On Speaker Adaptive Training of Artificial Neural Networks

Jan Trmal, Jan Zelinka, Luděk Müller

Department of Cybernetics, University of West Bohemia, Czech Republic

jtrmal@kky.zcu.cz, zelinka@kky.zcu.cz, muller@kky.zcu.cz

Abstract

In the paper we present two techniques improving the recognition accuracy of multilayer perceptron neural networks (MLP ANN) by means of adopting Speaker Adaptive Training. The use of the MLP ANN, usually in combination with the TRAPS parametrization, includes applications in speech recognition tasks, discriminative features production for GMM-HMM and other. In the first SAT experiments, we used the VTLN as a speaker normalization technique. Moreover, we developed a novel speaker normalization technique called Minimum Error Linear Transform (MELT) that resembles the cMLLR/fMLLR method [1] with respect to the possible application either on the model or features.

We tested these two methods extensively on telephone speech corpus SpeechDat-East. The results obtained in these experiments suggest that incorporation of SAT into MLP ANN training process is beneficial and depending on the setup leads to significant decrease of phoneme error rate (3 % – 8 % absolute, 12 % – 25 % relative).

Index Terms: speaker adaptive training, SAT, TRAPS, VTLN, neural network, phoneme recognition

1. Introduction

The Speaker Adaptive Training approach is used routinely during the construction of state-of-the-art GMM-HMM recognition systems. Application of SAT leads to more robust performance characteristics, lower complexity of models and increased recognition performance. On the other hand, the recognition phase is slightly more complex and an additional diarization module is needed.

On the other hand, according to our knowledge, no research directed on SAT in the field of neural networks has been reported. Therefore, we decided to investigate this task and find out, if the proposed approach is viable. Besides answering the question of recognition accuracy improvement, we tried find answers to the following questions:

- Is it reasonable to assume that the ANN itself is able to learn "how to normalize" the speakers?
- If it is so, to what extent? Is the normalization ability backed up by the size of the ANN?
- Is the size of ANN the only factor limiting the generalization of the ANN? Does the TRAPS frame [2] (spanning about 300 ms) contain all the information needed for speaker normalization?

2. TRAPS parametrization

The used TRAPS feature vectors are constructed from the log-output of mel-filter bank. The intention of this section is only to explain the terms used further in this paper, the full process of the construction is described in detail in [3].

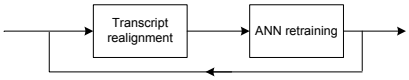

Figure 1: A scheme of training of SI model

Assume for the given k-th frame of speech, the vector of mean-normalized log-outputs from the mel-filter bank is $\boldsymbol{c}(k) = [c_1(k), c_2(k), \ldots, c_N(k)]$, where N is the number of frequency bins and K is the total number of feature vectors.

The vector of D consecutive outputs of the p-th filter bank $\boldsymbol{c}_p(k) = [c_p(k-D+1), \ldots, c_p(k)]$, $p = 1, \ldots, N$ is then decorrelated by a $D \times N_\Psi$, $N_\Psi \leq D$ matrix $\boldsymbol{\Psi}$

$$\tilde{\boldsymbol{c}}_p(k) = \boldsymbol{c}_p(k) \cdot \boldsymbol{\Psi} \qquad p = 1, \ldots, N \tag{1}$$

Usually, the $\boldsymbol{\Psi}$ matrix is a discrete cosine transform matrix. The vectors $\tilde{\boldsymbol{c}}_p(k), p = 1, \ldots, N$, are merged together, yielding the TRAPS vector $\tilde{\boldsymbol{c}}(k)$ of size M, $M = N_\Psi N$, $\tilde{\boldsymbol{c}}(k) = [\tilde{\boldsymbol{c}}_1(k), \ldots, \tilde{\boldsymbol{c}}_p(k), \ldots, \tilde{\boldsymbol{c}}_N(k)]$. The vector $\tilde{\boldsymbol{c}}(k)$ is then used as the input $\boldsymbol{x}(k)$ in eq. (2).

The TRAPS feature vectors are usually quite long, since they span several hundreds of milliseconds of the original acoustic track. The features are fed into the ANN trained to produce phoneme posterior probabilities.

