RECURRENT NEURAL NETWORKS FOR SPEECH MODELING AND SPEECH RECOGNITION

Tan LEE †, P.C. CHING †, L.W. CHAN ††

† Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
†† Department of Computer Science, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
Email : pcching@ee.cuhk.hk

Abstract

This paper describes a new method of utilizing recurrent neural networks (RNNs) for speech modeling and speech recognition. For each particular speech unit, a fully connected recurrent neural network is built such that the static and dynamic speech characteristics are represented simultaneously by a specific temporal pattern of neuron activation states. By using the temporal RNN output, an input utterance can be represented as a number of stationary speech segments, which may be related to the basic phonetic components of the speech unit. An efficient self-supervised training algorithm has been developed for the RNN speech model. The segmentation for input utterances and the statistical modeling for individual phonetic segments are performed interactively in this training process. Some experimental results are used to demonstrate how the proposed RNN speech model can be used effectively for automatic recognition of isolated speech utterances.

1 Introduction

Artificial neural networks (ANN) have been applied to automatic speech recognition in two different approaches [1]. Conventional neural network models are widely used to classify static patterns which are extracted from pre-segmented speech. However, this pre-segmentation procedure could be erroneous since the segment boundaries are not clearly defined in many cases. This prevents the static pattern classification method from attaining satisfactory recognition performance. Furthermore, since only stationary properties of individual segments are considered, the static model is in incomplete as many useful temporal features of the speech signal are ignored. More recently, neural network models with delay connections or recurrent connections have been introduced to deal with the time-varying nature of speech signals [2][3]. Short-time acoustic features are presented to the dynamic neural networks sequentially and the recognition is based on temporal integration of the output sequence. In other words, time is introduced as an additional dimension of input feature space. To cope with the increased complexity, a dynamic model usually requires a much greater network size than a static one and is, therefore, limited to small vocabulary applications.

In this paper we describe a novel approach of utilizing recurrent neural networks for speech modeling and speech recognition. The One-Class-in-One-Network (OCON) architecture [4] is employed. For each speech unit that corresponds to a class of speech patterns with similar acoustic properties, an RNN based dynamic model is built by a well designed training process. In this approach, the speech unit is modeled as an unseparable entity and pre-segmentation procedure is not required. The speech unit referred here may be a syllable, a word or even a sentence. It can be divided into a set of acoustically stationary segments that are concatenated in the time domain [5]. In the proposed RNN speech model, the occurrence of each of these segments is represented by the activation of a particular output neuron. The temporal relationships of adjacent segments are characterized by a specific sequential order governing the changes of neuron activation states. Based on the sequential output of the RNN, a temporal error function is defined and is used as a distance measure between an input utterance and the speech model. For speech recognition applications, this error function will be used as the major discrimination factor between different models.

2 Description of an RNN Speech Model

A fully connected RNN, as shown in Figure 1, is adopted to model a particular speech unit Γ. The activation level of the nth neuron $y_n(t)$, is given by the following difference equation,

$$y_n(t) = f_n\left[\sum_{l=1}^{N} w_{nl} y_l(t-1) + \sum_{m=1}^{M} w_{nm} u_m(t)\right] \qquad (1)$$

where $u_m(t)$ is the mth input component at time t, w_{nl} is the recurrent connection weight from the lth neuron to the nth neuron, and w_{nm} is the connection weight from the mth input component to the nth neuron. The operator $f_n(\circ)$ is the sigmoid function. The feedforward weights that connect input to the neurons are trained to recognize the static features of individual speech segment while the recurrent connections among neurons are used to characterize the temporal variation of these features.

