MAXIMUM LIKELIHOOD NONLINEAR TRANSFORMATIONS BASED ON DEEP NEURAL NETWORKS

Xiaodong Cui, Vaibhava Goel

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

ABSTRACT

This paper investigates modeling nonlinear transformations based on deep neural networks (DNNs). Specifically, a DNN is used as a nonlinear mapping function for feature space transformation for HMM acoustic models. The nonlinear transformations are estimated under the sequence-based maximum likelihood criterion. The likelihood partition function is evaluated using the Monte Carlo method based on importance sampling. The DNN is first pre-trained approximately to a linear transformation then followed by fine-tuning using the gradient descent algorithm. In addition, a deep stacked architecture is proposed that builds a DNN as a series of sub-networks hierarchically with each representing a nonlinear transformation. A block-wise learning strategy is introduced. LVCSR speaker adaptation experiments on the proposed maximum likelihood nonlinear transformation have shown superior results than the widely-used CMLLR transformation.

Index Terms— deep neural networks, nonlinear transformation, maximum likelihood, Monte Carlo method, importance sampling

1. INTRODUCTION

Maximum likelihood linear regression (MLLR) [1][2] has been one of the most successful techniques for speaker or environmental adaptation in automatic speech recognition (ASR). Despite its success, MLLR has its limitations one of which is that the transformation is linear. Artificial neural networks (ANNs) are known as universal approximators [3]. Given the hidden layers with nonlinear activation functions in the neurons, an ANN can give rise to a nonlinear transformation that is more powerful than linear transformations. With the advent of deep neural networks (DNNs) [4], applications using DNNs as nonlinear approximators can be widely found in speech-related areas such as noise robustness [5], speech enhancement [6], voice conversion [7], etc.. This paper investigates nonlinear feature transformations that are based on deep neural networks (DNNs) for HMM-based speech recognition. The nonlinear transformations are estimated under the sequence-based maximum likelihood (ML) criterion and can be considered an extension of the constrained maximum likelihood linear regression (CMLLR)[2] that has been commonly used in the speech community.

Different from CMLLR where the likelihood partition function is analytical due to the linear form of the transformation, the proposed maximum likelihood nonlinear transformation (MLNT) does not have a closed form for the partition function. In this work, the likelihood partition function is evaluated using the Monte Carlo (MC) method based on importance sampling [8][9][10]. Before the ML estimate of MLNT, the networks for the nonlinear transformations are first pre-trained under the minimum mean square error (MMSE) criterion with the CMLLR-transformed target. After the pre-training, fine-tuning is applied where the sequence-based ML estimation is carried out using the gradient descent (GD) algorithm.

We also propose a deep stacked architecture that hierarchically constructs a series of nonlinear transformations in one deep neural network where each sub-network in it serves as a building block representing a nonlinear transformation. A block-wise learning strategy is introduced. In this strategy, each additional sub-network is first initialized to a CMLLR linear transformation by fixing the lower network and taking the output of the lower network as features. Then it is followed by fine-tuning using the GD algorithm under the sequence-based ML criterion.

The remainder of the paper is organized as follows. Section 2 gives the mathematical formulation of the proposed MLNT and the evaluation of the likelihood partition function using importance sampling. It also addresses the MMSE pre-training of the DNNs for nonlinear transformations. Section 3 presents the deep stacked architecture and its block-wise learning strategy. Results of LVCSR speaker adaptation experiments are provided in Section 4 followed by a discussion in Section 5.

2. MATHEMATICAL FORMULATION

Let $\boldsymbol{\mathcal{O}} = \{\boldsymbol{o}_1, \cdots, \boldsymbol{o}_\mathsf{T}\}$ be a feature sequence of an utterance with T frames from a speaker. Suppose we know some hidden Markov model (HMM) acoustic model λ. For simplicity, we assume there is only one Gaussian in the Gaussian mixture model (GMM) distribution in each HMM state. The extension to multiple Gaussians is straightforward. We want to create a transformation such that the transformed feature sequence from this speaker $\hat{\boldsymbol{\mathcal{O}}} = \{\hat{\boldsymbol{o}}_1, \cdots, \hat{\boldsymbol{o}}_\mathsf{T}\}$, where $\hat{\boldsymbol{o}}_t = f(\boldsymbol{o}_t)$, maximizes the following likelihood given the acoustic model

$$f^* = \max_f \log P(\hat{\boldsymbol{\mathcal{O}}}|\lambda, \boldsymbol{\mathcal{O}}). \tag{1}$$

Here the transformation is modeled by a deep neural network as illustrated in Fig.1, where the parameters of the mapping function are the weights $\mathbf{W}$ of the network. The input and output of the network have the same dimensionality. The hidden layers have nonlinear activation functions while the output layer has identity activation functions. The nonlinear transformation is denoted by $f_\mathbf{W}$.