3. Multi-layer perceptron artificial neural network

Any forward operation of a L-layer MLP ANN can be described as follows

$$\boldsymbol{a_0}(k) = \boldsymbol{x}(k)\boldsymbol{W}_0 \tag{2}$$

$$\begin{aligned} \boldsymbol{y}_i(k) &= \vec{g}_i(\boldsymbol{a}_{i-1}(k)) \\ \boldsymbol{a}_i(k) &= \boldsymbol{y}_i(k)\boldsymbol{W}_i \end{aligned} \qquad i = 1, \ldots, L-1 \tag{3}$$

$$\boldsymbol{z}(k) = \vec{g}_L(\boldsymbol{a}_{L-1}(k)) \tag{4}$$

where the $D_i \times D_{i+1}$ matrices $\boldsymbol{W}_i$, $i = 0, \ldots, L-1$, are called weight matrices and the vector functions $\vec{g}_i$, $i = 1, \ldots, L-1$, are called transfer functions. The weight matrices are trained to minimize a loss function $\boldsymbol{E}$ that is usually of the following form

$$\boldsymbol{E}(\boldsymbol{Z}, \boldsymbol{T}) = \sum_{k=0}^{K} E(\boldsymbol{z}(k), \boldsymbol{t}(k)) \tag{5}$$

where the $K \times M$ matrix $\boldsymbol{Z}$ represents network outputs and the $K \times M$ matrix $\boldsymbol{T}$ represents the target values (teacher data). The pair of k-th rows of the matrices $\boldsymbol{Z}$ and $\boldsymbol{T}$ represents the k-th output $\boldsymbol{z}(k)$ and the target vector $\boldsymbol{t}(k)$.

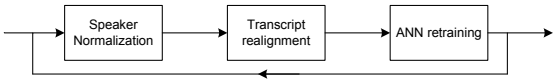

Figure 2: A scheme of training of a SAT model

The most usual choices of function E are E_{MSE} (i.e. mean square error) or cross-entropy E_{XENT}. Usually, the MLP ANN for phoneme recognition has sigmoid transfer function in hidden layers and softmax transfer function in the output layer and are trained using E_{XENT} criterion. Such configuration guarantees that MLP ANN output will converge to posterior probabilities. See [4] for more info.

For the training there is a wide variety of methods to use. The most common one is the backpropagation and its modifications.

4. Speaker Normalization

There exists a significant gap between recognition scores of speaker-independent (SI) models and speaker-dependent (SD) models — the SD models reportedly perform better. Obviously, the reason of this gap lies in the speaker variability. The variability comprises the linguistic background, emotional state and physical attributes of the speaker.

Modeling the variability among speakers within the SI model is a more complex task than to use the SD model. This increased complexity leads to increased training and more importantly increased recognition time. Moreover, despite of the increased complexity, the performance of the SI model is often inferior.

4.1. Vocal Tract Length Normalization

One of the causes of speaker variability is a different length of the vocal tract. The different length of the vocal track manifests itself in shift of the voice pitch in the frequency spectrum. There is a variety of methods that deal with this problem.

The common one is called VTLN (Vocal Tract Length Normalization). The application of this method is usually tied to use of MFCC coefficients, albeit the principle is general. During the computation of centers of frequency bins of the mel-filter bank, the computed centers are shifted (warped) in a non-linear fashion to reflect the shift of the voice pitch of the given speaker relative to the "normalized" speaker.

Usually, the transforms is expressed as $\tilde{\omega} = f(\omega, \alpha_s)$, where ω is the original frequency in the mel-domain, $\tilde{\omega}$ is the transformed mel-frequency and α_s is the speaker-dependent normalization factor. The function $f()$ is usually *piecewise linear transform* [5] or *bilinear transform* [6].

For our experiments, we used the bilinear transform. The bilinear transform is defined as

$$\tilde{\omega} = \omega + 2\arctan\left(\frac{(1-\alpha)\sin\omega}{1-(1-\alpha)\cos\omega}\right) \tag{6}$$

and the value α is usually determined by means of grid-search on the space of utterance likelihood.

The VTLN is reported to be used routinely in TRAPS-MLP framework [3], however, to our best knowledge, it is used as a speaker normalization technique, without the SAT complement. We believe that the VTLN factors are determined using a standalone classifier, without the iterative re-assignment optimizing the real likelihood of the hybrid ANN-HMM recognition system.

The application of VTLN factors during the TRAPS construction is straightforward, since the only change is made in the filter-bank setup phase. In this phase, the filters are shifted according to the chosen transform and factor. The rest of the parametrization process remains unchanged.

4.2. Minimum Error Linear Transform for MLP ANN

It has been shown [7] that VTLN can be represented as a linear transform of the original unnormalized coefficients. Therefore, the linear transform itself can be used for speaker normalization, even without explicit link to VTLN. The main difference is that instead of shifting the frequency banks (and thus changing the parametrization process), the linear transform works on feature level. Moreover, since the number of free variables is bigger, the normalization ability might be better.

