Expert Systems with Applications 36 (2009) 6368–6374

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Dynamic programming prediction errors of recurrent neural fuzzy networks for speech recognition

Chia-Feng Juang *, Chun-Lung Lai, Chiu-Chuan Tu

Department of Electrical Engineering, National Chung-Hsing University, Taichung 402, Taiwan, ROC

ARTICLE INFO

Keywords:
Phrase recognition
Recurrent fuzzy systems
Fuzzy neural networks
Recurrent neural fuzzy networks
Noisy speech recognition

ABSTRACT

This paper proposes Mandarin phrase recognition using dynamic programming (DP) prediction errors of singleton-type recurrent neural fuzzy networks (SRNFNs). This method is called DP-SRNFN. The recurrent property of SRNFN makes it suitable for processing temporal speech patterns. A Mandarin phrase comprises monosyllabic words. SRNFN training is based on the word unit. There are N_w SRNFNs for modeling N_w words, and each SRNFN receives the current frame feature and predicts the next one of its modeling word. In recognizing N_P phrases, the prediction error of each trained SRNFN is computed, and DP is used to find the optimal path that maps the input frames to the best matched SRNFNs (words) for each of the N_P phrases. The accumulated error of each phrase model is computed from its optimal path and the one with the minimum error is the recognition result. To verify DP-SRNFN performance, this study conducted experiments on recognizing 30 Mandarin phrases. SRNFN training with noisy features for phrase recognition under different noisy environments was also conducted. DP-SRNFN performance is compared with the hidden Markov models (HMMs). Results show that DP-SRNFN achieves higher recognition rates than HMM in both clean and noisy environments.

1. Introduction

Researchers have published many studies on automatic speech recognition (ASR) in recent years. Extensive applications in ASR include voice dialog systems for telephone numbers, account numbers, and portable wireless devices (Brems & Wattenbarger, 1994; Hunt, 2001; Viikki, 2001). In an intelligent car, voice commands for mobile telephones or car instruments create a safe and convenient interface (Hunt, 2001; Lin, Lin, & Wu, 2002). These and other applications have made ASR technology a very popular research direction for friendly human machine interfacing. Hidden Markov models (HMMs) have become one of the dominant approaches in ASR, especially for large vocabulary or phrase recognition applications (He, Kwong, & Hong, 2004; Rabiner & Juang, 1993; Wu & Huo, 2007; Zeng & Liu, 2006).

Besides HMMs, another popular ASR technique is the use of artificial neural networks (ANNs) (Ahmad, Ismail, & Samaon, 2004; Jin & Freeman, 2006; Lippmann, 1989; Petek, 2000; Waibel, Hanazawa, Hinton, Shiano, & Lang, 1989). ANNs have been proposed for their nonlinear mapping functionality, learning ability, and flexible architecture, which can easily accommodate contextual inputs and feedback. Researchers have proposed some recurrent ANNs for ASR because speech signals have temporal properties (Ahmad et al., 2004; Mesbahi & Benyetton, 2004). In contrast to pure feedforward architectures, which exhibit static input–output behavior, these recurrent models are able to store information from the past (e.g., prior system states) and are thus more appropriate for analyzing dynamic systems. In addition to recurrent ANNs, researchers have also proposed recurrent fuzzy neural networks that have proved to perform better than recurrent ANNs (Juang, 2002; Juang, Chiou, & Lai, 2007; Lin & Chen, 2005; Mastorocostas & Theocharis, 2002; Theocharis, 2006; Zhang & Morris, 1999). In Juang et al. (2007), a singleton-type recurrent neural fuzzy network (SRNFN) was proposed for isolated Mandarin word recognition. SRNFN recognition performance was shown to be better than other recurrent fuzzy systems and feedforward neural networks simulated in that study. This paper applies SRNFN to Mandarin phrase recognition using the dynamic programming (DP) prediction errors of SRNFN. This method is called DP-SRNFN hereafter.

Previous studies proposed combining DP with ANNs for speech recognition (Tebelskis & Waibel, 1990), but these approaches have two major disadvantages. These studies first used feedforward neural networks, where the number of context speech frames in network inputs must be decided by experimental trials. Including context frames in the inputs increases the input dimension and the overall network size. Second, many neural models are required to model an English phoneme (Tebelskis & Waibel, 1990) or a

* Corresponding author.
E-mail address: cfjuang@dragon.nchu.edu.tw (C.-F. Juang).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.07.061

Japanese word (Iso & Watanabe, 1990), which generates a large amount of neural models. Training these neural models to model a phoneme or a word is complex and requires the use of DP in each training iteration to determine the best match between neural models and input speech frames. The DP-SRNFN proposed in this paper can handle these two disadvantages. The recurrent property of the SRNFN makes it possible for the DP-SRNFN to address the first disadvantage by using only the current speech frame as the network input. For the second problem, only one SRNFN is used for modeling a word. The number of recognition words is therefore equal to the number of SRNFNs, which reduces the number of SRNFNs required. Training SRNFNs is easy, and the DP technique is not required during training. The DP technique is used only during the phrase recognition phase.

This paper proposes DP-SRNFN for Mandarin phrase recognition. The DP-SRNFN method trains one SRNFN to model the temporal relationship of one word. If there are a total of N_w different words in the recognized phrases, then there is a total of N_w SRNFNs. The temporal property of a phrase can be modeled by linking SRNFNs. Using DP solves the optimal mapping problem between the connected SRNFNs and phrase comprising words. Phrases are recognized by the accumulated DP-SRNFN prediction errors for each phrase. This paper studies DP-SRNFN performance in noise-free and noisy environments, and compares its performance with HMMs.

