INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Deep Activation Mixture Model for Speech Recognition

Chunyang Wu & Mark J.F. Gales

Cambridge University Engineering Dept,
Trumpington St., Cambridge, CB2 1PZ, U.K.
{cw564, mjfg}@eng.cam.ac.uk

Abstract

Deep learning approaches achieve state-of-the-art performance in a range of applications, including speech recognition. However, the parameters of the deep neural network (DNN) are hard to interpret, which makes regularisation and adaptation to speaker or acoustic conditions challenging. This paper proposes the deep activation mixture model (DAMM) to address these problems. The output of one hidden layer is modelled as the sum of a mixture and residual models. The mixture model forms an activation function contour while the residual one models fluctuations around the contour. The use of the mixture model gives two advantages: First, it introduces a novel regularisation on the DNN. Second, it allows novel adaptation schemes. The proposed approach is evaluated on a large-vocabulary U.S. English broadcast news task. It yields a slightly better performance than the DNN baselines, and on the utterance-level unsupervised adaptation, the adapted DAMM acquires further performance gains.

Index Terms: deep learning, mixture model, speaker adaptation

1. Introduction

Progress in deep learning [1, 2, 3] has improved the performance of state-of-the-art speech recognition systems. The multi-layer hidden transformations and activations in a deep neural network (DNN) and related network variations allow complex and difficult data to be well modelled. However, this highly-distributed representation means that it is hard to interpret the model parameters. This causes challenges to adaptation, and more general regularisation.

To reduce over-fitting, regularisation techniques are commonly used in the DNN training. The weight decay method adds an L2-norm of DNN weights to the training criterion, which encourages smaller weights in optimisation. The dropout approach [4] temporally turns off a random set of activations during the training procedure; as a result, the overall DNN turns out to be a boosted model consisting of many sub-DNNs, which prevents over-fitting to the training data. However, conventional regularisations can hardly improve the interpretability of network parameters. In [5, 6], the concept of stimulated learning regularises the neurones of different regions to correspond to different phonemes, which aids interpretation and visualisation of the DNN.

In addition to modifying the training process, structured neural networks have also been investigated. The DNN topology is explicitly modified: different types of parameters or neurones are restricted to model specific functions. The mixture density network [7, 8, 9, 10, 11] parametrises the mixture components via DNNs, which in turn yields a "deep" probability density function. Similarly, [12] stacks Gaussian mixture layers for density estimation. The multi-task neural networks [13, 14, 15] extend the output layer with auxiliary tasks to better regularise the primary one. Another category of structured DNNs focuses on improving the capability of adaptation: interpretable modules are imposed on the network structure, exposing meaningful parameters to adapt the speaker-independent (SI) model. In [16, 17, 18], additional linear layers are introduced as speaker-dependent (SD) transforms. The speaker code model [19] introduces SD descriptors as features to bottom DNN layers. The learning hidden unit contributions [20] and the parametric activation [21] schemes introduce an SD scaling factor on each hidden-layer activation. The multi-basis adaptive neural network [22, 23] combines multiple parallel subnetworks to handle acoustic distortions, and in [24, 25], hidden-layer transformations are adapted via weight interpolation. The differentiable pooling method [26] introduces hidden-activation candidate pools to obtain the SD compensation.

This paper proposes a novel structured deep neural network, referred to as the deep activation mixture model (DAMM). Inspired by the stimulated DNNs [5, 6], the DAMM encourages activations in regions of network to be related. However, rather than being implemented as a regularisation term during training, the hidden activations are explicitly modelled as the sum of a mixture and residual models. The mixture model expands an activation contour that roughly describes the behaviours of activations, while on the other hand, the residual model adds variations to the contour for all hidden activations. Consequently, the result activations stay on a contour controlled by the mixture model, which triggers nearby neurones over the contour to be similar. In contrast to mixture density networks [7], this approach utilises the contour of a mixture-model distribution, instead of estimating a density function in a "deep" configuration. The DAMM activations are related and controlled by the mixture model, which has the potential to improve network regularisation. Meanwhile, this highly-restricted mixture model can be robustly re-estimated. It allows novel approaches to DNN rapid adaptation, when there is insufficient adaptation data. The proposed model is evaluated on a large-vocabulary U.S. broadcast news transcription task.