2.1. Gradient

Given the objective function in Eq.1, the optimization is carried out by GD. For each utterance, the likelihood of the transformed feature sequence can be computed as

$$P(\hat{\boldsymbol{\mathcal{O}}}|\lambda, \boldsymbol{\mathcal{O}}) = \sum_{i=1}^{M}\sum_{j=1}^{M} \alpha_{t-1}(j) a_{ji} b_i(\hat{\boldsymbol{o}}_t) \beta_t(i) \tag{2}$$

where M is the number of HMM states; a_{ji} are the state transition probabilities from state j to state i; α and β are forward and backward probabilities; $b_i(\cdot)$ is the observation distribution of state i. It

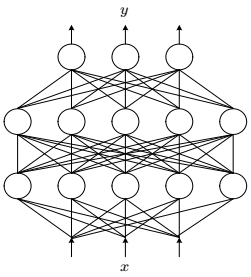

Fig. 1. A deep neural network as a nonlinear transformation function.

can be shown that

$$\frac{\partial \log P(\hat{\boldsymbol{\mathcal{O}}}|\lambda, \boldsymbol{\mathcal{O}})}{\partial \log b_i(\hat{\boldsymbol{o}}_t)} = \gamma_t(i) \tag{3}$$

where $\gamma_t(i)$ is the posterior probability of being in state i at time t. The gradient of the log-likelihood with respect to the weights of the network can be evaluated as follows

$$\begin{aligned}\frac{\partial \log P(\hat{\boldsymbol{\mathcal{O}}}|\lambda, \boldsymbol{\mathcal{O}})}{\partial \mathbf{W}} &= \sum_{t=1}^{T}\sum_{i=1}^{M} \frac{\partial \log P(\hat{\boldsymbol{\mathcal{O}}}|\lambda, \boldsymbol{\mathcal{O}})}{\partial \log b_i(\hat{\boldsymbol{o}}_t)} \frac{\partial \log b_i(\hat{\boldsymbol{o}}_t)}{\partial \mathbf{W}} \\ &= \sum_{t=1}^{T}\sum_{i=1}^{M} \gamma_t(i) \frac{\partial \log b_i(\hat{\boldsymbol{o}}_t)}{\partial \mathbf{W}}\end{aligned} \tag{4}$$

which is the sum of gradients from each HMM state weighted by their posterior probabilities.

Assume the observation likelihood of the transformed data in each HMM state i have the following form

$$b_i(\hat{\boldsymbol{o}}_t) = \frac{1}{Z_i} \exp\left\{-E_i(\boldsymbol{o}_t)\right\} \tag{5}$$

where

$$E_i(\boldsymbol{o}_t) = \frac{1}{2}[f_{\mathbf{W}}(\boldsymbol{o}_t) - \boldsymbol{\mu}]^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} [f_{\mathbf{W}}(\boldsymbol{o}_t) - \boldsymbol{\mu}] \tag{6}$$

$$Z_i = \int_{\boldsymbol{o}_t} \exp\left\{-E_i(\boldsymbol{o}_t)\right\} d\boldsymbol{o}_t \tag{7}$$

are the energy function and partition function for state i.

From Eq.5 one has

$$\log b_i(\hat{\boldsymbol{o}}_t) = -E_i(\boldsymbol{o}_t) - \log Z_i \tag{8}$$

and its gradient can be computed as

$$\begin{aligned}\frac{\partial \log b_i(\hat{\boldsymbol{o}}_t)}{\partial \mathbf{W}} &= -\frac{\partial E_i(\boldsymbol{o}_t)}{\partial \mathbf{W}} - \frac{1}{Z_i}\frac{\partial Z_i}{\partial \mathbf{W}} \\ &= -\frac{\partial E_i(\boldsymbol{o}_t)}{\partial \mathbf{W}} + \int_{\boldsymbol{o}_t} b_i(\hat{\boldsymbol{o}}_t) \frac{\partial E_i(\boldsymbol{o}_t)}{\partial \mathbf{W}} d\boldsymbol{o}_t\end{aligned} \tag{9}$$

where

$$\frac{\partial E_i(\boldsymbol{o}_t)}{\partial \mathbf{W}} = \frac{\partial f_{\mathbf{W}}(\boldsymbol{o}_t)}{\partial \mathbf{W}} \frac{\partial E_i(\boldsymbol{o}_t)}{\partial f_{\mathbf{W}}(\boldsymbol{o}_t)} = \frac{\partial f_{\mathbf{W}}(\boldsymbol{o}_t)}{\partial \mathbf{W}} \boldsymbol{\Sigma}^{-1} [f_{\mathbf{W}}(\boldsymbol{o}_t) - \boldsymbol{\mu}] \tag{10}$$

