MLP-BASED FACTOR ANALYSIS FOR TANDEM SPEECH RECOGNITION

Marc Ferràs and Hervé Bourlard

Idiap Research Institute, CH-1920 Martigny, Switzerland

ABSTRACT

In the last years, latent variable models such as factor analysis, probabilistic principal component analysis or subspace Gaussian mixture models have become almost ubiquitous in speech technologies. The key to its success is the joint modeling of multiple effects in the speech signal they address. In this paper, we propose a novel approach to use phone and speaker variabilities together to estimate phone posterior probabilities on a tandem speech recognition system. A Multilayer Perceptron (MLP) with 5 layers and a central bottleneck linear layer is used as a basic processing block that mimics the processing undergone in factor analysis. With multiple factors, phone and a speaker MLP are merged at the bottleneck level to obtain better estimates for the phone posterior probabilities used in the ASR system. Experiments on the WSJ corpus show that the joint phone-speaker modeling can significantly outperform phone modeling alone in terms of Frame Error and Word Error Rates.

Index Terms— tandem speech recognition, factor analysis, neural network, multilayer perceptron

1. INTRODUCTION

In the last years, latent variable models [1] introducing hidden variables to explain the correlations amongst a set of observations have received special attention in speech technologies. The use of multiple factor analysis to jointly model speaker and session variabilities affecting the mean vectors of Gaussian Mixture Models (GMM), so-called Joint Factor Analysis (JFA) [2, 3], set a new performance standard of speaker verification systems a few years ago. Recently, the i-vector [4] approach to speaker verification using another latent variable model, Probabilistic Linear Discriminant Analysis (PLDA) [5, 6], to compensate for session variability effects has become the state-of-the-art. These techniques have also made their way into the speech recognition field with the Subspace GMM [7] with satisfactory results. All of them are inspired on previous work on parametric adaptation and training such as Cluster adaptive Training [8] and Eigenvoices [9], using analogous variability models and parameter estimation strategies.

The above mentioned models make the assumption that a small set of variables can explain the correlation of the observations by means of a linear transformation. The observations are thus constrained to originate from a low dimensional subspace and have a low-rank covariance matrix. It is common to have more than one set of latent variables that capture different variabilities, e.g. speaker and session variabilities in JFA or phone and speaker variabilities in SGMM. Assuming that the latent variables follow some a priori distribution, these techniques obtain prior knowledge about the correlations between observations and latent variables by estimating the so-called factor loading matrices, in factor analysis terminology., in the training phase while only the latent variables are estimated in the test phase. These models are essentially generative and the estimates are usually found by maximizing the likelihood function, conditioned to the hidden variables.

On the other side, progress in artificial neural networks has shown the potential of deep learning architectures for acoustic modeling [10]. Deep neural networks (DNN) are capable of learning complex patterns at multiple levels to solve a particular task, e.g. phone prediction. Keeping these points in mind, we propose a neural network based model (MLP-FA) for multiple latent variable analysis applied to tandem speech recognition [11] in this paper. We borrow a five layer Multi-Layer Perceptron (MLP) bottleneck architecture [12], with compression and prediction stages, as the basic latent variable model and we extend it to deal with multiple sources of variability. This approach has some divergences with the statistical models mentioned above but it still provides a form of subspace based representation that preserves the desired variabilities assumed to be present in the observations. While essentially bringing analogous processing power compared to FA or SGMM, a neural net architecture offers some advantages such as the straightforward use of discriminative criteria in the parameter optimization process at the price of having data labels available for each source of variability.

The paper is organized as follows: Section 2 introduces the Factor Analysis framework in the speaker recognition context. Section 3 makes an analogy of factor analysis using a MLP based implementation with Section 3.1 focusing on the training of a phone-speaker MLP. Sections 4 and 5 desccribe the experimental setup and the results for the phone-speaker MLP-FA experiments. Canclusions are presented in Section 6.