The rest of this paper is organised as follows. The deep activation mixture model is proposed in Section 2. Experimental results are reported in Section 3 and finally, this paper is concluded in Section 4.

The research leading to these results was 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; research projects and donations from Google and Amazon. The U.S. Government is authorised 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.

http://dx.doi.org/10.21437/Interspeech.2017-1233

2. Deep Activation Mixture Model

The deep activation mixture model can be viewed as an extension to a standard multi-layer perceptron (MLP). Rather than treating hidden activations independently, the DAMM defines the activations as the combination of two models: the mixture model forms a smooth activation contour that the neurones should roughly echo; the residual model aids the variations around the contour.

The topology of the deep activation mixture model is illustrated in Figure 1. Similar to an MLP, it consists of an input layer to load the feature vector $\boldsymbol{x}$

$$\boldsymbol{h}^{(0)} = \boldsymbol{x}, \tag{1}$$

several hidden layers and a output layer. The output layer is

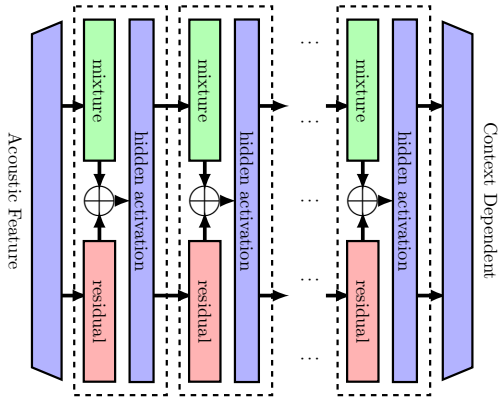


Figure 1: *Deep Activation Mixture Model.*

defined as a softmax layer to predict the posterior of the context-dependent target y

$$P(y=i|\boldsymbol{x}) = \frac{\exp\left(\boldsymbol{w}_i^{(L)T}\boldsymbol{h}^{(L-1)} + c_i^{(L)}\right)}{\sum_j \exp\left(\boldsymbol{w}_j^{(L)T}\boldsymbol{h}^{(L-1)} + c_j^{(L)}\right)}, \tag{2}$$

where L is the total number of layers; $\boldsymbol{W}^{(L)}$ and $\boldsymbol{c}^{(L)}$ are the output layer transformation.

Inspired by the stimulated learning [5], the DAMM aims at encouraging activations in regions of the network to be related. However, this kind of region information is explicitly defined via a mixture model in the DAMM, instead of an inexplicit regularisation in the stimulated learning. The activations of one hidden layer are firstly rearranged to a grid, *e.g.*, a layer with 1024 neurones can form a 32×32 two-dimensional grid. This grid is fitted to the unit square $[0,1]^2$ of a two-dimensional *network-grid space*. Therefore the i-th activation can be represented as a point in this space, denoted as $\boldsymbol{s}_i$. This network grid specifies a spatial ordering to the activations in one layer. It is then possible to define and train the smooth activation contours based on this spatial ordering. On the l-th hidden layer, the output of the activations $\boldsymbol{h}^{(l)}$ is defined as the sum of a mixture model $\boldsymbol{h}_{mix}^{(l)}$ and a residual model $\boldsymbol{h}_{res}^{(l)}$

$$\boldsymbol{h}^{(l)} = \boldsymbol{h}_{mix}^{(l)} + \boldsymbol{h}_{res}^{(l)}. \tag{3}$$

The position-dependent prior is explicitly governed by the mixture model $\boldsymbol{h}_{mix}^{(l)}$. It describes the rough behaviours of activations via the contour of the probability density function of a Gaussian mixture distribution (Figure 2(a))

$$h_{mix,i}^{(l)} = g^{(l)} \sum_{k=1}^{K} \omega_k^{(l)} \mathcal{N}\left(\boldsymbol{s}_i; \boldsymbol{\mu}_k^{(l)}, \boldsymbol{\Sigma}_k^{(l)}\right), \tag{4}$$

where K stands for the total of Gaussian components. $g^{(l)}$ is a scaling factor to assign the importance of the mixture model

$$g^{(l)} = f\left(\boldsymbol{q}^{(l)T}\boldsymbol{h}^{(l-1)} + r^{(l)}\right), \tag{5}$$