Suppose the speech unit Γ consists of K acoustically stationary segments. Each of these segments essentially corresponds to a basic phonetic constituent of Γ. In the proposed RNN speech model, a phonetic segment is represented by an output neuron in the RNN. Without loss of generality, the first K neurons are selected as output neurons. Let $\{\bar{u}(1), \bar{u}(2), \cdots, \bar{u}(T)\}$ be the short-time feature

0-7803-2431-5/95 $4.00

sequence obtained from an utterance U for Γ, where $\bar{u}(t)$ denotes the feature vector of analysis frame at time t. To relate a particular frame with, for example, the kth segment of Γ, the kth output neuron is activated while the other output neurons are inhibited, i.e. $y_k(t) \rightarrow 1.0$ and $y_j(t) \rightarrow 0.0$ for all $1 \le j \le K$ and $j \ne k$. Thus the K segments of Γ are related to the K different activation states of the dynamic RNN model. Let $s(t)$ denote the activation state of the RNN at time t. Then a state sequence $\{s(1), s(2), \cdots, s(T)\}$ is obtained from the network output by,

$$s(t) = \arg\max_{i=1}^{K}\{y_i(t)\} \tag{2}$$

The temporal relationships of the K constituent segments can be reflected by the following constraints on $s(t)$,

(1) $s(t_1) \le s(t_2)$ if $t_1 \le t_2$

(2) $s(t+1) - s(t) = 0$ or $s(t+1) - s(t) = 1$

(3) For all $1 \le k \le K$, there exists t such that $s(t) = k$

These constraints are imposed because of all segments must occur in sequential order and that no segment is to be skipped or omitted. A state sequence that fulfills conditions (1)-(3) is referred as a valid state sequence. If the RNN model is capable of producing a valid state sequence for the input utterance $\{\bar{u}(t)\}$, segmentation for the utterance can be performed as follows,

$$\tau(k) = min\{t | s(t) = k\} \tag{3}$$

where $\tau(k)$ denotes the beginning time instant of the kth segment.

3 Training for an RNN Speech Model

A temporally supervised training algorithm has been devised by Williams and Zipser [6] for RNNs. In this algorithm, the RNN connection weights are adapted by applying the gradient descent technique to a temporal error function defined by,

$$E = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{K}\sum_{k=1}^{K}[d_k(t) - y_k(t)]^2 \tag{4}$$

where $d_k(t)$ and $y_k(t)$ are the desired output and the actual output of the kth output neuron respectively. It appears that Williams and Zipser's method can be applied directly to the training of the proposed RNN speech model since both network architectures are exactly the same. However, the derivation of the target output sequence $\{\bar{d}(t)\}$ needs some special consideration. In fact, a well defined target output sequence is available only if the segmentation of the training utterance is known prior to training. This is obviously impractical, especially when a large amount of training data is involved. Therefore an iterative self-segmentation procedure is developed.

Let $\Omega = \{U^{(1)}, U^{(2)}, \cdots, U^{(P)}\}$ a group of P training utterances for the speech unit Γ. Suppose the feature sequence extracted from $U^{(p)}$ is denoted by $\{\bar{u}^{(p)}(t)\}$. It is reasonable to assume that all these training sequences possess similar speech characteristics that are essential to Γ. Now, when the feature vectors are presented to the RNN, there are basically two different kinds of network responses. Firstly, if a valid state sequence $\{s(t)\}$ is generated by (2), the target output will be given by,

$$d_k(t) = \begin{cases} 1.0 & if\ s(t) = k \\ 0.0 & if\ s(t) \ne k \end{cases} \tag{5}$$

That is, $\{\bar{d}(t)\}$ is obtained from the RNN itself instead of from external "supervisor". Secondly, if the RNN does not produce a valid state sequence, a hypothesized segmentation is used to define the target output sequence. This situation occurs most probably at the beginning of the training process, or when a completely new utterance is being applied. The hypothesized segmentation is very likely to be inaccurate and will be replaced as soon as a self-generated segmentation becomes available.

The complete training procedure is summarized by a flowchart as shown in Figure 2. At the beginning, an initial number of segments for Γ is assumed. For the first training data $\{\bar{u}^{(p)}(t)\}$, an initial segmentation is used. The simplest way of initializing such a segmentation is to divide the utterance into even portions. Better estimation can be made using energy profile and short-time zero-crossing rate. While the RNN connection weights are being adjusted, the initial segmentation is revised by equation (2) & (3) according to the actual network response. Whenever the segmentation is changed, the target output $\{\bar{d}(t)\}$ will be updated accordingly and the training is continued. This iterative process will be terminated when the self-generated segmentation converges and the associated error function E falls below a prescribed threshold. For each of the subsequent training data, similar procedure is applied. Now, since the RNN has been trained with training data from the same class, it is increasingly probable that the RNN produces a valid initial segmentation for a new training sequence.