2. FACTOR ANALYSIS

Factor Analysis (FA) is a statistical tool that aims at explaining the correlation amongst a set of measured variables in terms of a smaller number of latent variables plus modeling errors. The latent variables (or factors), that are unknown although assumed to be independently and normally distributed, are linearly combined to predict the measured variables. For speech applications, it is common to estimate

This work was supported by the European Union under the FP7 Integrated Project inEvent (Accessing Dynamic Networked Multimedia Events), grant agreement 287872. The authors gratefully thank the EU for their financial support and all project partners for a fruitful collaboration.

the parameters involved in such linear combination so that the likelihood function of the speech data given a GMM is maximized. We consider the model

$$\mathbf{z} = \mathbf{m} + \mathbf{U}\mathbf{x} + \boldsymbol{\epsilon} \tag{1}$$

with the measures in vector $\mathbf{z}$ being the mean vectors of the GMM, $\mathbf{m}$ being a constant vector and $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ with $\boldsymbol{\Psi} = \boldsymbol{\delta}\mathbf{I}$. Note that the correlation between the variables in $\mathbf{z}$ is accounted for by the factor loading matrix $\mathbf{U}$ only.

This framework has been successfully used for the adaptation of a GMM-Universal Background Model (UBM) in classification tasks such as speaker recognition [2, 3]. The measured variables $\mathbf{z}$ represent the speaker-adapted mean vectors from the GMM-UBM with mean parameters $\mathbf{m}$ and $\mathbf{U}\mathbf{x}$ represent a speaker dependent term that moves $\mathbf{z}$ towards the speaker data. In the training phase, both $\mathbf{U}$ and $\mathbf{x}$ are estimated to maximize the likelihood of the adapted model using the speech data for each speaker in a database. An Expectation-Maximization (EM) algorithm [9] alternately estimates the posterior distribution of $\mathbf{x}$ and then uses matrix regression to improve the estimate of $\mathbf{U}$. Only the posterior distribution of $\mathbf{x}$ is performed in the adaptation phase assuming $\mathbf{U}$ is fixed. As many parameters as the dimension of the subspace are thus estimated in the adaptation phase.

However, the success of FA lies on separately modeling different variabilities by using separate low-rank terms. In speaker recognition, it is common to use two factor loading matrices $\mathbf{U}$ and $\mathbf{V}$ defining the session and speaker subspaces respectively. In this case, we can use the model

$$\mathbf{z} = \mathbf{m} + \mathbf{U}\mathbf{x} + \mathbf{V}\mathbf{y} + \boldsymbol{\epsilon} \tag{2}$$

where the $\mathbf{U}\mathbf{x}$ and $\mathbf{V}\mathbf{y}$ terms are fit to the session and speaker variabilities by using the session- and speaker-wise sufficient statistics respectively. Speaker models that are more robust to session variability can then be obtained as $\mathbf{m} + \mathbf{V}\mathbf{y}$.

Note that FA explicitly models the error $\boldsymbol{\epsilon}$ as possibly having a different variance for each observation in $\mathbf{z}$, as opposed to Principal Component Analysis (PCA) where all errors have the same unity variance. In practice, the variability accounted by this term is small, especially if the low rank terms are estimated first.

3. MLP-BASED FACTOR ANALYSIS

The variability model of Eq. 2 assumes linearity in the mean supervector space, which translates into a sort of piece-wise linear transformation. Analogously, Multilayer Perceptron (MLP) neural networks are able to approximate any function, although in a continuous manner. Switching from the FA model above to an MLP model, with its side effects in terms of training criterion and error modeling, is our proposal in this paper.