where $f(\cdot)$ is the sigmoid function; $\boldsymbol{q}^{(l)}$ and $r^{(l)}$ are parameters applied to $\boldsymbol{h}^{(l-1)}$, the hidden output of the previous layer; the mixing weights $\boldsymbol{\omega}^{(l)}$ of the Gaussian mixtures are given by

$$\omega_k^{(l)} = \frac{\exp\left(\boldsymbol{a}_k^{(l)T}\boldsymbol{h}^{(l-1)} + b_k^{(l)}\right)}{\sum_{\tilde{k}} \exp\left(\boldsymbol{a}_{\tilde{k}}^{(l)T}\boldsymbol{h}^{(l-1)} + b_{\tilde{k}}^{(l)}\right)}, \tag{6}$$

defined by a softmax function with $\boldsymbol{A}^{(l)}$ and $\boldsymbol{b}^{(l)}$ as parameters.

The residual model $\boldsymbol{h}_{res}^{(l)}$ is given as the hyperbolic tangent activations

$$\boldsymbol{h}_{res}^{(l)} = \tanh\left(\boldsymbol{W}^{(l)T}\boldsymbol{h}^{(l-1)} + \boldsymbol{c}^{(l)}\right), \tag{7}$$

associated with parameters $\boldsymbol{W}^{(l)}$ and $\boldsymbol{c}^{(l)}$. It is introduced to represent variations on different activations in the grid, enriching the expressiveness of every single activation (Figure 2(b)).

Overall, the mixture model $\boldsymbol{h}_{mix}^{(l)}$ forms a smooth activation contour; the residual term $\boldsymbol{h}_{res}^{(l)}$ models fluctuations around the contour. The number of Gaussian components K is smaller than that of neurones in one hidden layer. Therefore, the mixture model is highly restricted due to fewer parameters associated with it.

2.1. Training

In the DAMM, two categories of parameters are optimised: the mixture model $\mathcal{M}_{mix}$ and the residual model $\mathcal{M}_{res}$

$$\mathcal{M}_{mix} = \left\{\boldsymbol{q}^{(l)}, r^{(l)}, \boldsymbol{A}^{(l)}, \boldsymbol{b}^{(l)}\right\}_{1\le l<L}, \tag{8}$$

$$\mathcal{M}_{res} = \left\{\boldsymbol{W}^{(l)}, \boldsymbol{c}^{(l)}\right\}_{1\le l\le L}. \tag{9}$$

Notice that all $\boldsymbol{\mu}_k^{(l)}$ and $\boldsymbol{\Sigma}_k^{(l)}$ are fixed during the training phase of DAMM, for encouraging the mixture model to generate a meaningful activation contour. However in Section 2.2, these parameters are used as the speaker-dependent transform to adapt a well-trained DAMM.

Define $\boldsymbol{\theta} = \mathcal{M}_{mix} \cup \mathcal{M}_{res}$ as the parameters in a deep activation mixture model. The optimisation can follow the same fashion as the back-propagation algorithm, which tries to minimise some criterion $\mathcal{F}(\cdot)$ over a training set of T samples $\{(\boldsymbol{x}_t, y_t); 1 \le t \le T\}$. In this paper, the training criterion is

$$\mathcal{F}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \eta\mathcal{R}(\boldsymbol{\theta}) \tag{10}$$

where $\mathcal{L}(\boldsymbol{\theta})$ is a standard criterion, *e.g.*, the cross-entropy criterion

$$\mathcal{L}_{ce}(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T} \log P(y_t|\mathbf{x}_t;\boldsymbol{\theta}); \tag{11}$$

$\mathcal{R}(\boldsymbol{\theta})$ is the standard L2 regularisation term. In this work, it is only applied to the residual-model parameters $\mathcal{M}_{res}$

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{l}\sum_{i}\left(|c_i^{(l)}|^2 + \sum_{j}|w_{ij}^{(l)}|^2\right). \quad (12)$$

The hyper-parameter η stands for the contribution of the regularisation. The proposed DAMM expects to minimise $||\boldsymbol{h}^{(l)} - \boldsymbol{h}_{mix}^{(l)}||_2^2$ in order to regularise the activations on the contour of the mixture model. Eq. 12 penalises the residual weights to be close to zero, thus keeping $\boldsymbol{h}_{res}^{(l)}$ to model tiny variations.