Let's express the matrix $\boldsymbol{W}_0$ in eq. (2) as

$$\boldsymbol{W}_0 = \boldsymbol{\Gamma}\boldsymbol{W}'_0 \tag{7}$$

where $\boldsymbol{W}'_0$ is the original, unnormalized weight matrix of SI ANN and $\boldsymbol{\Gamma}$ is a a $D_0 \times D_0$ *normalization* matrix that must be determined during the speaker normalization phase of the SAT process.

Suppose that for a given set of utterances, for the given speaker, we want to find the matrix $\boldsymbol{\Gamma}$ that minimizes the criterion eq. (5), therefore need to solve the following equation

$$\sum_{k=1}^{K} \frac{\partial E(k)}{\partial \boldsymbol{\Gamma}} = 0 \tag{8}$$

The gradients can be obtained using a similar approach as used in backpropagation.

Using the equations eq. (7) and eq. (2) – eq. (4) and applying the matrix derivative chain rule, we arrive to the following expression (for simplicity, we drop the index k)

$$\frac{\partial E}{\partial \boldsymbol{\Gamma}}^{\mathrm{T}} = \left(\frac{\partial E}{\partial \boldsymbol{z}}\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{a}_{L-1}}\prod_{i=L-1}^{1}\left(\boldsymbol{W}_i^{\mathrm{T}}\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{a}_{i-1}}\right)\boldsymbol{W'}_0^{\mathrm{T}}\right)^{\mathrm{T}}\boldsymbol{x} \tag{9}$$

The derivatives $\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{a}_{i-1}}$ and $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{a}_{\boldsymbol{L-1}}}$ are $D_i \times D_i$ ($D_L \times D_L$ respectively) matrices. For the matrix $\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{a}_{i-1}}$, the element σ_{ab} at coordinates (a, b) is given by

$$\sigma_{ab} = y_a\delta_{ab} - y_a y_b \tag{10}$$

in case when g_i is a softmax transfer function and

$$\sigma_{ab} = \delta_{ab} y_a(1 - y_b) \tag{11}$$

in case when g_i is a sigmoidal transfer function. Analogously for the matrix $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{a}_{\boldsymbol{L-1}}}$.

For the error function expressions $\frac{\partial E}{\partial \boldsymbol{z}}$ the following expressions hold

$$\frac{\partial E_{\mathrm{XENT}}}{\partial z_{ij}} = -\delta_{ij}\frac{t_i}{z_j} \tag{12}$$

$$\frac{\partial E_{\mathrm{MSE}}}{\partial z_{ij}} = t_i - z_j \tag{13}$$

We call the matrix $\boldsymbol{\Gamma}$ determined using the aforementioned approach the Minimum Error Linear Transform (MELT) matrix. The difficulty with the direct application of this algorithm lies in the size of the matrix $\boldsymbol{\Gamma}$, i.e. in the number of its free parameters. As already mentioned, the TRAPS usually span quite

a long temporal window and the dimension of feature vector is relatively high (several hundreds). On the other hand, it is not uncommon that during training the amount of data belonging to one speaker is quite small (tens or hundreds of seconds). Therefore, the direct use of MELT training algorithm would lead inevitably to overtraining. This problem might be overcomed by clustering the speakers and using Cluster-Adaptive Training (CAT) instead of SAT, but this is not the way we want to go.

The solution lies in reduction of number of free parameters by expressing the matrix $\mathbf{\Gamma}$ as a function of S-dimensional vector $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_S]$, $\mathbf{\Gamma} = \mathbf{\Gamma}(\boldsymbol{\gamma})$, where $S \ll L_0 \times L_0$ and optimizing the $\frac{\partial E}{\partial \boldsymbol{\gamma}}$ instead of $\frac{\partial E}{\partial \mathbf{\Gamma}}$ Using the matrix calculus we get

$$\frac{\partial E}{\partial \gamma_i} = \mathrm{Tr}\left[\left(\frac{\partial E}{\partial \mathbf{\Gamma}}\right)^{\mathrm{T}} \frac{\partial \mathbf{\Gamma}}{\partial \gamma_i}\right] \qquad \text{for} \quad i = 1, \ldots, S \qquad (14)$$

where the computation of the expression $\frac{\partial \mathbf{\Gamma}}{\partial \gamma_i}$ is straightforward, since by definition the $\Gamma_{kl}(\boldsymbol{\gamma})$ is known for every element Γ_{kl} of the matrix $\mathbf{\Gamma}$. Then, instead of solving the eq. (8), we solve the following equation

$$\frac{\partial E}{\partial \boldsymbol{\gamma}} = 0 \qquad (15)$$

Unfortunately, there is no simple nor general way of designing the relation $\mathbf{\Gamma} = \mathbf{\Gamma}(\boldsymbol{\gamma})$. We experimented with the following approach. Considering that the TRAPS features are constructed from log-outputs of mel-filter bank, we have decided to establish a link between the log-outputs interpolation and the final TRAPS features transformation.