The EM algorithm used for FA training involves two steps that are iterated: a) factor estimation, and b) factor loading matrix estimation. The proposed MLP-FA approach unfolds these two steps onto a single feedforward architecture, computing the factors using a non-linear function, e.g. f_1, and then predicting the outputs using another non-linear function, e.g. f_0, having the same role as the factor loading matrices in FA. The feedforward structure can be as simple as a 5-layer MLP with a bottleneck layer, involving two 3-layer MLPs that can approximate arbitrary non-linear functions each, namely f_1 and f_0. For a given an input vector $\mathbf{v}$ the factors $\mathbf{x}$ are computed, and the MLP model can be written by

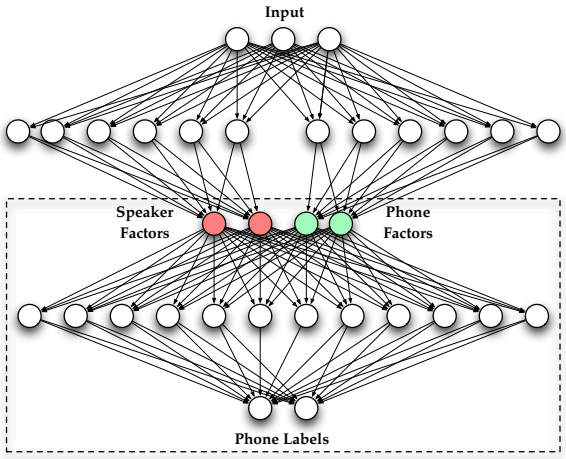

Fig. 1. MLP-FA architecture for phone prediction using phone-speaker factors. During predictor estimation, only the parameters of the second 3-layer MLP are trained.

$$\mathbf{z} = f_0(\mathbf{x}) + \boldsymbol{\epsilon} \quad \text{with } \mathbf{x} = f_1(\mathbf{v}) \tag{3}$$

In an autoassociative setup, such an architecture has been shown to succesfully perform Non-linear Principal Component Analysis [13], i.e. minimizing the reconstruction mean square error. If the outputs are set to be class labels it can also be used for classification, which is indeed our goal. In such case, we typically use softmax activation functions at the output layer and cross-entropy error function as the training criterion. The compressed layer uses linear activation functions to mimic FA. Such a model is equivalent to that of Eq. 1 except that the errors $\boldsymbol{\epsilon}$ are not explicitly modeled, so they might end up being modeled by f_0, as in Principal Component Analysis (PCA). In order to mimic the FA assumptions we use a zero-mean and unit-variance normalized linear bottleneck layer that can further provide better initialization of the parameters for training.

Eq. 3 accounts for one source of variability whose low-dimensional representation is $\mathbf{x}$. It is straighforward to extend the framework to include more than one hidden vectors $\mathbf{x}_i$ with $1 \leq i \leq F$ as

$$\mathbf{z} = f_0(\mathbf{x_1}, \ldots, \mathbf{x_F}) + \boldsymbol{\epsilon} \quad \text{with } \mathbf{x_i} = f_i(\mathbf{v}) \tag{4}$$

where the function f_0 has the role of merging the hidden vectors to predict the desired output. The input to f_0 is the output of several functions f_i each trained to model a different type of variability. This architecture is shown in Fig. 1.

In conclusion, MLP-FA constrains the 5-layer MLP architecture to explain different variabilities while letting the parameters adapt automatically to optimize the training criterion.

3.1. Training phone-speaker MLP-FA

The MLP-FA architecture of Eq. 4 itself can be seens as a constrained 5-layer MLP, where the compressed layers are trained to represent the desired variabilities while letting the network parameters adapt automatically to optimize the training criterion.