For the residual-model parameters, the gradients $\frac{\partial\mathcal{F}}{\partial\boldsymbol{W}^{(l)}}$ and $\frac{\partial\mathcal{F}}{\partial\boldsymbol{c}^{(l)}}$ can be recursively calculated with

$$\frac{\partial\mathcal{F}}{\partial\boldsymbol{h}_{res}^{(l)}} = \frac{\partial\mathcal{F}}{\partial\boldsymbol{h}^{(l)}}. \quad (13)$$

For the parameters in the mixture model, $\frac{\partial\mathcal{F}}{\partial\boldsymbol{q}^{(l)}}$ and $\frac{\partial\mathcal{F}}{r^{(l)}}$ can be calculated with

$$\frac{\partial\mathcal{F}}{\partial g^{(l)}} = \sum_{k}\sum_{i}\omega_k^{(l)}\mathcal{N}\left(\boldsymbol{s}_i;\boldsymbol{\mu}_k^{(l)},\boldsymbol{\Sigma}_k^{(l)}\right)\frac{\partial\mathcal{F}}{\partial h_i^{(l)}} \quad (14)$$

and $\frac{\partial\mathcal{F}}{\partial\boldsymbol{A}^{(l)}}$ and $\frac{\partial\mathcal{F}}{\boldsymbol{b}^{(l)}}$ can be calculated with

$$\frac{\partial\mathcal{F}}{\partial\omega_k^{(l)}} = g^{(l)}\sum_{i}\mathcal{N}\left(\boldsymbol{s}_i;\boldsymbol{\mu}_k^{(l)},\boldsymbol{\Sigma}_k^{(l)}\right)\frac{\partial\mathcal{F}}{\partial h_i^{(l)}}. \quad (15)$$

The training scheme is listed in Algorithm 1. The DAMM is optimised in a layer-wise pre-training mode and subsequently fine-tuned. During the pre-training phase (Line 1–5), the l-th iteration first initialises and adds the l-th layer mixture and residual models parameters, denoted by $\mathcal{M}_{mix}^{(l)}$ and $\mathcal{M}_{res}^{(l)}$. The mixture model is randomly initialised while the residual model is initialised as 0 which is expected in the L2 regularisation of the residual model. Unlike the powerful residual model, the

Algorithm 1 Layer-wise Training Mode of DAMM.

1: **for** $l := 1$ **to** L **do**
2: **initialise** $\mathcal{M}_{res}^{(l)} = \mathbf{0}, \mathcal{M}_{mix}^{(l)}$
3: **update** $\mathcal{M}_{mix}$
4: **update** $\mathcal{M}_{res}$
5: **end for**
6: **finetune** $\mathcal{M}_{res}$

mixture model is highly restricted. A joint optimisation on both would degrade the mixture one to learn nothing since the powerful residual model is likely to absorb its functions. Therefore, they are separately optimised: the update of mixture model is performed till convergence while the residual one is tuned for fewer epochs. In this way, the mixture model is turned to its maximal extent before introducing the residual one. After the pre-training, the residual model is then fully optimised (Line 6).

2.2. Adaptation

A significant advantage of the DAMM is that the mixture model can be robustly adapted to boost the hidden activations, instead of modifying the DNN neurones independently. The adapted mixture model can be expressed as

$$h_{mix,i}^{(ls)} = g^{(l)}\sum_{k=1}^{K}\omega_k^{(l)}\mathcal{N}\left(\boldsymbol{s}_i;\boldsymbol{\mu}_k^{(ls)},\boldsymbol{\Sigma}_k^{(ls)}\right) \quad (16)$$

where s stands for the speaker index. The estimation of $\boldsymbol{\mu}_k^{(ls)}$ and $\boldsymbol{\Sigma}_k^{(ls)}$ would change the activation contour generated by the mixture model, thus contributing to the DAMM adaptation.