The normalized bin output $c_i(k)$ (using the notation from section discussing the TRAPS construction) is obtained from the old frequency bin output $c'_i(k)$.

$$c_i(k) = \sum_{j=1}^{N} \gamma_{ij} c'_j(k) \qquad i = 1, \ldots, N \qquad (16)$$

Following the modified TRAPS construction process presented in Section 2, we arrive to the following matrix[1]

$$\mathbf{\Gamma} = \begin{pmatrix} \boldsymbol{\gamma}_{11} & \cdots & \boldsymbol{\gamma}_{1N_\Psi} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\gamma}_{N_\Psi 1} & \cdots & \boldsymbol{\gamma}_{N_\Psi N_\Psi} \end{pmatrix} \qquad (18)$$

and the matrix element $\boldsymbol{\gamma}_{ik}$ represents a $N \times N$ diagonal matrix

$$\boldsymbol{\gamma}_{ik} = \gamma_{i \cdot N_\Psi + k} \mathbf{I} \qquad (19)$$

where $\mathbf{I}$ is an $N \times N$ identity matrix and the expression $i \cdot N_\Psi + k$ maps the 2D coordinates (i, j) into the vector $\boldsymbol{\gamma}$.

5. Experiments

5.1. Speech Corpus

For the experiments we used the SpeechDat-East telephone speech corpus. The corpus contains recordings of approximately 1000 speakers. For training the ANN, we used only the phonetically balanced sentences (ID S0-S9, X0-1). Note that some speakers are not represented by a complete set of the 12 phonetically balanced sentences. Moreover, since there is a large portion of silence in the recordings, we removed it. The training set contains recordings from approx. 900 speakers and the testing set contains recording from 200 speakers.

Our phonetic alphabet contained 37 three-state phonemes. The recognition was done on a hybrid ANN-HMM phoneme recognizer system. To eliminate the influence of language model, we used the zerogram language model.

[1] The original, "full" construction process leads to a matrix

$$\mathbf{\Gamma'} = \begin{pmatrix} \mathbf{\Gamma} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma} \end{pmatrix} \qquad (17)$$

where the matrix $\mathbf{\Gamma'}$ is then used as the matrix $\mathbf{\Gamma}$ in eq. (7) and the matrix $\mathbf{\Gamma}$ is the matrix above.

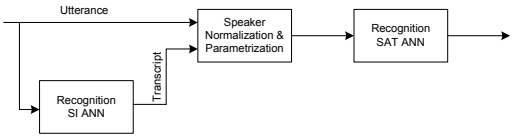

Figure 3: A scheme of an one-pass unsupervised recognition using SAT ANN

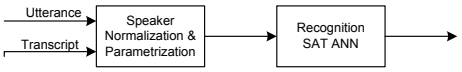

Figure 4: A scheme of an supervised recognition using SAT ANN

5.2. Speaker Independent ANN

For the training of the SI ANN we split the training set into a training set and a validation set. Using the training data, we trained two networks: first with 1500 neurons in the hidden layer (resulting in approx. 600k parameters in total) and the second with 2500 neurons in the hidden layer (1.1M parameters in total). See Fig. 1 for the scheme of SI ANN training. These ANN's are denoted as "base" in the tables Tab. 1, Tab. 2 and Tab. 3.

5.3. Speaker Adaptive Training of ANN

For the SAT ANN training, we used the SI ANN obtained in the previous phase as the startup initialization. Using this network, we determined the VTLN α coefficient using grid-search or MELT matrices using the previously described algorithm. After normalization of speakers we re-trained the ANN and the resulting ANN was used again in the speaker normalization phase. See Fig. 2 for the scheme of SAT training.

5.4. Recognition Experiments

During recognition, the assumption of availability of reference phone level transcript may or may not be true. We call the case when the reference transcript is available as "supervised normalization", the other case is called "unsupervised normalization". We evaluated both these situations.