If only one source of variability, e.g. phone variability, is present, both f_1 and f_0 functions can be jointly optimized by using the backpropagation algorithm with the cross-entropy criterion using data that effectively samples the phone variability. If the training data samples both phone and speaker variabilities and both phone and speaker labels are also available we can train a phone-speaker MLP-FA model using the following two-step training procedure, that is also based on backpropagation:

- **Factor estimation**: Two 5-layer MLPs are trained to predict phone and speaker labels respectively from the training data. We retain the parameters of the first 3-layer sections, i.e. up to the bottleneck layer, for each MLP. The resulting 3-layer MLPs compute two compressed representations, e.g. functions f_p and f_s, that preserve inter-phone and inter-speaker variabilities. The process is illustrated in Fig. 2 where the dashed line wraps the parameters being trained and grayed layers are dropped after training.
- **Factor Loading Function estimation**: The outputs of f_p and f_s are merged into single compressed vectors retaining both inter-phone and inter-speaker variabilities. An additional 3-layer MLP merging both vectors to predict the phone labels is trained. Fig. 1 illustrates this step with only the parameters of the 3-layer MLP being trained. Note that this 3-layer MLP has more inputs and more parameters than the original 3-layer MLPs.

Such an approach to training can host any number of variabilities into the same framework as long as the corresponding labels are available and the training data samples such variabilities.

The main advantage of MLP-FA over a regular MLP when both are targetting phone prediction is the use of supervision they make. A single MLP is supervised using phone labels only, while the presented MLP-FA is supervised twice, with phone and speaker labels. In a single MLP, the interaction between phone and speaker variabilities has to be discovered in a non supervised manner using a phone discriminating criterion only, which means interpolation in practice. In MLP-FA, supervision affects the phone and speaker compressed layers but also the output phone classes.

4. TASK AND EXPERIMENTAL SETUP

We address the continuous speech recognition task evaluated on the Wall Street Journal (WSJ) English data, involing read speech recording in clena conditions. Our recognition system uses tandem features obtained from a phone posterior probability estimator and modeled using phonemic HMM.

The phone posterior probabilities are computed using either MLP or MLP-FA models with a sliding window of 9

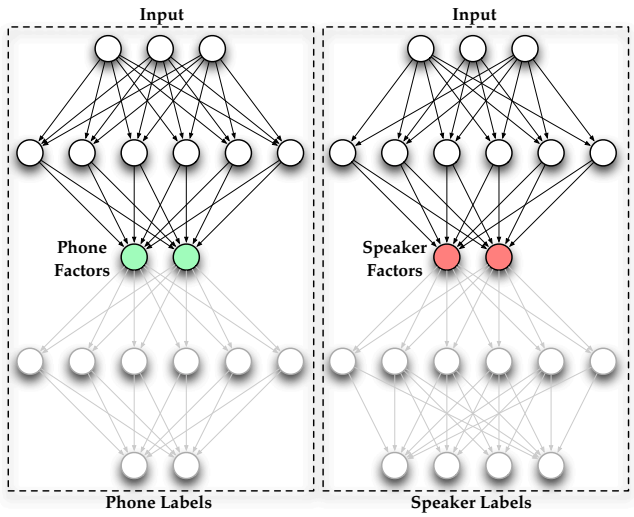

Fig. 2. Training the phone and speaker MLPs separately. The parameters enclosed by the dashed line are trained. Grayed out units are not retained for further processing.

contiguous frames of 12 MFCC plus energy plus Δ and $\Delta\Delta$ features, for a total of 351 inputs. The backpropagation algorithm maximizing the cross-entropy error function is used to train the parameters of the MLP with 40 English phone labels for each file of the training data. The Quicknet toolkit [14] was adapted to use a normalized linear bottleneck layer for this purpose. These labels are obtained by aligning the training data against the manual word transcriptions using auxiliary HMM trained on the acoustic features described above. The MLP-FA parameters are trained as described in Section 3.1 using both speaker and phone labels for the training corpus. Both MLP and MLP-FA models use a 5-layer bottleneck architecture where the bottleneck units use a linear activation function. An additional normalization constraint is enforced on each of these units to have zero mean and unit variance over the training data at each training epoch. The first and second 3-layer sections use tan sigmoid functions and the output units use softmax functions. The training data is taken from the si_tr_200 and the si_tr_84 WSJ training data sets involving around 80 hours of speech and 284 speakers. We keep 10% of the available data as the development set and 90% as the training set. We us the Frame Error Rate evaluated on the development set to assess the performance of the MLP.

