應用改良式逆傳遞做語音辨識

陳志良 郭志鳴 洪英真
高雄工學院電子工程系

摘要

本文提出一改良式的逆傳遞網路訓練演算法，基本構想是多賦予每個運算神經元三個類似連線權位的可調參數，使其在訓練過程中更具自主性。此演算法不僅可加快訓練速度，且能降低訓練過程陷入局部最小值的機會。最後依此演算法訓練的三層認知器用以測試小字彙量的中文字語音辨識，結果語者相關的部份可達90到99%的正確率；在語者無關的部份亦有80%以上的正確率。

（關鍵字：語音辨識；類神經網路；多層認知器）

Applying Improved Back-Propagation to Speech Recognition

Chih-Liang Chen, Jyh-Ming Kuo and Ying-Cheng Horng
Department of Electronic
Kaohsiung Polytechnic University

ABSTRACT

Based on the idea of using heterogeneous processing units (PUs) in a network, a variation of Back-Propagation (BP) learning algorithm is presented. Three parameters, which are adjustable like connection weights, are incorporated into each PU to increase its autonomous capability by enhancing the activation function. The improved BP training algorithm thus is developed for updating the three free parameters as well as connection weights. The improved BP is intended not only to speedup the training procedure, but also to reduce the occurrence of local minima. The algorithm has been intensively
tested on the speech recognition problem. Results show that the performance of correct rates can reach 90 to 99% for the speaker-dependent case; The performance for speaker-independent case also is above 80%.

(KEY WORDS: Speech Recognition; Neural Networks; Multilayered Perceptron)

I. INTRODUCTION

By introducing an improved Back-Propagation (BP) training algorithm, we intended to alleviate the training problems associated with the standard Back-Propagation (BP) [1]: the slow training speed the presence of local minima. The key idea is to incorporate three free parameters into each PU so that each PU is able to change its output range and the slope of sigmoid function.

The improved BP training algorithm, which is a simple extension of the standard BP, will be described first. Although several equations were developed, they are very easy to implement, because only little more computation complexity is introduced. Then, we describe the methods of computing the Cepstrum parameters that capture the feature of speech signals. The derived Cepstrum parameters finally are used as input format for training three-layer Perceptron to recognize the isolated Chinese characters.

II. TRAINING ALGORITHM

To describe the improved BP training algorithm, the following notations are adopted:

- \( RMS \) Root Mean-Square error
- \( net_j \) total input to processing unit (PU) \( j \)
- \( w_{ij} \) weight of connection between PU \( i \) and PU \( j \)
- \( d_{j}^{(p)} \) desired output of PU \( j \) when training pattern \( p \) is applied
- \( y_{j}^{(p)} \) actual output of PU \( j \) when training pattern \( p \) is applied
- \( \delta_{j} \) error signal for PU \( j \)
- \( \eta \) learning rate
- \( \alpha \) momentum

The objective is to minimize the Root-Mean-Square-error \( (RMS) \) over all training patterns by making adjustments on connection weights and other introduced free parameters.

\[
RMS = \frac{1}{P} \sum_{p=1}^{P} \left[ \frac{1}{M} \sum_{j=1}^{m} \sqrt{(d_{j}^{(p)} - y_{j}^{(p)})^2} \right]
\]

(1)

where \( P \) is the total number of training patterns, \( M \) is the number of output units.

A. Forward Computing Phase
In the forward computing phase, each PU computes the weighted sum of inputs and weights by

$$net_j = \sum_l w_{lj}y_i + \theta_j$$

(2)

In the standard BP, this weighted sum is transferred through a nonlinear, differentiable sigmoid function so as to produce the actual output. However, based on idea of increasing the autonomous capability for each PU in a network, we modify the activation function as follows:

$$y_j = f(net_j, U_j, L_j, T_j) = \frac{U_j - L_j}{1 + e^{-net_j/T_j}} + L_j$$

(4)

where $U$ and $L$ represent the upper bound and the lower bound of the activation function, respectively, i.e., the output ranges ($L$, $U$). The temperature $T$ represents the slope of the activation function. Note that we want to make the three adjustable variables ($U$, $L$, $T$) adjustable --- similar to the connection weights. Therefore, for each PU, a new training rule for each PU must be developed. Not only the rule can update the connection weights, but also can update the three adjustable variables such that each PU is able to adjust its maximum output, minimum output, and temperature as necessary.