It should be noted that some adjacent segments may be combined with each other during the training because of their similar features. Therefore the eventual number of segments represented by the RNN speech model may be smaller than the initial value.

The training process of an RNN speech model can be further illustrated by the example shown in Figure 3. In this example, an RNN that consists of 12 neurons is built for the Cantonese syllable /tsam/ (Cantonese is a commonly used Chinese dialect, and a typical monosyllabic and tonal language). This syllable is composed of three phonemes, namely /ts/, /a/ and /m/. The waveform and the spectrogram of the beginning training utterance are displayed with aligned time axis. The initial segmentation is obtained by dividing the utterance evenly into 8 portions. Segments with similar acoustic features are merged together as training proceeds and the segment boundaries are also adjusted. Subsequently

a 4-segment model is reached. The first two segments form the non-continuant affricate /ts/ while the other two correspond to the middle vowel /a/ and nasal ending /m/ respectively.

4 RNN Speech Models for Speech Recognition

The training of an RNN speech model is aimed at producing a temporal representation, i.e. a valid state sequence, for each and every training utterance. In addition, the error function E associated with this state sequence is minimized. In general, a small value of E implies a fairly good modeling of the input utterance. For speech recognition, the value of E can be used as the major discrimination factor between different speech models. To recognize an input utterance U, the output sequences from all models are examined and only those models producing valid state sequences will be considered for further decision. From the remaining eligible models, the one with minimum value of E is selected to be the result of recognition.

The performance of this baseline recognition system has been evaluated for the recognition of isolated Cantonese digits "0" - "10". Each of these digits, as shown in Table 1, contains a single syllable. Therefore 11 RNN syllable models need to be built. The number of segments found in each syllable model is given in Table 1 for both single speaker and multi-speaker cases.

Digit	Phonetic Transcription	No. of segments found in the RNN models (single speaker)	No. of segments found in the RNN models (multi-speaker)
"0"	/liŋ/	3-4	3
"1"	/jat/	2	2
"2"	/ji/	1	1
"3"	/sam/	3-4	3
"4"	/sei/	3-4	3
"5"	/ŋ/	1	1
'6"	/luk/	2-3	3
"7"	/tsat/	2-4	3
"8"	/bat/	2	2
"9"	/gau/	3-4	3
"10"	/sap/	2-3	3

Table 1: The Cantonese Digits "0"-"10" and the Constructed RNN Speech Model

The speech data for our experiments were obtained from five male speakers. Each speaker was asked to produce 12 complete sets of the 11 digits, labeled as trial 1 to trial 12. The odd number trials are grouped to form data set A and the even number trials form data set B. For all utterances, speech features are extracted every 10 ms with a 20 ms analysis window and the feature vector consists of 13 components as,

$$[e, \Delta e, zcr, cep_1, cep_2, \cdots, cep_7, \Delta cep_4, \cdots, \Delta cep_6]$$

where the first three components are the frame energy, delta frame energy and frame zero-crossing rate respectively, cep_i and Δcep_j denote the ith cepstral coefficient and the jth delta cepstral coefficient respectively.

In single speaker speech recognition, data set A of a speaker is used for training first while data set B from the same speaker is used for testing. The training data set and the test data set are later on swapped and the experiment is repeated. The overall recognition rates for training data and test data are found to be 95.3 % and 87.7 % respectively.

In the multi-speaker experiment, the training data includes the data set A from all 5 speakers and the test data includes all the data set B. Similar to the single speaker case, the experiment is repeated with swapped combination of training data and training data. It is observed that a hypothesized initial segmentation is often needed when the training utterance comes from a new speaker. Having been trained with the first training utterance from each speaker, the RNN is capable of generating valid segmentation for most of the subsequent training utterances. The overall recognition rates for training data and test data are found to be 88.0 % and 82.9 % respectively.