Substituting Eq.10 back to Eq.9, one has

$$\begin{aligned}\frac{\partial \log b_i(\hat{\boldsymbol{o}}_t)}{\partial \mathbf{W}} &= -\frac{\partial f_{\mathbf{W}}(\boldsymbol{o}_t)}{\partial \mathbf{W}} \boldsymbol{\Sigma}^{-1} [f_{\mathbf{W}}(\boldsymbol{o}_t) - \boldsymbol{\mu}] \\ &+ \int_{\boldsymbol{o}_t} b_i(\hat{\boldsymbol{o}}_t) \left\{ \frac{\partial f_{\mathbf{W}}(\boldsymbol{o}_t)}{\partial \mathbf{W}} \boldsymbol{\Sigma}^{-1} [f_{\mathbf{W}}(\boldsymbol{o}_t) - \boldsymbol{\mu}] \right\} d\boldsymbol{o}_t\end{aligned} \tag{11}$$

Since $f_{\mathbf{W}}$ is a nonlinear transformation based on a neural network, the derivative term $\frac{\partial f_{\mathbf{W}}(\boldsymbol{o}_t)}{\partial \mathbf{W}}$ can be evaluated recursively using the back-propagation algorithm. The second term on the RHS does not have an analytical solution but it can be evaluated using the Monte Carlo methods. However, directly sampling from distribution $b_i(\hat{\boldsymbol{o}}_t)$ is not trivial as $\hat{\boldsymbol{o}}_t$ is after a nonlinear transformation and, as a result, $b_i(\hat{\boldsymbol{o}}_t)$ is not Gaussian any more. In this work, *importance sampling* is used for evaluating this expectation term using a proposal distribution that is easier to deal with.

2.2. Importance Sampling

Suppose one wants to evaluate an expectation of a function $f(x)$ with respect to some distribution $p(x)$. The importance sampling method draws samples from a proposal distribution $q(x)$. Let's express the distributions explicitly with the normalization term

$$p(x) = \frac{1}{Z_p}\tilde{p}(x), \quad q(x) = \frac{1}{Z_q}\tilde{q}(x) \tag{12}$$

Assume $\tilde{p}(x)$ can be evaluated easily but Z_p can not. Choose a proposal distribution $q(x)$ such that both $\tilde{q}(x)$ and Z_q can be evaluated easily. Draw samples $x^{(k)}$ $(k = 1, \cdots, K)$ from $q(x)$ and one has

$$\begin{aligned}E[f] &= \int_x f(x)p(x)dx = \frac{Z_q}{Z_p}\int_x f(x)\left[\frac{\tilde{p}(x)}{\tilde{q}(x)}\right] q(x)dx \\ &\approx \frac{Z_q}{Z_p}\frac{1}{K}\sum_{k=1}^{K}\tilde{r}_k f(x^{(k)})\end{aligned} \tag{13}$$

where $\tilde{r}_k = \tilde{p}(x^{(k)})/\tilde{q}(x^{(k)})$ and the ratio of two normalization terms can be evaluated using the same samples

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int_x \tilde{p}(x)dx = \int_x \frac{\tilde{p}(x)}{\tilde{q}(x)}q(x)dx = \frac{1}{K}\sum_{k=1}^{K}\tilde{r}_k \tag{14}$$

It follows that

$$E[f] \approx \frac{Z_q}{Z_p}\frac{1}{K}\sum_{k=1}^{K}\tilde{r}_k f(x^{(k)}) = \sum_{k=1}^{K} w_k f(x^{(k)}) \tag{15}$$

where

$$w_k = \frac{\tilde{r}_k}{\sum_{m=1}^{K}\tilde{r}_m} = \frac{\tilde{p}(x^{(k)})/\tilde{q}(x^{(k)})}{\sum_{m=1}^{K}\tilde{p}(x^{(m)})/\tilde{q}(x^{(m)})} \tag{16}$$

Eq.15 shows that with importance sampling, the expectation is approximated by a weighted average using the samples drawn from the proposal distribution, where the weights computed by Eq.16 are used to compensate the bias between the proposal and target distributions.

