t^m. \tag{2}$$

From the predicted speech feature vector $\hat{S}_t$, a prediction residual $\|\hat{S}_t - S_t\|^2$ is calculated. A word model is represented as a sequence of such MLP predictors.

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 accumulated prediction residual $D(m)$.

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

2.2 Conventional BP Algorithm

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

$$\left(w_{jk,n(t)}^m\right)_{q+1} = \left(w_{jk,n(t)}^m\right)_q + \eta(s_{k,t} - \hat{s}_{k,t}) h_{j,t}^m, \tag{4.a}$$

$$(v^m_{ij,n(t)})_{q+1} = (v^m_{ij,n(t)})_q + \eta\delta^m_{j,t}h^m_{j,t}(1-h^m_{j,t})\bar{S}_{i,t}\ , \tag{4.b}$$

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

$$\text{and } \delta^m_{j,t} = \sum_{k=1}^{K}\big(s_{k,t}-\hat{s}_{k,t}\big)\cdot w^m_{jk,n(t)}\ .$$

η is a learning coefficient. I, J, and K are the number of input units, hidden units, and output units, respectively, and $\bar{S}_t = \big(s_{1,t-\tau},\cdots,s_{K,t-\tau},\cdots,s_{K,t-1}\big) = (\bar{s}_1,\cdots,\bar{s}_i,\cdots,\bar{s}_I)$ is an input vector.

3. 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.

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.

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

$$\text{where } D^{\theta}_m\big(t,n(t)\big) = \sum_{k=1}^{K}\big(s_{k,t}-\hat{s}_{k,t}\big)^2$$

$\sum_{t=1}^{T} D^{\theta}_m\big(t,n(t)\big)$ 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 optimal path θ^*.

$$g_m(x,V,W) = \min_{n(t)}\sum_{t=1}^{T} D^{\theta^*}_m\big(t,n(t)\big) \tag{5.b}$$

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

Step 2. A misclassification function $d_m(x,V,W)$ is properly chosen. A general form of this function is shown in eq. (6.a). By controlling the value of ξ, the competing classes that can participate in the process of optimizing the recognizer are determined.

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

In extreme case, if $\xi \to \infty$, the misclassification function of eq. (6.a) becomes eq. (6.b) as $(M-1)^{-\frac{1}{\infty}} \to 1$, where word class n' is the most confusable class to the correct word class m. We have derived a training algorithm based on eq. (6.b) [12]. In this paper, eq. (6.a) is adopted, and a general training algorithm is derived.

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

Step 3. A smoothed loss function is introduced. 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.

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

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.

3.2 New Discriminative Training Formula by GPD Algorithm

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

$$L(x,V,W) = \sum_m l_m(x,V,W) \tag{8}$$

$$V_{t+1}^m = V_t^m + \delta V_t^m, \quad \text{where } \delta V_t^m = -\eta U \nabla l_m \text{ in matrix form,} \tag{9}$$

$$W_{t+1}^m = W_t^m + \delta W_t^m, \quad \text{where } \delta W_t^m = -\eta U \nabla l_m \text{ in matrix form,} \tag{10}$$

where 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.

By applying the probabilistic descent algorithm and combining eq.s (5.a), (6.a) and (7) with (8), (9) and (10), new discriminative training algorithm formulas eq.s (11) are derived.

For $S \in C^m$,

$$\delta w^m_{jk,n(t)} = \eta \alpha l_m (1 - l_m)(s_{k,t} - \hat{s}^m_{k,t}) h^m_{j,t}, \tag{11.a}$$

$$\delta v^m_{ij,n(t)} = \eta \alpha l_m (1 - l_m) \delta^m_{j,t} h^m_{j,t} (1 - h^m_{j,t}) \bar{s}_{i,t}, \tag{11.b}$$

and for all $l \neq m$,

$$\delta w^l_{jk,n^*(t)} = -\eta \alpha l_m (1 - l_m) \nu_l (s_{k,t} - \hat{s}^l_{k,t}) h^l_{j,t}, \tag{11.c}$$

$$\delta v^l_{ij,n^*(t)} = -\eta \alpha l_m (1 - l_m) \nu_l \delta^l_{j,t} h^l_{j,t} (1 - h^l_{j,t}) \bar{s}_{i,t}, \tag{11.d}$$

$$\text{where } \nu_l = \frac{e^{-g_l(x,V,W)\xi}}{\sum_{s, s \neq m} e^{-g_s(x,V,W)\xi}}.$$

Comparing eq.s (11) with eq.s (4), we can easily find important differences. Eq. (11.a) and eq. (11.b) represent the cost-weighted gradient descent method for correct class m while eq. (11.c) and eq. (11.d) represent the cost-weighted gradient ascent method for near-miss class l, both along the optimal paths.

4. EXPERIMENTS

We have evaluated our new discriminative training algorithm on a data base of ten isolated Korean digits with one version of each digit pronounced by 20 male speakers. Only 80 speech data have participated in training and other 120 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 0-th order) were derived as an input feature vector for each frame.

We have used NPM among the several predictive neural network models, but this algorithm can be easily applied to the other models without loss of generality. The NPM is trained with 300 iterations. Learning rate was 0.03. Only 20 iterations were performed with $\alpha = 0.01$ and $\eta = 0.01$ for the proposed algorithm. The proposed algorithm has achieved about 40 % reduction of recognition error.

Conventional Algorithm	Proposed Algorithm
93.3 % (112/120)	95.8 % (115/120)

Table 1. Recognition Result

5. CONCLUSION

In this paper, we proposed a new discriminative training algorithm for the predictive neural network models using the GPD method on the expected recognition error count function derived from the

MCEF. As a result, we derive new training formulas eq. (11). As experimental results, 40 % reduction of recognition error has been achieved comparing with the conventional training algorithm. The proposed training algorithm needs 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 further improvement. Nonuniform weightings on the optimal path by the DP can be considered. This technique has already been used in the DTW-based recognizer. The weighting function can be adaptively obtained from the GPD algorithm. Another possibility is choosing the other 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. He Jun and H. Leich, "A discriminative training algorithm for speech recognizer based on the neural prediction model," *Proc. EUSIPCO-92*, pp. 423-426, 1992.

8. Y. D. Liu, Y. C. Lee, H. H. Chen and G. Z. Sun, "Discriminative training algorithm for predictive neural network models," *Proc. IJCNN-92*, pp. 685-690, 1992.

9. A. Mellouk and P. Gallinari, "A discriminative neural prediction system for speech recognition," *Proc. ICASSP-93*, pp. 533-536, 1993.

10. K. Iso, "Speech recognition using dynamical model of speech production," *Proc. ICASSP-93*, vol. 2, pp. 283-286, 1993.

11. B. Petek and A. Ferligoj, "Exploiting prediction error in a predictive-based connectionist speech recognition system," *Proc. ICASSP-93*, vol. 2, pp. 267-269,1993.

12. K. M. Na, J. Y. Rheem and S. G. Ann, "A Discriminative training algorithm for predictive neural network models," will appear in *Proc. ISCAS-94*, May 1994, London, UK.