ESANN'1997 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 16-17-18 April 1997, D-Facto public., ISBN 2-9600049-7-3, pp. 1-6

Speaker Normalization with a Mixture of Recurrent Networks

Edmondo Trentin and Diego Giuliani

Istituto per la Ricerca Scientifica e Tecnologica
I-38050 Povo (Trento), Italy
{trentin,giuliani}@irst.itc.it

Abstract.
This work introduces a multiple connectionist architecture based on a mixture of Recurrent Neural Networks to approach the problem of speaker adaptation in the acoustic feature domain (i.e. *speaker normalization*). Normalization is applied to the case of a speaker-independent (SI) speech recognition system based on continuous density hidden Markov models. The technique for combining multiple recurrent models is discussed. Recognition experiments with a continuous speech large dictionary task shows that the proposed architecture is capable to tangibly improve recognition performance, allowing for a 21.9% reduction of the word error rate.

1. Introduction

This work deals with speaker normalization [6]. The aim is the reduction of the difference between the acoustic space of a speaker and the training acoustic space of a speech recognizer, in order to increase recognition performance.

Speech recognition systems work on a parametric representation of acoustic data. Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_I\}$ be the parametric representation of an input speech signal, e.g. corresponding to a sentence uttered by the new speaker, where $\mathbf{x}_i \in \mathbf{R}^d$ is feature vector at time i. We want to estimate a feature transformation ϕ, such that the transformed test utterance $\hat{X} = \{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_I)\}$ *better matches* the recognizer training data, resulting, on average, in improved recognition performance. In general ϕ is estimated starting from a learning corpus of vector pairs $\ell = \{(\mathbf{x}_n, \mathbf{y}_n) \mid n = 1, ..., N\}$, where $\{\mathbf{x}_n\}$ are feature vectors in the acoustic space of the new speaker, while the corresponding $\{\mathbf{y}_n\}$ are vectors in the training space of the recognizer. The training set ℓ is in general obtained from a (small) set of adaptation utterances collected from a new system user during an enrollment session. A suitable procedure is adopted here for generation of ℓ in the case of SI speech recognizers. Estimating ϕ can be afforded either with statistical techniques [1, 8] or artificial neural networks (ANN) [6, 4].

A generalized Multi-Layer Perceptron [5] is first considered to solve the above regression problem. In order to exploit the sequential nature of speech patterns, the use of Recurrent Neural Networks (RNN) is then investigated. Multiple linear models, as well as mixtures of both static and recurrent nets, are finally developed and compared. The mixture of recurrent models requires an *ad-hoc* combination method based on statistical considerations on the transformed metrics of the input space, induced by the dynamics of the networks. Results in a continuous speech, large dictionary task obtained with a mixture of 8 phone-class dependent RNN yields a significant 21.9% Word Error Rate (WER) reduction with respect to the SI continuos density hidden Markov model (CDHMM) based recognizer alone.

2. The recognizer and the experimental environment

A set of 34 context independent acoustic-phonetic speech units is modeled with left-to-right Continuous Density HMMs [9] . Each emission probability is modeled with a mixture consisting of 16 Gaussian components with diagonal covariance matrices. System training was performed using 2140 utterances from 100 speakers (50 males and 50 females).

The recognition task consists of dictation of fragments of newspaper articles taken from the financial Italian journal *Il Sole 24 Ore*. It is a continuous speech task with word dictionary size of 10,000. For each of four test speakers (3 males and 1 female), two sets of utterances were collected: the adaptation set (15 utterances) and the test set (30 utterances). Each utterance presents a duration of 12 sec. on average, approximately corresponding to 19 words.

Speech signals were processed in order to obtain an adequate parametric representation. For each frame, 8 Mel Scaled Cepstral Coefficients (MSCCs) [2] and the log-energy, together with their first and second order time-derivatives, were extracted and arranged in a 27-dimensional feature vector.

The training set $\ell = \{(\mathbf{x}_i, \mathbf{y}_i)\}$ is built as follows. Each adaptation utterance is aligned against the concatenation of Markov models corresponding to its phonetic transcription. A suitable Viterbi [9] alignment strategy puts in correspondence each feature vector of the input sequence to a specific model, a specific state within the model, and an individual Gaussian component (among the component densities attached to the state). An association is thus established between the input feature vector and the mean vector of the corresponding Gaussian component. The latter plays the role of a synthetic pattern in the training acoustic space of the recognizer, and becomes the corresponding target vector for the input frame.