	Recognition Accuracy		Rejection Rate	
Digit	Single Speaker	Multi-Speaker	Single Speaker	Multi-Speaker
"0"	88.3 %	86.7 %	0.0 %	3.3 %
"1"	86.7 %	85.0 %	0.0 %	0.0 %
"2"	95.0 %	96.7 %	0.0 %	1.7 %
"3"	85.0 %	78.3 %	8.3 %	6.7 %
"4"	85.0 %	90.0 %	8.3 %	8.3 %
"5"	91.7 %	70.0 %	0.0 %	10.0 %
"6"	85.0 %	63.3 %	0.0 %	0.0 %
"7"	91.7 %	78.3 %	0.0 %	0.0 %
"8"	91.7 %	90.0 %	0.0 %	0.0 %
"9"	80.0 %	88.3 %	0.0 %	0.0 %
"10"	85.0 %	85.0 %	0.0 %	1.7 %

Table 2: Recognition Results for Isolated Cantonese Digits (Test Data)

The recognition results for the test data in both single speaker and multi-speaker applications are given in Table 2. By examining these results carefully, we note that recognition errors are mainly due to the followings :

(1) Inadequate Modeling

Consider an input utterance in which the desired RNN speech model is unable to generate an invalid state sequence. If none of the other RNN speech models produces a valid state sequence, this utterance is rejected. Otherwise, it is misclassified as another undesired syllable in which a valid state sequence is present. In general, it is relatively easy for an RNN speech model with smaller number of activation states to generate valid state sequences. This is because the temporal constraints, i.e. conditions (1)-(3) in Section 2, are less probable to be violated when the total number of segments becomes small. It is found from the experimental results that a number of utterances are misclassified as syllable "5" which is represented by a single-segment RNN. For these utterances, the desired speech models are not selected due to their inadequacy of generating valid state sequences. The "5" model is probably the only one that produces valid state sequence and therefore it is selected.

(2) Lack of Discriminative Training

The RNN speech models are being trained independently but also individually as well. Therefore although individual RNN model has learned about the essential features of a particular syllable, it knows nothing about how this syllable is different from the others. This is why those syllables with one or more phonetic components in common are easily confused. A typical confusion set include the digits "1", "7", "8" and "10" which all share the same vowel /a/. Besides, "6" and "9" could also be easily confused due to the same vowel /u/. To reduce these recognition errors, discriminative training has to be incorporated. The basic idea of discriminative training is to reduce the temporal error function for the desired RNN speech model and simultaneously increase the error functions for all other models. Research work in this direction is still on-going and preliminary results are quite encouraging.