Importance sampling also provides a way to evaluate the partition function of a distribution. According to Eq.14

$$\frac{Z_p}{Z_q} = \frac{1}{K}\sum_{k=1}^{K}\tilde{r}_k \tag{17}$$

It follows that

$$\log Z_p = \log\left(\frac{1}{K}\sum_{k=1}^{K}\tilde{r}_k\right) + \log Z_q \tag{18}$$

If the partition function Z_q of the proposal distribution $q(x)$ can be evaluated analytically and the partition function Z_p of the target distribution $p(x)$ can be evaluated according to Eq.18.

2.3. Network Pre-training

For the DNN-based nonlinear transformation illustrated in Fig.1, the network is first pre-trained towards a linear transformation. Given the acoustic model λ and the feature sequence $\boldsymbol{\mathcal{O}}$, the CMLLR transformation $\{\boldsymbol{A}, \boldsymbol{b}\}$ is estimated to maximize the following likelihood function [2]

$$\log P(\hat{\boldsymbol{\mathcal{O}}}|\boldsymbol{A}^{-1}(\boldsymbol{\mu}-\boldsymbol{b}), \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\boldsymbol{A}^{-T}, \boldsymbol{\mathcal{O}}) \tag{19}$$

which is accomplished by the EM algorithm [11]. Once the CMLLR transformation is in place, set the target vector y in Fig.1 to

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \tag{20}$$

where $\boldsymbol{x}$ is the input feature vector. Given input and the CMLLR-transformed target, the weights $\mathbf{W}$ of the network are initialized randomly and then optimized by the GD algorithm under the MMSE criterion

$$\min_{\mathbf{W}} ||\boldsymbol{y} - \boldsymbol{x}||^2 \tag{21}$$

Therefore, after the pre-training, the weights $\mathbf{W}$ of the network are brought to the vicinity of the CMLLR transformation before the fine-tuning under the sequence-based ML criterion introduced in Section 2.1 using the GD algorithm.

The success of importance sampling requires the proposal distribution overlaps well with the target distribution. Therefore, if the nonlinear mapping $f_{\mathbf{W}}$ renders a heavy mismatch between target distribution $b_i(\hat{\boldsymbol{o}}_t)$ and the proposal distribution then the importance sampling will not work well. A natural option of the proposal distribution would be the original observation distribution $b_i(\boldsymbol{o}_t)$ of the acoustic model. However, this will only work well if $f_{\mathbf{W}}$ is not far away from identity mapping. When $f_{\mathbf{W}}$ becomes highly nonlinear, the match between the two distributions will be poor.

In this work, the proposal distribution is chosen to be the CMLLR-transformed original state observation distribution. In the MMSE pre-training, CMLLR is first estimated and the weights of the network $\mathbf{W}$ are initialized close to the CMLLR linear transformation. Therefore, a good match between the two distributions are expected. Furthermore, since CMLLR is a linear transformation, the proposal distribution is still Gaussian from which samples can be easily drawn. Under this condition, the proposal distribution can be expressed as

$$q(\boldsymbol{o}_t) = \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{A}^{-1}(\boldsymbol{\mu}-\boldsymbol{b}), \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\boldsymbol{A}^{-T}) \tag{22}$$

and its partition function is $\log Z_q = \log \bar{Z}_q - \log|\boldsymbol{A}|$ where $\bar{Z}_q$ is the partition function of the original state observation Gaussian distribution. Along with Eq.18, the partition function of the target distribution can be obtained.

3. A STACKED ARCHITECTURE

The nonlinear transformation created by the neural network in Fig.1 can serve as a building block in a stacked architecture as shown in Fig.2. This deep architecture represents a nonlinear transformation itself but it is constructed by a composite of a series of nonlinear transformations described by sub-networks in Fig.1. The sub-networks are called "blocks" in Fig.2 and each block has nonlinear activation functions in its hidden layers and identity activation functions in the output layer. With the hierarchical nonlinear transformations contributed by each sub-network, this deep stacked architecture may have a better representation capability overall for nonlinearity. The weights of this stack architecture are denoted by

$$\mathbf{W}^{(N)} = (\mathbf{W}_1, \cdots, \mathbf{W}_n, \cdots, \mathbf{W}_N) \tag{23}$$

where $\mathbf{W}_n$ represents the weights of the nth block.