B. Backward Updating

Rumelhart et al [1][2] suggested using gradient descent to perform steepest descent on a surface in weight space whose height at any point in weight space is equal to the error measure, i.e., the adjustment of weights is proportional to the first derivative of the output function in each PU. Therefore, the first derivative with respect to the weight space is derived as Eq. (5a). Likewise, the first derivatives with respect to the three more adjustable variables are derived as Eq. (5b-5d).

$$f_{net,j}^\prime = \frac{\partial f(net_j, U_j, L_j, T_j)}{\partial net_j} = \frac{-(y_j - U_j)(y_j - L_j)}{T_j(U_j - L_j)}$$

(5a)

$$f_{U,j}^\prime = \frac{\partial f(net_j, U_j, L_j, T_j)}{\partial U_j} = \frac{1}{1 + e^{-net_j/T_j}}$$

(5b)

$$f_{L,j}^\prime = \frac{\partial f(net_j, U_j, L_j, T_j)}{\partial L_j} = 1 - \frac{1}{1 + e^{-net_j/T_j}} = 1 - f_{U,j}^\prime$$

(5c)

$$f_{T,j}^\prime = \frac{\partial f(net_j, U_j, L_j, T_j)}{\partial T_j} = \frac{net_j(y_j - U_j)(y_j - L_j)}{T_j^2(U_j - L_j)} = \frac{-net_j}{T_j} f_{net,j}^\prime$$

(5d)

The error signals $\delta$ with respect to net, $U$, $L$, $T$ for output units, which have specific desired outputs, are given by:

$$\delta_{net,j} = f_{net,j}^\prime(d_j - y_j)$$

(6a)

$$\delta_{U,j} = f_{U,j}^\prime(d_j - y_j)$$

(6b)
\[ \delta_{l_j} = f'_j (d_j - y_j) \] \hspace{2cm} (6c)  
\[ \delta_{T_j} = f'_j (d_j - y_j) \] \hspace{2cm} (6d)

However, since specifically desired outputs for hidden units are not available, the above equations become useless for hidden units. Just as the name "back-propagation" implies, the error signal for each hidden unit, as shown in Eq. (7a), is determined recursively in terms of the error signals of the units to which it directly connects and the weights of those connections. Likewise, the error signals with respect to the \( U, L, T \) for hidden units are the accumulated back-propagated values from the connected units as shown in Eq. (7b-7d).

\[ \delta_{net} = f'_{net} \sum_k \delta_{net_k} w_{jk} \] \hspace{2cm} (7a)  
\[ \delta_{u_j} = f'_{u_j} \sum_k \delta_{u_k} w_{jk} \] \hspace{2cm} (7b)  
\[ \delta_{l_j} = f'_{l_j} \sum_k \delta_{l_k} w_{jk} \] \hspace{2cm} (7c)  
\[ \delta_{T_j} = f'_{T_j} \sum_k \delta_{T_k} w_{jk} \] \hspace{2cm} (7d)

With the error signals calculated in backward-computing phase, each PU adjusts its connection weights \( w \), bias \( \theta \), upper-bound \( U \), lower-bound \( L \), and temperature \( T \) by:

\[ \Delta w_{y} (n+1) = \eta_{net} \delta_{net_i} y_i + \alpha_{net} \Delta w_y (n) \] \hspace{2cm} (8a)  
\[ \Delta \theta_j (n+1) = \eta_{net} \delta_{net} + \alpha_{net} \Delta \theta_j (n) \] \hspace{2cm} (8b)  
\[ \Delta U_j (n+1) = \eta_{u_j} \delta_{u_j} + \alpha_{u_j} \Delta U_j (n) \] \hspace{2cm} (8c)  
\[ \Delta L_j (n+1) = \eta_{l_j} \delta_{l_j} + \alpha_{l_j} \Delta L_j (n) \] \hspace{2cm} (8d)  
\[ \Delta T_j (n+1) = \eta_{T_j} \delta_{T_j} + \alpha_{T_j} \Delta T_j (n) \] \hspace{2cm} (8e)

Where \( \eta_{net} \) is a learning rate similar to the step size in gradient search algorithm. Rumelhart also suggested including a momentum term \( \alpha_{net} \), which determines the effect of past weight changes on the current direction of movement in weight space, to increase the training speed without leading to oscillation. Likewise, the \( U, L, \) and \( T \) are given their own learning rates and momentums.

**III. SAMPLING AND PREPROCESSING**

The time-domain voice input from microphone first is sampled by the Sound Blaster card and recorded. The sampling rate is set 10 KHz to capture the most features of voice.

The recorded data then are segmented as a voice frame by removing the fractions of silence or noise. Both the zero-crossing rate (ZCR) and energy are used to determine the beginning and ending of a single pronounced Chinese word. If the ZCR greater than a preset value, say 5, it could be a beginning or ending. The energy of a discrete-time signal in a short time duration is calculated by
\[ E(n) = \sum_{m=0}^{N-1} [W(m)x(n-m)]^2 \]  

(9)

where \( W(m) \) is a moving window for \( N \) samples. If \( E(n) \) smaller than a critical value, e.g., 30, it must be a silence duration or background noise.

### IV. FEATURE EXTRACTION

Since the data in the derived time-domain voice frame are too large to handle, the \( 16 \times 10 \) Cepstrum parameters, which are derived by the following processes, are used as feature of voice signal.