The mean vector and the covariance matrix of any Gaussian component are parametrised as follows. $\boldsymbol{\mu}_k^{(l)}$ is restricted in the valid range of the grid $[0,1]^2$. For the covariance matrix, in the two-dimensional case of this paper, $\boldsymbol{\Sigma}_k^{(l)}$ of a bivariate Gaussian can be factorised as

$$\boldsymbol{\Sigma}_k^{(l)} = \begin{vmatrix} \sigma_{k1}^{(l)2} & \rho_k^{(l)}\sigma_{k1}^{(l)}\sigma_{k2}^{(l)} \\ \rho_k^{(l)}\sigma_{k1}^{(l)}\sigma_{k2}^{(l)} & \sigma_{k2}^{(l)2} \end{vmatrix}, \quad (17)$$

where $\boldsymbol{\sigma}_k^{(l)}$ is the unit variance vector which should positive and $\rho_k^{(l)}$ is the correlation coefficient which should lay in the range $[-1,1]$. Thus they are parametrised as

$$\boldsymbol{\sigma}_k^{(l)} = \exp\left(\tilde{\boldsymbol{\sigma}}_k^{(l)}\right), \quad \rho_k^{(l)} = \tanh\left(\tilde{\rho}_k^{(l)}\right), \quad (18)$$

in order to comply with their mathematical constraints. By introducing $\boldsymbol{\sigma}_k^{(l)}$ and $\rho_k^{(l)}$, the positive-definite property of $\boldsymbol{\Sigma}_k^{(l)}$ can inherently be satisfied, requiring no additional constraints during optimisation.

Define a vector $\boldsymbol{\phi}_k^{(ls)}$ consisting of the adaptable parameters of the k-th Gaussian component in the l-th layer

$$\boldsymbol{\phi}_k^{(ls)} = \left[\mu_{k1}^{(ls)}, \mu_{k2}^{(ls)}, \sigma_{k1}^{(ls)}, \sigma_{k2}^{(ls)}, \rho_k^{(ls)}\right]^T \quad (19)$$

and $\boldsymbol{\phi}^{(s)}$ as a super-vector concatenating $\boldsymbol{\phi}_k^{(ls)}$ of Gaussian components in all hidden layers. The adaptation criterion $\mathcal{F}(\boldsymbol{\phi}^{(s)})$ is the cross-entropy criterion over the adaptation data

$$\mathcal{F}(\boldsymbol{\phi}^{(s)}) = \mathcal{L}_{ce}(\boldsymbol{\phi}^{(s)}). \quad (20)$$

The Gaussian components can be updated via the stochastic gradient descent scheme. The gradient of $\boldsymbol{\phi}_k^{(ls)}$ is given as

$$\frac{\partial\mathcal{F}}{\partial\boldsymbol{\phi}_k^{(ls)}} = g^{(l)}\sum_{i}\omega_k^{(l)}\frac{\partial\mathcal{N}}{\partial\boldsymbol{\phi}_k^{(ls)}}\frac{\partial\mathcal{F}}{\partial h_i^{(ls)}}. \quad (21)$$

3. Experiments

The experiments were conducted on a large-vocabulary U.S. English broadcast news (BN) transcription task. The training data consisted of the 144-hour 1996 & 1997 Hub-4 English Broadcast Speech dataset (`LDC97S44`, `LDC98S71`). 288 shows were included, from approximately 8k speakers. In evaluation, both the BN 2.7-hour `Dev03` and 2.6-hour `Eval03` testsets were used. The utterances of both testsets were processed by automatic segmentation and the averaged utterance durations were respectively 10.7 and 10.9 seconds. Decoding was performed with the RT04 tri-gram language model [27]. The adaptation of the DAMM was evaluated in the utterance-level unsupervised fashion: the hypothesis alignments of the SI DAMM are used to tune the adaptive parameters.

3.1. Setup

The GMMs, DNNs and the proposed models were trained on an extended version of HTK Toolkit 3.5 [28]. The 39-dimensional PLP+Δ+$\Delta\Delta$ features were processed by both corpus-level cepstral mean normalisation (CMN) and cepstral variance normalisation (CVN) were used to train a GMM-HMM model, containing approximately 6k tied triphone states on the maximum

likelihood (ML) estimation. The features were subsequently extended with the triples using HLDA to estimate a sequential minimum-phone-error (MPE) model. This MPE model then was used to generate the state alignments of the training set for training DNN systems.

For the DNN baselines, the 468 DNN input feature was formed by the PLP+Δ+$\Delta\Delta$+$\Delta\Delta\Delta$ in a context window of 9 frames. The DNN introduced 5 hidden layers with 1024 sigmoid nodes in each layer. The network was initialised in a layer-wise pre-training mode and then optimised via back-propagation in the cross-entropy criterion. 28 shows with around 600 speakers in the raw training data were randomly selected as the cross validation set. The L2-regularisation penalty was set to be 10^{-6}. This well-trained CE DNN system was then used to generate the lattices of the training set and further tuned for three iterations under the MPE criterion to obtain the MPE DNN system.