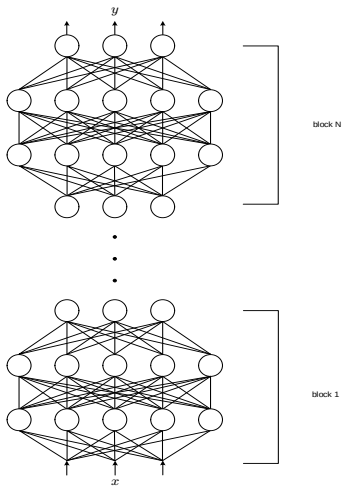

Fig. 2. A stacked deep architecture consisting of hierarchical sub-networks to represent a nonlinear transformation, which is a composition of a series of nonlinear transformations described by each sub-network (block).

The deep architecture in Fig.2 can be learned block-wise. The training of the first block follows what is described in Section 2. That is, one first estimate CMLLR using the EM algorithm based on the HMM acoustic model in the conventional way [2] and pre-train the weights of the block using MMSE with the CMLLR-transformed target to approximate the CMLLR linear transformation. After the pre-training, fine-tune the weights of the block using the GD algorithm based on the sequence-based ML criterion.

Suppose the lower $n-1$ blocks have been successfully trained. Take the output of the low $n-1$ blocks

$$\boldsymbol{\mathcal{O}}^{(n-1)} = f_{\mathbf{W}^{(n-1)}}(\boldsymbol{\mathcal{O}}) \tag{24}$$

and estimate the CMLLR transformation of $\boldsymbol{\mathcal{O}}^{(n-1)}$ with respect to the acoustic model λ

$$\log P(\hat{\boldsymbol{\mathcal{O}}}|\boldsymbol{A}^{-1}(\boldsymbol{\mu}-\boldsymbol{b}), \boldsymbol{A}^{-1}\boldsymbol{\Sigma}\boldsymbol{A}^{-T}, \boldsymbol{\mathcal{O}}^{(n-1)}). \tag{25}$$

Since the $f_{\mathbf{W}^{(n-1)}}$ is nonlinear, the CMLLR estimated for the nth block will not be subsumed. Once the CMLLR is estimated, fix the weights $\mathbf{W}^{(n-1)}$ of the lower $n-1$ blocks and run MMSE pre-training of the nth block with CMLLR-transformed target to initialize the weights of the nth block. After that, sequence-based ML fine-tuning is performed through all the weights of the n blocks including the lower $n-1$ blocks.

The CMLLR estimate is the local optimum in the sense of linear transformation after the maximization step of the EM algorithm. The initialization of weights of the network close to the CMLLR linear transformation by the MMSE pre-training gives the GD search a reasonable starting point for the nonlinear transformation estimate. In addition, in the block-wise pre-training of the stacked architecture, the CMLLR estimate of each additional block is expected to bring

the estimate of the previous nonlinear transformation away from its local optimum and provide a better starting point to start the GD search for a new local optimum.

4. EXPERIMENTAL RESULTS

Speaker adaptation experiments are conducted on two English LVCSR tasks from the DARPA Transtac program. The first task has 11 speakers in the test set and each speaker has about 3-4 minutes of speech data recorded in quiet environment. The second task has 7 speakers in the test set and each speaker has about 12-15 minutes of speech data recorded in noisy environment. Both tasks are spontaneous speech sampled at 16KHz. In the proposed MLNT, the dimensionality of the input and output layers of each block in the stacked architecture is 40 which is equal to the dimensionality of the input feature space. One hidden layer with 100 hidden units is used in each block. We found that in this deep stacked architecture using more than one hidden layer only gives marginal overall improvements. Hyperbolic tangent activation functions are used in hidden units and identity functions are used in the output units of each block. The MMSE pre-training runs 100 iterations using a step size of 1e-4 while the sequence-based ML fine-tuning runs 5 iterations using a step size of 5e-8. When evaluating the likelihood partition function, 100 samples are drawn from the proposal distribution in the importance sampling method. Note that the partition functions also have to be evaluated in decoding.

Table 1 presents the results on the first test set with clean speech. The baseline acoustic model is a discriminative model trained under the boosted maximum mutual information (BMMI) criterion on feature-space MMI (FMMI) discriminative features [12] using 60 hours of clean speech. The model has 3K quinphone states and 50K Gaussians. The adaptation is carried out for each speaker. CMLLR yields 2.3% absolute improvement over the baseline. The proposed MLNT uses 5 blocks, whose WER is 23.6%, 1.4% absolute better than CMLLR.

model	WER
FMMI+BMMI baseline	**27.3**
FMMI+BMMI+CMLLR	**25.0**
MLNT block 1	24.7
MLNT block 2	24.2
MLNT block 3	24.0
MLNT block 4	23.7
MLNT block 5	**23.6**

Table 1. Word error rates (WERs) of baseline, CMLLR and the proposed MLNT on the test set of clean speech.