A logarithm function Gaussianizes the phone posterior features before being modeled by phonemic HMM. These are intra-word triphones with 16 Gaussian mixtures as observation densities for each of the 3000 tied states ($\sim$19000 triphones). The HMM parameters are trained using maximum-likelihood estimation. We use the HDecode large vocabulary together with a decoding lexicon with around 23000 words and the large vocabulary tri-gram language models provided in the WSJ corpus. We use the Word Error Rate (WER) as the measure to evaluate recognition performance of these systems.

The evaluation data of the WSJ corpus poorly samples

speaker variability, as they involve 8 speakers only. Since MLP-FA models phone as well as speaker variabilities explicitly we extended the evaluation data set by joining all individual data sets of the speaker independent evaluation condition, i.e. si_et_05, si_et_20, si_et_h1 and si_et_h2. This data set involves 1091 utterances and 18 speakers, compared to around 300 utterances and 8 speakers for the individual data sets.

5. RESULTS

We ran experiments comparing the performance of the tandem ASR systems using MLP and MLP-FA models. For further reference we also set a baseline system directly modeling the MFCC features with HMM trained in the same way as those used for the tandem systems.

We set the total number of parameters of the 5-layer MLP to the 10% of the total number of frames in the training data. We further assume that hidden layers 1 and 3 have the same number of units and we keep that number (5320) fixed for other MLP experiments. This allows to control the complexity of the MLP and to compare their performance in a meaningful way.

The first set of experiments allowed the optimal number of phone factors, i.e. bottleneck layer units, to be found for the 5-layer MLP. Their results are shown in the second part of Table 1. We manually explored several plausible values and took the one minimizing the FER on the development data. Using 50 phone factors minimizes such FER and also the WER, whereas a larger or smaller number of units increases both errors. In these experiments, FER and WER are correlated, so the smaller the FER the smaller the WER. The absolute WER obtained by the 5MLP 50ph system compares favorably to the MFCC baseline system and also to 3MLP, a tandem system using a more standard 3-layer MLP architecture with the same number of parameters. Note, however, that the FER of the 3MLP system is actually smaller than that of 5MLP 50ph. We believe this is related to the shape of the MLP output distributions. Predicting phone labels originating from a linear layer might result in smoother distributions than if they originate from a MLP with only sigmoid units. In this series of experiments we also tried to assess the effect of applying speaker-wise mean and variance normalization on the MFCC features input to the MLP. The results for the 5MLP 50ph mvn system obtain higher FER and WER than the other systems, suggesting that MLP actually exploit such information and that they do more efficiently.

A second set of experiments assesses the effect of explicitly modeling the speaker variability using the MLP-FA model on the ASR performance. MLP-FA partly uses MLP trained for phone and speaker discrimination. We choose the 5MLP 50ph system that obtains the smallest FER and WER. The third part of Table 1 shows results for three speaker discrimination experiments using a 5-layer MLP and 60, 125 and 250 speaker factors respectively. Even if these MLP discriminate speech frames from the 284 speakers of the training set, 60 factors are enough to obtain around 25% FER, the smallest error rate amongst the three speaker MLP shown in the table. In the lowest part of Table 1 we present results for the MLP-FA based systems using 50 phone factors and either 60, 125 or 250 speaker factors. These three MLP-FA setups obtain significantly lower FER compared to the MLP model alone with a maximum relative gain of around 9% FER, which corresponds to 6.5% WER, for the 5MLP-FA 50ph+125spk system. The second column of Table 1 shows the number of parameters trained using phone and speaker labels respectively. Still the number of parameters is comparable. The difference between the number of parameters used by 5MLP-FA 50ph+125spk (2.9M) and 5MLP 50ph (2.6M) are the weights that merge the speaker and phone factors to predict the phone labels. Note that the best MLP-FA performance is not obtained with the best performing speaker MLP, which might reflect complex interactions between the phone and speaker variabilities.