This remainder of the paper is organized as follows: Section 2 describes the SRNFN structure and training. Section 3 introduces DP-SRNFN phrase recognition. Section 4 demonstrates DP-SRNFN recognition results in clean and noisy environments. Finally, Section 5 draws conclusions.

2. SRNFN structure and training for phrase recognition

2.1. SRNFN structure and learning

The proposed DP-SRNFN is based on the use of singleton-type recurrent neural fuzzy networks (SRNFNs) (Juang et al., 2007). Fig. 1 shows the structure of SRNFN that has two external input variables x_1 and x_2 and a single output y. Accordingly, a SRNFN has two nodes in layer 1 and one node in layer 5. The feedback loops in layer 4 enables SRNFN to handle problems with temporal characteristics. To give a clear understanding of the mathematical function of each node, SRNFN functions are described layer by layer. In layer 1, each node corresponds to one input variable and directly transmits input values to the next layer, thus requires no computation. In layer 2, each nodes correspond to one fuzzy set and calculates a membership value. This layer uses two types of membership functions. Gaussian membership functions which locally map the input spatial space to the output space are used for external input x_j, and the mathematical function is

$$M_G^i(x_j) = \exp\left\{-\left(\frac{x_j - m_{ij}}{\sigma_{ij}^2}\right)^2\right\} \tag{1}$$

where m_{ij} and σ_{ij} are, respectively, the center and the width of the Gaussian membership function of the ith term of the jth input variable x_j. For internal variable h_i, the following sigmoid membership function is used:

$$M_S(h_i) = \frac{1}{1+\exp\{-h_i\}} \tag{2}$$

Each internal variable has a single corresponding fuzzy set. Links in layer 2 are all set to unity. In layer 3, each node represents a fuzzy logic rule and performs antecedent matching of this rule using the following AND operation

$$\mu_i = M_S(h_i)\cdot\prod_{j=1}^{n} M_G^i(x_j) = \frac{1}{1+\exp\{-h_i\}}\cdot\prod_{j=1}^{n}\exp\left\{-\left(\frac{x_j - m_{ij}}{\sigma_{ij}}\right)^2\right\} \tag{3}$$

where n is the number of external inputs. The link weights are all set to unity. In layer 4, each node corresponds to one context node and performs a defuzzification operation for internal variables h. The simple weighted sum is calculated in each node

$$h_i = \sum_{k=1}^{r}\mu_k w_{ik} \tag{4}$$

As in Fig. 1, the delayed value of h_i is fed back to layer 1 and acts as an input variable to the antecedent part of a rule. Each rule has a corresponding internal variable h and is used to decide the temporal history's degree of influence on the current rule. In layer 5, each node corresponds to one output variable and performs weighted average operations for output y. The mathematical function is

$$y = \frac{\sum_{i=1}^{r}\mu_i b_i}{\sum_{i=1}^{r}\mu_i} \tag{5}$$

where b_i is a fuzzy singleton value functioning as the consequent part of output variable y.

In a SRNFN, there are initially no rules, and the rules are constructed by online structure and parameter learning. That is, no pre-assignment of fuzzy if-then rules is required in a SRNFN. Constructing a SRNFN can be divided into two subtasks: structure learning and parameter learning. The objective of structure learning is to determine the number of fuzzy rules, the initial location of membership functions, and the initial consequent parameters. A pre-defined threshold F_{in} is used as a criterion for the generation of fuzzy rules. More rules are generated for a larger value of F_{in}. The initial width of each generated Gaussian fuzzy set is decided by a pre-defined constant β. The objective of parameter learning is to optimally tune the free parameters of the constructed network using a real-time recurrent learning algorithm. The learning speed of the RTRL is controlled by a learning constant η. Details of the learning algorithm can be found in Juang et al. (2007).

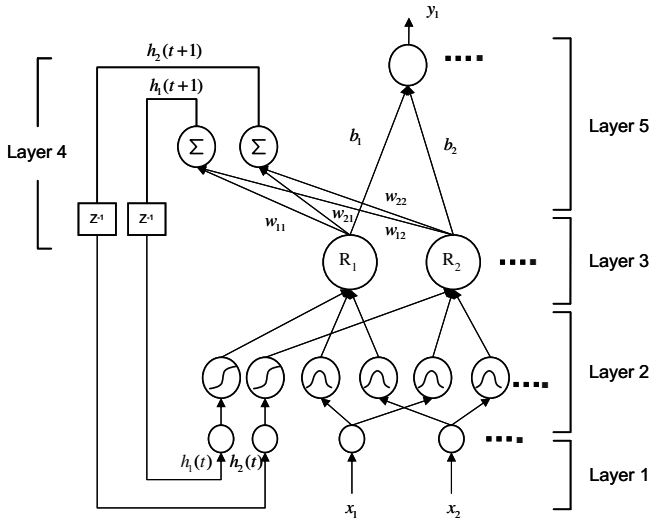

Fig. 1. Structure of the singleton-type recurrent neural fuzzy network (SRNFN).

2.2. SRNFN training for phrase recognition

SRNFN training is based on the word unit. This study uses SRNFNs as pattern predictors instead of pattern discriminators. If