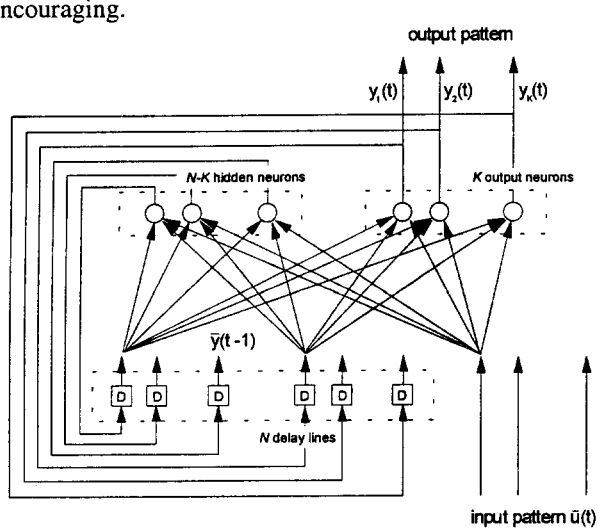

Figure 1: A Fully Connected Recurrent Neural Network

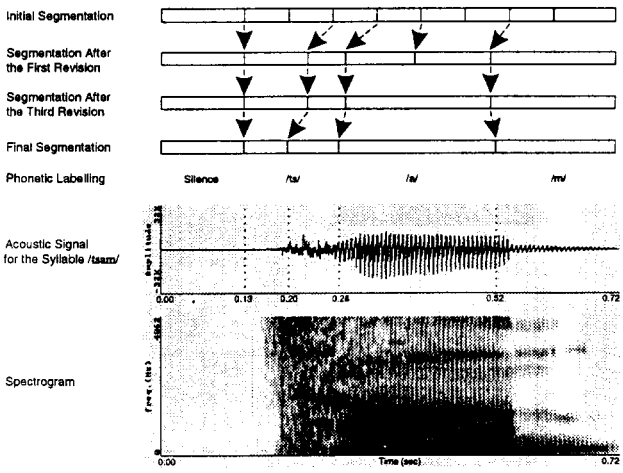

Figure 3 : Training of an RNN Model for the Syllable /tsam/ — An Example

REFERENCES

[1] Richard P. Lippmann, *Review of Neural networks for Speech Recognition*, Neural Computation 1, pp.1 - 38, 1989.

[2] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, *Phoneme Recognition Using Time-Delay Neural Networks*, IEEE Trans. on ASSP, Vol. 37, No.3, pp.328 -339, 1989.

[3] A.J. Robinson and F. Fallside, *Static and Dynamic Error Propagation Networks with Application to Speech Coding*, NIPS (edited by D. Anderson), pp.632 - 641, American Institute of Physics, New York, 1988.

[4] S.Y.Kung, *Digital Neural Networks*, Prentice-Hall, Inc., 1993.

[5] Victor W. Zue, *Acoustic Processing and Phonetic Analysis*, Trends in Speech Recognition (edited by Wayne A. Lea), pp.101 - 124, Prentice Hall, Inc., 1980.

[6] Ronald J. Williams and David Zipser, *A Learning Algorithm for Continually Running Fully Recurrent Neural Networks*, Neural Computation 1, pp.270 - 290, 1989.

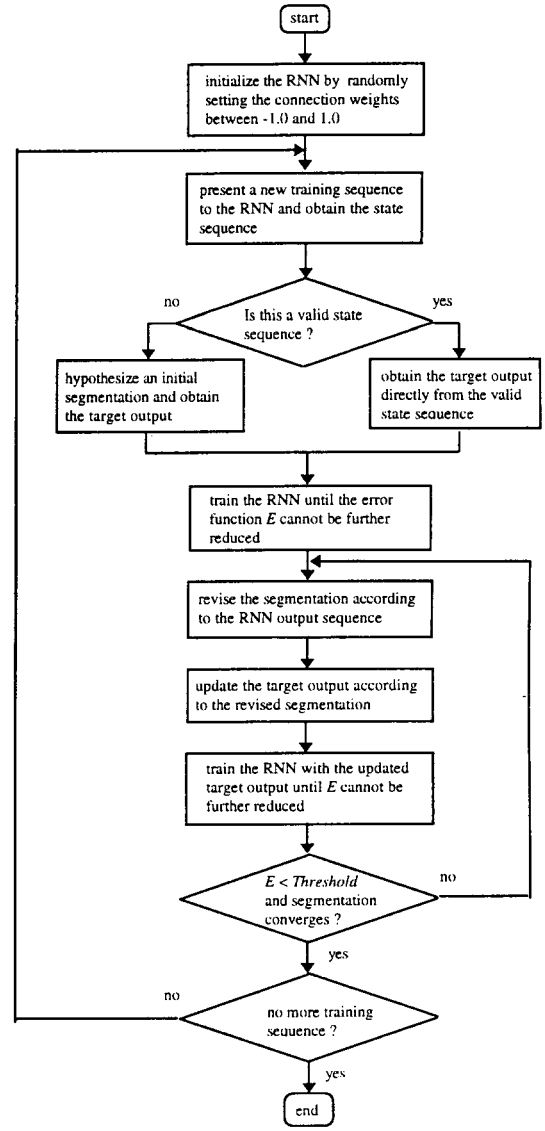

Figure 2: Training Procedure for an RNN Speech Model