System	#Mprm	FER(%)	WER(%)
MFCC	—	—	9.79
3MLP	2.6/—	24.28	9.88
5MLP 30ph	2.4/—	25.44	10.01
5MLP 50ph	2.6/—	25.18	9.66
5MLP 70ph	2.8/—	26.03	10.05
5MLP 50ph mvn	2.6/—	26.98	9.95
5MLP 60spk	—/1.6	25.09	—
5MLP 125spk	—/2.1	28.34	—
5MLP 250spk	—/2.6	39.85	—
5MLP-FA 50ph+60spk	2.9/1.0	23.01	9.27
5MLP-FA 50ph+125spk	3.3/1.1	22.91	9.01
5MLP-FA 50ph+250spk	3.9/1.3	23.22	9.27

Table 1. Frame Error rate (FER) of phone and speaker posterior MLPs and Word Error Rate (WER) of several LVCSR systems using tandem features from 3-layer and 5-layer bottleneck MLP. The second column shows the number of parameteres trained using phone/speaker labels (in millions).

6. CONCLUSION

We proposed a novel approach to multiple factor analysis based on a Multilayer Perceptron architecture (MLP-FA) that mimics factor analysis and we evaluated it on the task of phone label prediction in a tandem ASR system. The phone labels are predicted by a 5-layer bottleneck MLP that is constrained to use low dimensional representations of phone and speaker variabilities. A simple training procedure that uses phone and speaker target labels and the backpropagation algorithm are used for this purpose. First, the experiments on the WSJ corpus showed that tandem systems using a 5-layer MLP architecture with a normalized linear bottleneck layer can outperform 3-layer MLP and a system using MFCC features alone. Regarding the MLP-FA architecture, merging phone and speaker low-dimensional vectors, obtained 9% of FER improvement over the phone MLP and over 6.5% of WER gain. This technique is fairly simple to implement and can be adapted to modeling any number of sources of variability as long as labels are available.

7. REFERENCES

[1] C. M. Bishop, "Latent Variable Models," in *Learning in Graphical Models, M. I. Jordan Ed.*, 1999, pp. 371–403.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980–988, 2007.

[3] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2008.

[4] N. Dehak, R. Dehak, P. Kenny, N Brummer, P. Ouellet, and P. Dumouchel, "Support Vector Machines versus Fast Scoring in the Low-Dimensional Total Variability Space for Speaker Verification," in *Proc. INTERSPEECH*, 2009, pp. 1559–1562.

[5] S. J. D. Prince, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *Proc. IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[6] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. of the IEEE Speaker Odyssey Workshop*, June 2010.

[7] Dan Povey and Lukas Burget, "The Subspace Gaussian Mixture Model a Structured Model for Speech Recognition," *Computer Speech and Language*, 2011.

[8] M. J. F. Gales and S. J. Young, "Multiple-cluster Adaptive Training Schemes," in *Proc. IEEE ICASSP*, 2001, pp. 361–364.

[9] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[10] G. Hinton, L. Deng, G. E. Dahl, A. Mohammed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[11] D.P.W. Ellis, R. Singh, and S. Sivadas, "Tandem acoustic modeling in large-vocabulary recog- nition," in *Proc. IEEE ICASSP*, 2001, pp. 517–520.

[12] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and Bottle-Neck Features for LVCSR of meetings," in *Proc. IEEE ICASSP*, 2007, pp. 757–760.

[13] M. A. Kramer, "Nonlinear Principal Component Analysis using Autoassociative Neural Networks," *American Institue of Chemical Engineers Journal*, vol. 37, pp. 233–243, 1991.

[14] "Quicknet software," `http://www1.icsi.berkeley.edu/Speech/qn.html`, Accessed November 25, 2012.