Table 2 presents the results on the second test set with noisy speech. The baseline acoustic model is an ML model with multi-style training (MST) using LDA features. The MST training data is 60 hours including both clean and noisy speech. The signal-to-noise ratios (SNRs) of noisy training data are between 10dB and 25dB. The estimated SNRs of the noisy test data are between 5dB and 8dB. The model has 2K quinphone states and 80K Gaussians. In this case, CMLLR yields a significant improvement over the baseline by 17.4% absolute. This is mainly due to the mismatch between the MST acoustic model and the noisy speech from the test speakers. The proposed MLNT uses 3 blocks, whose WER is 27.2%, 3.3% absolute better than CMLLR.

From both tables, it can be seen that with the additional blocks the performance of the nonlinear transformation represented by the stacked architecture is gradually improved from the first block.

model	WER
LDA+ML baseline	**47.9**
LDA+ML+CMLLR	**30.5**
MLNT block 1	29.2
MLNT block 2	27.3
MLNT block 3	**27.2**

Table 2. Word error rates (WERs) of baseline, CMLLR and the proposed MLNT on the test set of noisy speech.

5. DISCUSSION AND FUTURE WORK

Speaker adaptation using nonlinear transformations such as piecewise linear CMLLR [13] has been previously investigated. The MLNT investigated in this paper uses DNNs for nonlinear feature transformations for HMM acoustic models, which can be considered an extension of CMLLR. Despite of a linear transformation, CMLLR is very effective at reducing statistical mismatch and has been used as an off-the-shelf technique. MLNT is first pre-trained by minimizing the MMSE between the input and CMLLR-transformed target so that the weights of the network are initialized (approximately) to the linear transformation of CMLLR. Then the weights are fine-tuned using the GD algorithm under the sequence-based ML criterion. In [14], ANNs are used for nonlinear feature space transformations. It appears that the impact of the partition function term to the likelihood function due to the nonlinearity is ignored. A fast nonlinear feature transformation method based on DNNs is studied in [15] for speaker adaptation, whose framework is similar to this paper. However, only the nonlinearity of the bias term is considered in [15]. This paper also proposes a novel stacked architecture that is hierarchically constructed using sub-networks as building blocks. Each sub-network itself is a DNN modeling a nonlinear transformation and the overall deep stacked architecture represents a composite of multiple nonlinear transformations, which is more powerful and yields better performance. The stacked architecture can be learned block-wise including again a pre-training step followed by a fine-tuning step.

Speaker adaptation in hybrid ANN/HMM systems are usually carried out by retraining an additional input or output layer [16][17]. The work discussed here is focused on a nonlinear transformation for generative GMM-HMM acoustic models.

For the future work, we would like to improve the evaluation of the likelihood partition function to make the MC sampling more efficient and less computationally demanding. We also would like to investigate possible ways to linearize the nonlinear transformations provided by the DNNs.

6. ACKNOWLEDGEMENTS

This work is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

7. REFERENCES

[1] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[2] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[3] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," in *IEEE Signal Processing Maganize*, November 2012, pp. 82–97.

[5] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 7398–7402.

[6] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 2, no. 1, pp. 65–68, 2014.

[7] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859 – 1872, 2014.

[8] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*, London: Methuen, 1975.

[9] G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, New York: Springer, 1995.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[12] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 4057–4060.

[13] S. S. Kozat, K. Visweswariah, and R. Gopinath, "Feature adaptation based on Gaussian posteriors," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006, pp. 221–224.

[14] V. Valtchev, "Discriminative methods in HMM-based speech recognition," *Ph.D Thesis, University of Cambridge*, 1995.

[15] K. Yao, D. Yu, L. Deng, and Y. Gong, "A fast maximum likelihood nonlinear feature transformation method for GMM-HMM speaker adaptation," *Neurocomputing*, vol. 128, pp. 145–152, 2014.

[16] A. Sankar V. Abrash, H. Franco and M. Cohen, "Connectionist speaker normalization and adaptation," in *Eurospeech*, 1995, pp. 2183–2186.

[17] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *Interspeech*, 2010, pp. 526–529.