3. Architecture and dynamics of the recurrent model

The basic neural model that was adopted is a 2-layer Multi-Layer Perceptron (MLP), trained with the backpropagation (BP) algorithm [5], in the stochastic gradient version. An adaptive learning rate and a momentum rate were used. An equal number of input and linear output units, namely 9, was set to coincide with the dimensionality of the normalization problem (8 MSCCs and the log-energy). The hidden layer was built of 48 sigmoidal units.

The class of RNN that was considered presents an extra set of *status* or *context units*, providing a kind of duplicate of the input layer. The context layer is linked with feedforward connections to the hidden layer. Direct lateral connections from each input unit to the corresponding context unit hold, with shared weight w_l. Self connections with weight w_s are also present in the context layer, as well as back recurrent links from each output unit to the corresponding *backup unit* of the newly introduced layer, with shared weight w_b. A time-delay mechanism is also allowed on additional backward connections, with the addition of extra units with delay lines. The role of the context units is to keep track of an internal, evolving state of the network. The dynamic of context unit k is described in terms of its output o_k at time $t \geq 0$, whose equations are

$$\begin{array}{l} o_k(0) = 0 \\ o_k(t+1) = w_l x_k(t) + w_s o_k(t) + \sum_{j=0}^{q-1} w_b^{j+1} y_k(t-j) \end{array} \tag{1}$$

where w_l is the common weight of lateral connections between input unit k and the context unit itself, $x_i(t)$ is i-th component of t-th pattern in the input sequence, $y_i(t)$ is i-th output of network at time t, and q is the number of steps back in time at which the past outputs are to be considered.

4. Multiple-model regression: combining recurrent nets

Speaker normalization can be formulated as a multiple regression problem. Given the set of observations $\ell = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, ..., N\}$ to be fitted, the regression equation considered here is in the form

$$\mathbf{y}_i = \sum_{j=1}^{c} w_j(\mathbf{x}_i) f_j(\mathbf{x}_i, \mathbf{\Theta}_j) + \epsilon_i \tag{2}$$

where c models $f_1(), ..., f_c()$ are simultaneously considered, along with their parameter vectors $\mathbf{\Theta}_1, ..., \mathbf{\Theta}_c$. ϵ_i is a random vector having mean $\mu_i = \mathbf{0} \in \mathbf{R}^d$ and covariance matrix Σ_i. The contribution from each model $f_j()$ is determined by the value of the corresponding *weight* $w_j()$ that, in turn, is a function of the independent vector $\mathbf{x}_i$. For a given input, weights must sum to 1. The problem is to determinate the parameters $\mathbf{\Theta}_1, ..., \mathbf{\Theta}_c$ that allow for the best fit of the observations in ℓ. When neural networks are used as regression models, the parameters are the connection weights. In terms of *Mixture of Experts* [7] eq. 2 is defined to be a *linear opinion pool* model.

In the speaker normalization setup we assume that the acoustic vectors of a given speaker can be described as random vectors drawn from a finite *mixture density* function $p(\mathbf{x})$ that can be expressed as

$$p(\mathbf{x}) = \sum_{j=1}^{c} p_j(\mathbf{x}) \Pi_j \tag{3}$$

with Gaussian component densities $p_j(\mathbf{x})$ and mixing parameters Π_j, for $j = 1, ..., c$. Here $p_j(\mathbf{x})$ denotes the class-conditional probability density function of $\mathbf{x}$ given j-th of c acoustic *populations*, or *classes*. Populations have *a priori* class probabilities $\Pi_1, \Pi_2, ..., \Pi_c$, respectively, and the obvious condition $\sum_{j=1}^{c} \Pi_j = 1$ holds. When labeled data are available, i.e. a class label is attached to each training vector $\mathbf{x}_i$, supervised parameter estimation can be performed on a class by class basis. A natural choice for the weights $w_j(\mathbf{x}_i)$ of eq. 2 is given, once eq. 3 holds, precisely:

$$w_j(\mathbf{x}_i) = \frac{\Pi_j p_j(\mathbf{x}_i)}{\sum_{k=1}^{c} \Pi_k p_k(\mathbf{x}_i)}. \tag{4}$$

Each model $f_j()$ of eq. 2 is specialized on the corresponding class, through a weighted training technique based on the weights computed so forth. In this work, the c component densities are estimated considering to which phonetic class a given feature vector $\mathbf{x}_i$ is assigned, according to the Viterbi alignment procedure, so that $c = 8$.

When each $f_j()$ in eq. 2 is realized using a static MLP, learning is accomplished using a gradient descent method that takes into account the weight $w_j(\mathbf{x}_i)$ that i-th training pattern in ℓ has on the training of j-th model. This requires the minimization of an objective function of the form

$$E = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{d} (y_{ik} - \sum_{j=1}^{c} w_j(\mathbf{x}_i) o_{jk}(\mathbf{x}_i))^2 \tag{5}$$

where $o_{jk}(\mathbf{x}_i)$ denotes k-th output of network j when fed with input $\mathbf{x}_i$. Calculation of a weighted version of the BP algorithm is straightforward, by taking partial derivatives of eq. 5 with respect to the connection weights.

