A NEW HYBRID HMM/ANN MODEL FOR SPEECH RECOGNITION

Xiaojing Xi[1] , Kunhui Lin[1], Changle Zhou[2],Jun Cai[2]
1. Software School of XiaMen University, China
2. Department of Computer Science XiaMen University, China

Abstract: Because of the application of the Hidden Markov Model (HMM) in acoustic modeling, a significant breakthrough has been made in recognizing continuous speech with a large glossary. However, some unreasonable hypotheses for acoustic modeling and the unclassified training algorithm on which the HMM based form a bottleneck, restricting the further improvement in speech recognition. The Artificial Neural Network (ANN) techniques can be adopted as an alternative modeling paradigm. By means of the weight values of the network connections, neural networks can steadily store the knowledge acquired from the training process. But they possess a weak memory, not being suitable to store the instantaneous response to various input modes. To overcome the flaws of the HMM paradigm, we design a hybrid HMM/ANN model. In this hybrid model, the nonparametric probabilistic model (a BP neural network) is used to substitute the Gauss blender to calculate the observed probability which is necessary for computing the states of the HMM model. To optimizing the network structure in and after the training process, we propose an algorithm to prune hidden nodes in a trained neural network, and utilize the generalized Hebbian algorithm to reconfigure the parameters of the network. Some experiments show that the hybrid model has a good performance in speech recognition.

Key words: HMM; ANN; BP; removing hidden nodes algorithm; generalized Hebbian algorithm

1. INTRODUCTION

Speech recognition is mainly to let the machine understand what human says which means under various situations, the machine can recognize the content of the speech accurately, and then execute the various intentions of the human according to the information. HMM has made great success in every field of speech processing, but it has many restrictions. The neural network memories and stores the information for a long time well by using the weight values, but the memory ability of instantaneous responding to the input model is weaker. So, it has difficulty to model the time variable. As both the HMM and the ANN have advantages and disadvantages, so we propose the hybrid HMM/ANN model to which has been attached importance by many research institutions, such an ICSI, SRI, SVR and so on. The performances of some speech recognition systems researched by them have overmatched the traditional HMM system. In this paper, we apply BP neural network to calculate the observation probability needed by the states of the HMM instead of the GM. In this hybrid model, we apply Continuous Destiny Hidden Markov Model (CDHMM) to build modeling for the short-time speech components, and calculate the observation probability of the CDHMM by using the ANN with the ability to classify the signal. In addition to, for optimization of the hybrid model, we proposed an algorithm to remove hidden nodes in a trained neural network, and optimize the parameters of the network by using the generalized Hebbian algorithm.

2. THE HYBRID HMM/ANN MODEL

The speech signal is a typical dynamic pattern sequence, the time relativity of the frame and the neighbor frame is very strong, so, if we want to apply the ANN to speech recognition, we must resolve the memory problem of instantaneous outputs. So we combine the ANN to the developed method HMM and form a Hybrid system, in which ANN may preprocess the data as former-end of HMM, or rear-process as back-end of HMM.

2.1 The combination of the HMM and the ANN

The combination of ANN and HMM has many ways, such as realizing the HMM using the ANN directly, the combination of the speech frame layer, the combination of the speech layer, the combination of the tone-phase layer, the combination of the sub-layer and so on. In this paper, we search after a new hybrid model instead of realizing the HMM by the ANN directly, the new hybrid model can optimize the HMM model and also can make use of the advantages of each technology: the time modeling of the HMM and the acoustic modeling of the ANN, specially, we calculate and estimate the observation probability by the ANN.

There are many methods to design and train the neural network. The simplest way is that we map the speech vector of one frame into observation probability; the network is trained from one frame to the next frame. This is the training for the network of the frame layer.

In the method of tone-phase layer, the input of the network comes from the whole speech phase instead of a speech frame or a fixed speech window. So it can utilize the correlation of the whole speech frames in the speech phrase better, and also we can use other information easily, for example the time-length, but we must divide the speech into speech phase firstly, and then the neural network can calculate the divided speech phase. Among these different combination methods, the experiment shows that the combination of the frame layer fits the characteristics of ANN and HMM better.

2.2 The application and algorithm of the hybrid model

In hybrid model, the acoustic modeling is completed by the ANN, the time-field modeling depends on the traditional HMM, and there are two kinds of methods to finish the acoustic modeling:

The first method is the predictive network, the input is the feature vectors of several continuous frames, the output is the prediction value of the next frame, we distribute a predictive network for every phoneme, and we select the network with minimal predictive error by comparing the predictive errors of the network of every phoneme matching with the currently speech segment, so it embodies the time-field correlation of the neighbor speech frames.

