A state of the art end-to-end speech recognition algorithm with a homogenous structure.

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

TODO(chanw.com) Revise the abstract. This paper proposes a new end-to-end speech recognition system referred to as Recurrent neural network Sequence Classification (RSC). Recently, end-to-end speech recognition systems have been gaining more attention from researchers and several different structures have been proposed. Typical examples include Connectionist Temporal Classification (CTC), Recurrent Neural Network Transducer (RNN-T), and Sequence-to-Sequence Modeling using the attention mechanism. The CTC-based approaches are using the CTC loss assuming the conditional independence property. Although the structure of CTC is simpler than other end-to-end systems, the performance is worse due to this conditional independence assumption. RNN-T and attention-based approaches have distinct components such as the prediction network and encoder in RNN-T, and the encoder, the decoder, and the attention layer in the attention-based model. Compared to these models, SHC has a homogeneous structure from the bottom to the top layer. LSTM and max-pooling layer are repeated followed by the top softmax layer. The embedded softmax output is fed back as input of each RNN layer. The training starts with flat initialization and alignment is made after every epoch. The loss is the Cross Entropy (CE) loss using this alignment result. In our experimental results, SHC algorithm has shown better performance than more complicated attention-based end-to-end speech recognition system. Another major advantage is training is significantly faster.

1 Introduction

Recently, deep neural network has significantly improved the performance of speech recognition systems [].

2 Review of end-to-end speech recognition systems

Recently, there have been growing interests in end-to-end speech recognition systems. [CITE] Fig. 1 shows the entire structure of the Connectionist Temporal Classification (CTC) and the attention-based end-to-end speech recognition system.

3 Sequence Homogeneously-Structured Classifier

Figure **??** shows the entire structure of the SHC. Speech-recognition task can be also considered as a

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

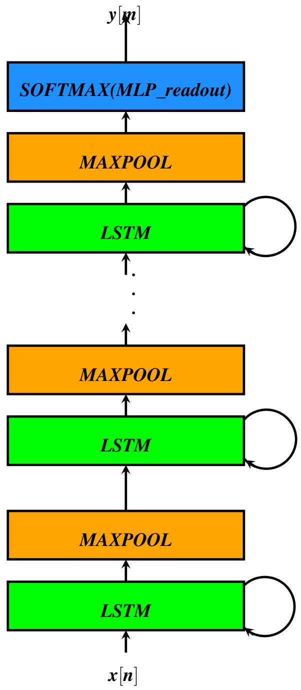

Figure 1: The entire structure of the sequence homogeneously-structured classifier (SHC).

3.1 The structure of the Sequence Homogeneously-Structured Classifier

Fig. shows the block diagram for enhancing the noisy feature.

Fig. 2 shows the structure of the SHC algorithm. The structure is very simple and somewhat similar to that of CTC shown in TODO

3.2 Training of the Sequence Homogeneously-Structured Classifier

In this section, we describe how to train SHC. The procedure is summarized in the following table.

3.3 Viterbi alignment and N-best alignment

In conventional frame-wise CE-training, Viterbi alignment (a.k.a forced alignment) has been performed to obtain the frame-level acoustic unit boundaries TODO(chanwcom). Even though the Viterbi algorithm has advantages in simplicity and efficiency, it is based on the conditional independence assumption.

In TFSC, we propose the following N-best alignment approach rather than the Viterbi alignment algorithm to find the alignment information.

for $m = 0, ..., M-1$ **do**
    for $l = 0, ..., N_b - 1$ **do**
        $\pi^{(l)}[m]$
    end for
end for
if $i \geq maxval$ **then**
    $i \leftarrow 0$
else
    if $i + k \leq maxval$ **then**
        $i \leftarrow i + k$
    end if
end if

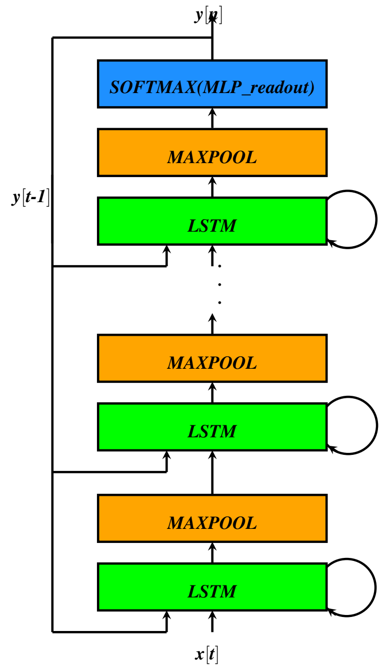

Figure 2: The entire structure of the sequence homogeneously-structured classifier (SHC).

3.4 Feature extraction

Even though the log-mel coefficients have been widely used as features for speech recognition [5], the log non-linearity has a disadvantage in that the value diverges to negative infinity as the mel filterbank coefficient approaches zero [1]. Thus, we use the power-law nonlinearity rather than the log-nonlinearity:

$$q[m, l] = (p[m, l])^{\frac{1}{15}} \tag{1}$$

We use the power coefficient of $\frac{1}{15}$ as in [1, 3].

In this section, we discuss how to predict the clean feature and the error ratio from the corrupt feature. Let us denote the clean target feature, the corrupt feature by $\boldsymbol{x}, \boldsymbol{y}$ respectively.

Using the room simulation system described in [2], we create a pair of clean and corrupt utterances. Log-mel [XX] features are calculated from these original clean and corrupt utterances. Let us denote the clean feature by $\boldsymbol{t}_i$ and the simulated corrupt feature by $\boldsymbol{x}_i$, respectively where i is the utterance index. The training set is represented by the following set:

$$\mathcal{T} = \{< \boldsymbol{x}_i, \boldsymbol{t}_i > | 0 \le i \le N-1\} \tag{2}$$

where N is the number of training examples.