#### A. Pre-emphasis and Hamming Window

The derived voice frame is pre-emphasized by the following transfer function, because the voice gain tends to decrease 6 dB/oct in the part of high frequency (1 to 5KHz) \[3\].

\[ H(z) = \frac{(Z - \alpha)}{Z} \]  

(10)

where \( \alpha \) ranges \((0.9, 1)\), we use 0.95. Then, the derived voice frame is further multiplied by the following Hamming window to smooth the irregular parts of voice signals while reserving the fundamental format.

\[ W(n) = \begin{cases} 
0.54 - 0.46 \cos\left(\frac{2n\pi}{(N - 1)}\right) & 0 \leq n \leq N - 1 \\
0 & \text{else} 
\end{cases} \]  

(11)

#### B. Autocorelation

The autocorelation of discrete-time signals is defined as

\[ \phi(k) = \sum_{n=-\infty}^{\infty} x(n)x(n+k) \]  

(12)

However, voice signal is not static signal, the definition of short-time duration autocorelation must be applied \[4\]:

\[ \phi_n(k) = \sum_{n=-\infty}^{\infty} x(n+i)x(n+i+k) \quad 0 \leq k \leq N - 1 \]  

(13)

that is, autocorrelation analysis is started from sample \( i \) with length \( n \).

#### C. Linear Predict Coding

The LPC is an efficient speech analysis technique for computing signal coefficients. The Durbin Recursive Procedure \[4\] is used to compute the following LPC parameters \( \{ \alpha_k \} \):

\[ \sum_{k=1}^{p} \alpha_k \phi_n(i-k) = \phi_n(i) \quad 1 \leq i \leq p \]  

(14)

where \( \phi_n(i) \) is short-time duration autocorelation function
D. Cepstrum

Finally, a set of Cepstrum parameters, which is known as an effective format for speech recognition[4], is defined as:

\[
C(r) = F^{-1}\{\log|X(k)|\} = \frac{1}{N} \sum_{k=0}^{N-1} \log|X(k)|e^{j2\pi k r/N} \quad 0 \leq n \leq N-1
\]  

(15)

where \(X(k)\) is the Fourier Transform of the time-domain speech signal \(x(n)\). \(F^{-1}\{\}\) is the inverse Fourier Transform.

The derived LPC parameters \(\{\alpha_k\}\) then is applied to compute the Cepstrum parameters \(C(n)\) by the following recursive function:

\[
C(n) = \alpha(n) + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) C(n) \alpha(n-k) \quad 1 \leq n
\]  

(16)

V. SIMULATIONS AND RESULTS

A three-layer Perceptron has been used for this study, since it has been proved as a universal approximator provided that there are sufficient number of hidden units [5-7]. The input layer is made up of \(16 \times 10\) units. The \(16 \times 10\) Cepstrum parameters then are fed into the corresponding input units. Ten output units are used to indicate the corresponding ten isolated speaking Chinese words (0 to 9), which are used for this experiment.

Three experiments have been made: (A) single-speaker, speaker-dependent, (B) multiple-speaker, speaker-dependent, and (C) multiple-speaker, speaker-independent. Table 1 shows that the correct rate is increased near 100\% by providing more training patterns for the case of single-speaker, speaker-dependent. For the case of multiple-speaker, speaker-dependent, the correct rate is lesser than the case (A) as expected. Also, increasing the number of hidden units does not significantly improve the performance as shown in Table 2. Finally, for the case of speaker-independent, as usual, Table 3 shows the correct rate is not as good as speaker-dependent cases. Further, increasing the training time to reduce the convergence criteria RMS do not improve the performance either.

| Table 1. Single-speaker, speaker-dependent with different training patterns |
|-----------------------------|---|---|---|
| Training patterns | 10 | 15 | 20 |
| Correct Rate(\%) | 94.00 | 96.32 | 99.30 |

<p>| Table 2. Multiple-speaker, speaker-dependent with different hidden units |
|-----------------------------|---|---|---|
| Hidden units | 12 | 14 | 16 |
| Correct Rate(%) | 92.42 | 91.17 | 92.20 |</p>
<table>
<thead>
<tr>
<th>RMS</th>
<th>0.04</th>
<th>0.035</th>
<th>0.03</th>
<th>0.025</th>
<th>0.023</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Rate(%)</td>
<td>78</td>
<td>81</td>
<td>81</td>
<td>80</td>
<td>82</td>
</tr>
</tbody>
</table>

**VI. CONCLUSION**

An improved BP training algorithm for Multilayered Perceptron (MLP) has been described. We have examined this variation on the speech recognition problem. The performance for small vocabulary, speaker-dependent case can reach 90%; The performance for speaker-independent case also is above 80%. In summary, MLP has potential for these kinds of problem. Applying the proposed improved BP training algorithm, the training speed is improved by incorporating the adjustable upper and lower bounds with small values of corresponding learning rates.

**REFERENCES**