The second method is classifying network. In this network, the input is still the feature vectors of several continuous frames, but the output is mapped directly into the states of HMM. The n output nodes represent n sorts, the input nodes are mapped into one sort among the n sorts. The classifying network is easy, intuitionistic and distinguishable in essence, modularity in design, and it can organize bigger system, and has the advantages of prefect mathematical explanations, so it can be integrated into the statistical recognition frames easily.

In this paper, the hybrid model that is unlike the traditional HMM/ANN model adopts the classified network to estimate the posterior probability. So we may adopt context-sensitive input pattern as the input of the neural network, it takes the time correlation of the speech vectors into account. The posterior probability of the output of the neural network is $p(q_i \mid X_{t-d}^{t+d})$ while the likelihood probability destiny needed by the states of the HMM is $p(x_t \mid q_i)$, according to the Bayes rules, we deduce the measure likelihood destiny probability from the posterior probability $p(q_i \mid X_{t-d}^{t+d})$,

$\frac{p(x_t \mid q_i)}{p(x_t)} = \frac{p(q_i \mid x_t)}{p(q_i)}$, in the recognition, because $p(x_t)$ is the same to all paths, so measure likelihood function doesn't influence the recognition result. Because the classifying network embodies the essence of the hybrid model, we utilize it to constitute the speech recognition system. When the hybrid model recognizes the speech that is calculated by ANN scale observation probability of the states of HMM, The course is divided into two steps: The first step: computing all states' scale observation probability at t;

The second step: calculating the accumulating probabilities of the active paths, removing the superfluous nodes according to these probabilities and then ensuring the active paths at t

For every frame speech vector, we may calculate the output vector $g_i(x_i)$ 、 $g_{j|i}(x_t)$ 、 $u_{i,j}(x_t)$ of every node network (BP network)by posteorder traversal, when traverse to the root node, we can get the output collective,

$$u(x_i)=\sum_i g_i(x)\sum_j g_{j|i}(x_t)*u_{i,j}(x_t)$$

The output collective is the estimation of the posterior probability. That the posterior probability divided by the transcendent probability is the measure likelihood probability $\frac{p(q_i \mid X_{t-c}^{t+c})}{p(q_i)}=\frac{p(X_{t-c}^{t+c} \mid q_i)}{p(X_{t-c}^{t+c})}$

In the algorithm of frame synchronization in Viterbi path searching, for every frame speech vector, we may add up the measure observation probability to the path probability of every active path

$$a_{\lambda_m,q_k}(x_t)=\max_{i\in pre(q_k)}\{a_{\lambda_m,q_k}(x_{t-1})+\log(a_{i,k})+\log(p(q_i \mid x_t))-\log(p(q_i))\}$$

$a_{\lambda_m,q_k}(x_t)$ is the optimum path accumulative probability, when reaching the state q_k of the model λ_m at t, pre(q_k) is the set including all the prior nodes in the searching networks, $a_{i,k}$ is the state transition probability from the prior node i to the node j. According to the computing structure of the accumulative probability of the currently active path, we may remove some paths whose accumulative probability below a threshold, and then, get the every speech vector frame of active paths at t+1, and continue according to the above steps until the end of the pronounce sentences. At last, we trace back according to the optimum path until we find the model sequence of HMM matching to the pronounce sentence, and the result is the preferred result of the acoustics model.

2. 3 Optimization of the hybrid model

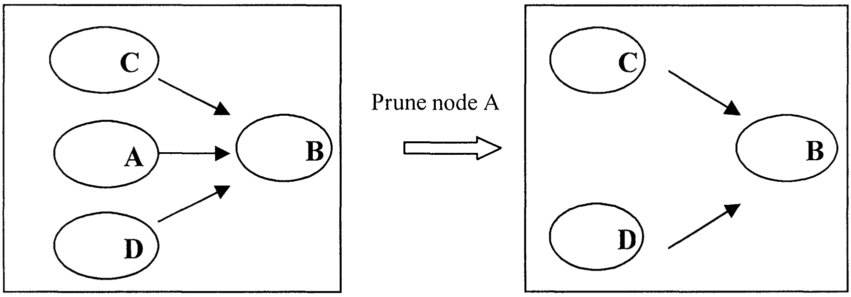
2.3.1 Optimization of the number of hidden nodes

The first problem needed to be considered is how to confirm the number of the hidden nodes

1. The increasing algorithm of hidden nodes: By increasing the number of hidden nodes to improve the frame recognition ratio during the training.
2. The removing algorithm of hidden nodes: By removing the superfluous hidden modes and the weight values during the training.

After comparing and analyzing, we put forward a new method to confirm the number of the hidden nodes.