In the ECE algorithm, we first estimate the true target $\boldsymbol{t}_i$. The estimation error is defined by the following equation:

$$\boldsymbol{e}_i = \boldsymbol{t}_i - \boldsymbol{x}_i, \quad 0 \le i \le N-1 \tag{3}$$

The norm of the error vector $\boldsymbol{e}_i$ would be generally large when the norm of the estimated clean feature is large.

Thus, instead of trying to estimate the error itself, the neural network tries to estimate the error ratio given as follows:

$$r_i = \tag{4}$$

4 Generation of simulated clean and noisy data sets

To train neural networks described in section **??**, we need a training set consisting of pairs of clean speech and noisy speech. Unfortunately, it is not easy to have such a training set from real speech utterances. Thus, we use the simulation system described in [2] to synthetically generate noisy speech utterances from the clean speech utterances.

In our application, for clean training set, we used Wall Street Journal (WSJ) si-284. For noise set, we used the Resource Management 1 (RM1) [4] 50 % of time and DEMAND noise [TODO(chanwcom) Adds reference] for the remaining 50 % of time.

TODO(chanwcom) Cite DEMAND noise set.

5 Estimation of the optimal weight in the EFW algorithm

In this section, we describe the procedure of feature weighting approach. Suppose that the feature before enhancement is represented by $\vec{x}$. Let us represent the inference neural network to estimate the target by $\mathcal{T}$.

$$\mathcal{X} = \{\} \tag{5}$$

Instead of directly using the enhanced feature $\widehat{\vec{y}}$, we consider the following interpolated feature $\boldsymbol{z}$ from the original corrupt input $\boldsymbol{x}$ and .

$$\vec{z} = \vec{w} \odot \widehat{\vec{y}} + (1 - \vec{w}) \odot \vec{x} \tag{6}$$

where $\odot$ denotes the Hadamard product (entry-wise product).

Let us denote the estimated variance vector by $\hat{\vec{v}}$:

$$\begin{aligned} \hat{\vec{v}} &= \mathrm{Var}[\widehat{\vec{y}}] \\ &= E[(y - y)^2 | \vec{x}[0], \vec{x}[1], ...\vec{x}[T]] \end{aligned} \tag{7}$$

In our discussion, let us assume that the expectation of $\hat{\vec{y}}$ is the same as the true target $\vec{y}$.

$$E[\widehat{\vec{y}}] = \vec{y} \tag{8}$$

Now, let us obtain the expected value of $\vec{z}$ from (6) and (8)

$$E[\vec{z}] = (1 - \vec{w}) \odot (\vec{x} - \vec{y}) \tag{9}$$

Thus, the bias of $\vec{z}$ is given by:

$$\begin{aligned} Bias_{\vec{z}} &= E[\vec{z}] - t \\ &= \vec{w} \odot (\widehat{\vec{y}} - \vec{y}) \end{aligned} \tag{10}$$

From (6) and (7), the variance of $\vec{z}$ is given by:

$$Var_{\vec{z}} = \vec{v} \odot w \odot w. \tag{11}$$

The mean squared error of the i-th element of the $\vec{z}$ is given by:

$$\begin{aligned} MSE_{\vec{z}_i} &= Bias_{\vec{z}_i}^2 + Var_{\vec{z}_i} \\ &= \left[(\vec{x}_i - \vec{y}_i)^2 + \vec{v}_i^2 \right] \vec{w}_i^2 - 2(\vec{x}_i - \vec{y}_i)^2 \vec{w}_i + (\vec{x}_i - \vec{y}_i)^2 \end{aligned} \tag{12}$$

Since the above equation (12) is a quadratic equation with respect to $\vec{w}_i$, we may find that the minimum value of $MSE_{\vec{z}_i}$ is obtained when $\vec{w}_i$ has the following value:

$$\vec{w}_i = \frac{(\vec{x}_i - \vec{y}_i)^2}{(\vec{x}_i - \vec{y}_i)^2 + \vec{v}_i} \tag{13}$$

Thus, the final form of the estimated vector $\widehat{\vec{z}}$ becomes:

$$\widehat{\vec{z}} = \widehat{\vec{w}} \odot \widehat{\vec{y}} + (1 - \widehat{\vec{w}}) \odot \vec{x} \tag{14}$$

where

$$\widehat{\vec{w}} = (\vec{x} - \vec{y})^{\odot 2} \left[(\vec{x} - \vec{y})^{\odot 2} + \vec{v} \right]^{\odot -1} . \tag{15}$$

Sequence enhancement

The feature enhancement is to map a sequence of corrupt features $\mathbf{X}$.

$$\mathbf{X} = (\vec{x}[0], \vec{x}[1], \cdots, \vec{x}[M-1]) \tag{16a}$$
$$\mathbf{Y} = (\vec{y}[0], \vec{y}[1], \cdots, \vec{y}[M-1]) \tag{16b}$$

Acknowledgments

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Remember that you can go over 8 pages as long as the subsequent ones contain *only* cited references.**

References

[1] C. Kim and R. M. Stern. Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition. *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, pages 1315–1329, July 2016.

[2] C. Kim, A. Misra, K.K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani. Generation of simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home. In *INTERSPEECH-2017*, pages 379–383, Aug. 2017.

[3] C. Kim and R. M. Stern. Power-normalized cepstral coefficients (pncc) for robust speech recognition. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 4101–4104, March 2012.

[4] Price, P, et al. *Resource Management RM1 2.0 LDC93S3B. DVD*. Linguistic Data Consortium, Philadelphia, PA, 1993.

[5] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust., Speech, and Signal Processing*, 28(4):357–366, Aug. 1980.