In the static case, weight is assigned to a given (local) model according to statistical considerations concerning the distributions of patterns in the acoustic feature space. This approach is no longer viable in the case of dynamic architectures, because the *competence* of an individual *expert* over a given input pattern is not a function of the current input only. Considering eq. 1, the behavior of the recurrent models can be seen as a static feed-forward computation over a transformed input space. This can be stated by saying that the recurrent connections induce a new, modified metric over the input space. The combination technique for RNN proposed herein, follows from an application of the weighting criteria discussed for static models on the transformed input space induced by the recurrent connections. Statistical criteria applied to the transformed space concern dynamical properties of each *expert* (each point in the new space is a trajectory in the original space), and competence is assigned according to sequences, rather than on individual data.

5. Experimental results

The regression techniques previously described, along with a linear regression based on *least squares* criterion (computed using the Singular Value Decomposition technique [3]) introduced for comparison purposes, were applied in a set of recognition experiments in combination with the SI recognition system.

The results are summarized in Table 1 where performance (averaged over the four test speakers) obtained with the SI system alone (*Baseline*) is compared with that obtained applying the three normalization modules. During experiments, just the 8 MSCCs and the log-energy are transformed by the normalization module. To complete the parametric representation, required by the SI recognizer, first and second order derivatives are computed from transformed coefficients.

Table 1 shows that the single linear module is unable to improve the baseline, while the model based on single MLP tangibly improves the system performance. Furthermore, the module based on single RNN outperforms the other models. This emphasizes that the required feature vector transformation is highly non-linear, and that RNN better exploits the sequential nature of the training data.

Table 1: *Average WERs for the test speakers using the SI recognition system and different normalization modules.*

Module	*Word Error Rate*	
	Single model	*Multiple model*
Baseline	11.4	11.4
Linear	11.5	9.6
MLP	10.2	9.3
RNN	9.3	8.9

Multiple regression modules outperform the corresponding single modules (this is particularly evident in the case of the multiple linear model). Module based on multiple RNN still performs better than the module based on multiple MLPs (although the improvement is not so evident as in the single model case), finally yielding a tangible 21.9% WER reduction with respect to the SI baseline.

6. Conclusions

This paper presented an approach to multivariate regression problems based on the combination of multiple RNN. The architecture and the dynamics of the nets, as well as a combination technique based on statistical properties of the feature space were described. The proposed technique is an extension to the mixture of static, generalized feed-forward connectionist models.

The technique was applied in a difficult and widely investigated non-linear regression task, namely speaker normalization for a SI recognition system. Linear regression resulted unable to improve performance of the baseline system. Regression based on single connectionist models allowed for a gain, and multiple connectionist models based on ANN mixtures further improved the performance. RNN better exploited training data, outperforming MLP in both the cases of single and multiple model regression, resulting in a remarkable 21.9% WER reduction with respect to the SI baseline when 8 models, one for each phone-class in the feature space, were used.

References

[1] J. R. Bellegarda, P. V. de Souza, A. J. Nádas, D. Nahamoo, M. A. Picheny, and L. R. Bahl. Robust speaker adaptation using a piecewise linear acoustic mapping. In *Proc. of ICASSP*, pages I–445–448, San Francisco, March 1992.

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

[3] F. Deprettere. *SVD and Signal Processing Algorithms, Applications and Architectures*. North-Holland, Amsterdam, 1988.

[4] C. Furlanello, D. Giuliani, and E. Trentin. Connectionist speaker normalization with Generalized Resource Allocating Networks. In D. S. Touretzky G. Tesauro and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 1704–1707, Cambridge MA, 1995. MIT Press.

[5] John Hertz, Anders Krogh, and Richard Palmer. *Introduction to the Theory of Neural Computation*. Addison Wesley, 1991.

[6] X. D. Huang. Speaker normalization for speech recognition. In *Proc. of ICASSP*, pages I–465–468, San Franscisco, March 1992.

[7] R. A. Jacobs. Methods for combining experts' probability assessments. *Neural Computation*, 7:867–888, 1995.

[8] H. Matsukoto and H. Inoue. A piecewise linear spectral mapping for supervised speaker adaptation. In *Proc. of ICASSP*, pages I–449–452, San Francisco, March 1992.

[9] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition,. *Proc. of IEEE*, 77(2):267–295, October 1989.