1. Getting the number of the clustering centers by analyzing the self-organizing data iteratively, and assigning a hidden node for a pair of clustering centers in different sorts. So to the input mode these hidden nodes form multidimensional space. In this space, the input nodes form decision-making curved surface easily, so we get the number of the hidden nodes N suited not only for training but the removing after training.
2. Training the BP network with n hidden nodes



Figl A novel algorithm for hidden nodes pruning

. Removing the redundancy nodes by iteration algorithm, and then adjust the weight value of the remainder nodes on condition that keeping the intrinsic input, at last we get the optimum nodes number of the network, as the Figl shows, after removing the node A, we adjust the weight value of the remainder nodes to make the pure input unchanged defined by the least-mean-square, i.e. as for all the models in the training set

$$\sum_{j\in\{A,C,D\}} w_{j,B}y_j(n) = \sum_{j\in\{C,\ D\}} (w_{j,B} + \delta_{j,B})y_j(n) \quad \forall n \in \{1,2...N\}$$

The $w_{j,B}$ is the weight value from the node j to B, $\delta_{j,B}$ is the residual error from the node j to B, $y_j(n)$ is the output of the node n-th, which is equivalent to a linear equation

$$\begin{bmatrix} y_C(1) & y_D(1) \\ M & M \\ y_C(N) & y_D(N) \end{bmatrix} \begin{bmatrix} \delta_{C,B} \\ \\ \delta_{D,B} \end{bmatrix} = w_{A,B} \begin{bmatrix} y_A(1) \\ M \\ y_A(N) \end{bmatrix}$$

With the LMS iterative algorithm we can get the optimized result of the linear equation under the least-mean-square condition. Because LMS algorithm resolves the above equation, the remained difference decreases with the decreasing of the iterative times monotonously, we can determine which nodes should be removed by calculating the initial remained difference of every node, the computing amount is very small, and the checking of the redundancy nodes is very easy.

2.3.2 The second improvement: the initialization of the weight parameters of network

Before training the network, every weight should be assigned a value. It is an important issue in ANN, which affects directly to the convergence of the training results. The main idea of the training is seeing out the study mechanism to initialize the weight values of Network using transcendental knowledge. In this paper, these classes are divided by the neural network, we utilize the Hebb algorithm to init the weight values, and the experiments show the good results.

Its principle: we set the hidden nodes as linear, as for the input mode x_i $i = 1,2,\ldots,k-1$,the weight value matrix brought by between the input nodes and the hidden nodes is $V_{k-1} = [v_{1,k-1}, v_{2,k-1}, v_{3,k-1}, \ldots, v_{n_k,k-1}]$, for x_k we update the weight matrix according to the Oja rule

$V_k = V_{k-1} + a_0(I - V_{k-1}V_{k-1}^T)x_k x_k^T V_{k-1} = V_{k-1} + a_0(x_k - V_{k-1}\bar{h}_{k-1}^T)\bar{h}_k^T$,in this formula, $\bar{h}_k = V_{k-1}^T x_k$, every weight vector is given by the following formula $v_{i,k} = v_{i,k-1} + a_0 \bar{h}_{i,k}(x_k - \sum_{i=1}^{n_k} \bar{h}_{l,k} v_{i,k-1})$

For the generalized Hebbian study rule, in the formula

$$V_k = V_{k-1} + a_0(x_k - V_{k-1}L_d(\bar{h}_k^T))L_h(\bar{h}_k^T)$$

$L_h(\bar{h}_k^T) = [L_h(\bar{h}_{1,k}), L_h(\bar{h}_{2,k}), \ldots, L_h(\bar{h}_{n_k,k})]$ is the function of the output $\bar{h}_k$.Every weight vector $v_{i,k}$ can be gotten as follows:

$$v_{i,k} = v_{i,k-1} + a_0 L_h(\bar{h}_{i,k})(x_k - \sum_{i=1}^{n_k} L_d(\bar{h}_{i,k}) v_{i,k-1})$$

The stop rule: if $n_k > n_i$, the stop rule is based on the decrease of error

$$E_v \qquad E_v = \sum_{k=1}^{m} \left\| \overline{x}_k - \sum_{i=1}^{n_k} L_d(\overline{h}_{l.k}) v_i \right\|^2, \overline{h}_k = V^T x_k$$

The form of the learning function can be defined as $\varphi(z) = \frac{d\rho^2(z)}{dz}$, $\rho(.)$ is the inspiring function, the initialization of the parameters of network can be divided into two parts, one is the weight values initialization between the input nodes and hidden nodes and the other is the weight values initialization between the hidden nodes and the output nodes. At first we initialize the connective weight value between the input nodes and the output nodes by using the generalized Hebbian algorithm and then init the connect weight values of the output layer by the supervisory training algorithm $v_i, i = 1,2,..., n_k$, the detailed process is as follows:

1. Initialize V with random values; Select $L_d(.)$、$L_h(.), a_0, \beta$; Set v=1;
2. Calculate $v_i, i = 1,2,..., n_k$, $\overline{h}_k = V_{k-1}^T x_k, v_i, i = 1,2,..., n_k$ and $\overline{x}_k = \sum_{i=1}^{n_k} L_d(\overline{h}_{i,k}) v_i$ according to the formula for every k=1， 2， …m
3. Determine the stop rule by using the previous phase
4. if v>1 then $E_v^{rel} = \frac{E_v^{old} - E_v}{E_v^{old}}$ else set $E_v^{old} = E_v$;
5. if v=1 or $E_v^{old} > E_v$ then set v=v+1 else loop to the step 2

And then using the supervisory training algorithm to initialize the connective weight values of the output layer $w_i, i = 1,2,..., n_0$, and initialization the weight values and using the sample couple $(y_k, h_k^3), k = 1,2,..., m$ to initialize the weight values between the hidden layer and output layer. The whole process is divided into three parts as follows: A. Generalized training rule:

$$\lambda E + (1-\lambda)E' = \lambda \sum_{k=1}^{m} \sum_{i=1}^{n_0} \phi_2(e_{i,k}) + (1-\lambda) \sum_{k=1}^{m} \sum_{i=1}^{n_0} \phi_1(e_{i,k}) \text{ in this}$$

formula, $\phi_2(e_{i,k}) = \frac{1}{2} e^2{}_{i,k}$ $\lambda : 1.0 -> 0.0, \lambda = \lambda(E) = \exp\left(-\frac{\mu}{E^2}\right)$, if the output of the network is Binary value, and the value is +1 or -1 then $\phi_1(e_{i,k}) = y_{i,k}(y_{i,k} - y_{i,k}^3)$

B. Updating the weight values basing on the grads decreasing

$w_{i,k} = w_{i,k-1} + \alpha\delta^{0}_{i,k} h^{\exists}_{k}$

C. The program for initializing the connective weight of the output layer

(1) Initialize W with random values

Select α, μ， Set $\lambda = 1, E_w = 0, \text{and } v = 1$

(2) For each k=1， 2， …,m Calculate $y^{\exists}_{i,k} = \sigma(w_i^T h^{\exists}_k), i = 1,2,\ldots,n_0$,

Evaluate $\delta^{0}_{i,k}, i = 1,2,\ldots,n_0$ Update $w_{i,} i = 1,2,\ldots,n_0$

Calculate $y^{\exists}_{i,k} = \sigma(w_i^T h^{\exists}_k), i = 1,2,\ldots,n_0$,

Set $E_w = E_w + \frac{1}{2}\sum_{i=1}^{n_0}(y_{i,k} - y^{\exists}_{i,k})^2$ Calculate

$\lambda = \lambda(E) = \exp(-\frac{\mu}{E^2})$ if v>1 then $E_w^{rel} = \frac{E_w^{old} - E_w}{E_w^{old}}$ $E_w^{old} = E_w$

If v=1 or $E_w^{old} > E_w$; then set v=v+1 and go to (2)

3. EXPERIMENT RESULT

The experiment shows that the recognition performance of the hybrid HMM/ANN excels the traditional HMM with the same number of parameter and the input characters. In order to implement the same recognizing performance, the HMM system has to use more parameters and more complex model structure. As for the same continuous speech database, the recognition error ratio of the Context-free traditional HMM for the testing sentences is 11%, using MLP output as the posterior probability estimation and with almost the same number of parameters, the error ratio is 5.8%, while using the hybrid model preferred in this paper, the error ratio is 4.1%. Meanwhile, the hybrid model has acquired upstanding effect in unspecific person recognition of and the key words detection.

REFERENCES

[1] Xiongwei Zhang, Liang Chen, and Jinbin Yang Modern Speech technology and applications, China Machine Press 2003.8 ISBN.7-1111-12795-1 219-222

[2] Jinhui Xie, Hidden Markov Model and its applications in speech processing, HuaZhong University Press, 1995.4 ISBN 7-5609-1094-7/TN.34 103-113

[3] Changning Huang, Ying Xia, Monograph of speech information processing [A], TSingHua University Press, 1996.4 ISBN7-302-01929-0/TP.879 489-508

[4] Tingyue Zhuang, Yunhe Pan and Fei Wu, Web-based Multimedia Information Analysis and Retrieve TsingHua Unversity Press 2002.9 ISBN 7-302-05584-X/TP.3299 122-272