5.4.1. Supervised Normalization

For the first set of experiments we used the reference transcript of the testing data. This assumption is not as strong as it may seem. During the adaptation phase, the speaker can be asked to read some prepared, preferably phonetically balanced text sentences or just some random text downloaded from the internet. In both these cases, the phone-aligned transcript can be created relatively easily. The scheme of this recognition pro-

	Iteration number				
	base	1.	2.	3.	4.
ANN-1500	76.64	79.98	81.30	81.79	82.01
ANN-2500	79.25	81.42	81.95	82.01	82.07

Table 1: SAT-VTLN performance (ACC), when the reference transcript is known

	Iteration number				
	base	1.	2.	3.	4.
ANN-1500	76.64	81.75	82.57	82.89	83.01
ANN-2500	79.25	83.03	83.45	83.60	83.60

Table 2: SAT-MELT performance (ACC), when transcript is known

cess is depicted in Fig. 4. The results from this experiments are shown in tables Tab. 1 and Tab. 2. We tested the performance of SAT ANN-HMM system after each SAT iteration to see how the training process converges.

5.4.2. Unsupervised Normalization

Next, we performed preliminary experiments to verify if some improvements can be achieved even without availability of the reference transcription. Using the SI ANN, we recognized the testing data. The recognized output was then used as a reference phone alignment and the computation of normalization factors was performed on the recognition result instead of on the reference phonetic transcript. See Fig. 3 for the scheme of this experiment.

6. Conclusion

In this paper we presented speaker normalization and speaker adaptive training enhancement of the TRAPS-MLP system. Incorporation of these techniques can greatly improve the recognition accuracy. According to our experiments, the improvement can achieve 25 % of reduction relative error rate when the reference phonetic transcript is available and by 12 % when a very crude unsupervised method is used. We believe that using more elaborate unsupervised approach would increase the recognition even further. However, evaluation and development of such techniques remains a task for further research.

Regarding to the questions asked in the introduction we can conclude the following: From the performance scores of both ANN's on SAT-VTLN task follows that the MLP ANN is able, to some extent, to cope with the variability of speakers and this ability is backed up by the size of the network. However, the upper bound of its normalization ability is close to SAT-VTLN performance (since after VTLN normalization, the ANN-2500 did not perform better than ANN-1500, i.e. it was not able to re-use it's additional free learning capacity to normalize the speaker even more). Second, a single TRAPS frame does not contain sufficient information to perform full speaker normalization. This assertion results from the fact that the SAT-trained ANN's performed better than the SI ANN's. From the fact that the SAT-MELT method systematically outperformed the SAT-VTLN we can conclude that the VTLN normalization is not able to remove all the speaker variability. In other words, the speaker variability is not characterized only by the length of the vocal tract.

	Method		
	base	VTLN	MELT
ANN-1500	76.64	80.16	80.20
ANN-2500	79.25	81.20	81.51

Table 3: SAT-MELT and SAT-VTLN performance (ACC) when the reference transcript is unknown

All the experiments presented in this paper were targeted on phoneme recognition in a hybrid MLP-HMM framework and other possible applications were not considered. Note that the phoneme posteriors (or bottleneck features[8] can be further used as features in the GMM-HMM model. In that framework, the Speaker Adaptive Training methods are used routinely. We believe that using the ANN-SAT (bottleneck) features could complement the standard GMM-SAT approach and could lead to improvement of the recognition score; however the evaluation of this gain remains to be performed in our future research.

7. Acknowledgements

This research was supported by the Ministry of Education of the Czech Republic, project No. MŠMT LC536, by the Grant Agency of the Czech Republic, project No. GAČR 102/08/0707 and by the University of West Bohemia, project No. SGS-2010-054. The access to the METACentrum computing facilities provided under the research intent MSM6383917201 is highly appreciated.

8. References

[1] M. Gales and P. Woodland, "Variance compensation within the MLLR framework," *Cambridge University Engineering Department, Tech. Rep. CUED/F-INFENG/TR*, vol. 242, 1996.

[2] H. Hermansky and S. Sharma, "Temporal patterns (TRAPS) in ASR of noisy speech," *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999. ICASSP'99. Proceedings.*, vol. 1, 1999.

[3] P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," *Lecture Notes in Computer Science*, vol. 2004, no. 3206, 2004.

[4] C. M. Bishop, *Neural networks for pattern recognition*. Oxford University Press, ISBN 0-19-853864-2, 2005.

[5] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *Speech and Audio Processing, IEEE Transactions on*, vol. 6, no. 1, pp. 49 –60, jan 1998.

[6] A. Acero, *Acoustical and environmental robustness in automatic speech recognition*. Springer, 1993.

[7] M. Pitz, S. Molau, R. Schlüter, and H. Ney, "Vocal tract normalization equals linear transformation in cepstral space," in *Proceedings of EuroSpeech 2001*, 2001, pp. 2653–2656.

[8] F. Grézl, M. Karafiát, and L. Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Proc. Interspeech 2009*, no. 9. International Speech Communication Association, 2009, pp. 2947–2950.