For the deep activation mixture models, 46 Gaussian components were introduced to the mixture model in each hidden layer, equal to the number of phonemes: Each component represented a phoneme and $\boldsymbol{\mu}_k^{(l)}$ was given by the 2D projection via the t-SNE [29] method over the acoustic feature means of different phonemes; Every $\rho_k^{(l)}$ was set to be 0 and $\boldsymbol{\sigma}_k^{(l)}$ was empirically set to be $\left[\sqrt{0.1}, \sqrt{0.1}\right]$. The cross-entropy DAMM model was initialised and well-tuned in the fashion of Algorithm 1. It introduced 5 hidden layers with 1024 nodes in each layer, which formed a 32×32 grid. The CE DAMM was used to generate the training-set lattices for the MPE training and the residual model of the well-trained CE DAMM was then tuned for 3 iterations under the MPE criterion to obtain the MPE DAMM system. In both CE and MPE training procedures, η was set to be 10^{-6}. In the adaptation of CE and MPE DAMM systems, the DAMMs were adapted to every testing utterance. The alignments of hypotheses generated by the respective SI DAMM were used to re-estimate the mean vector and covariance matrix of Gaussian components in the mixture model.

3.2. Results

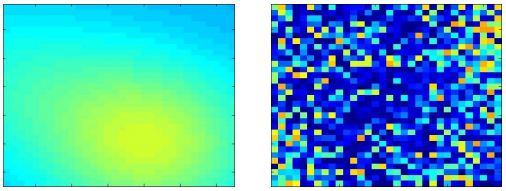
The word error rate (WER) comparison of cross-entropy SI systems is summarised in Table 1. The deep activation mixture

Table 1: *Cross-Entropy SI System Comparison.*

System	Dev03	Eval03
DNN	12.4	10.8
DAMM	**12.3**	**10.6**

model yielded a slightly better performance than the default DNN system. Figure 2 illustrates the first-layer activation behaviours of the mixture and residual model on one example frame. The mixture model in Figure 2(a) constructed an activation contour and the residual one in Figure 2(b) added a small variation to each activation, which was expected in this proposed model.

The SD performance of the adapted CE DAMM is given in Table 2, comparing the impacts of adapting the Gaussian mean vector $\boldsymbol{\mu}$, variance vector $\boldsymbol{\sigma}$ and correlation coefficient ρ. The change of $\boldsymbol{\sigma}$ applied homologous effects to activations located on nearby contour lines, while the move of $\boldsymbol{\mu}$ applied opposite effects to the activations on the same contour line, which could not correspond to the similarity of activations in the contour. Thus the adaptation on the covariance matrix yielded a more



(a) Mixture (b) Residual

Figure 2: *Grid Examples of DAMM Mixture and Residual Models on the First Hidden Layer.*

Table 2: *Adapted Cross-Entropy DAMM System Performance.*

System	Adapt μ	Adapt σ	Adapt ρ	Dev03	Eval03
SI	✗	✗	✗	12.3	10.6
SD	✓	✗	✗	12.2	10.6
	✗	✓	✗	12.1	10.5
	✗	✓	✓	12.1	10.5
	✓	✓	✗	12.1	**10.4**
	✓	✓	✓	**12.0**	**10.4**

effective impact than the mean vector. Via enabling the adaptation on all $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and ρ, the WER decreased by 0.3% and 0.2% on `Dev03` and `Eval03` respectively.

The SI MPE systems are compared in Table 3. The MPE DAMM yielded a similar performance as the MPE DNN base-

Table 3: *MPE SI System Comparison.*

System	Dev03	Eval03
DNN	**11.4**	10.1
DAMM	**11.4**	**10.0**

line. Table 4 summarises the adaptation performance of the MPE DAMM. The SD MPE DAMM on all the mean, variance

Table 4: *Adapted MPE DAMM System Performance.*

System	Adapt μ	Adapt σ	Adapt ρ	Dev03	Eval03
SI	✗	✗	✗	11.4	10.0
SD	✓	✓	✓	**11.1**	**9.8**

and correlation coefficient achieved further performance gains than the SI DAMM.

4. Conclusion

In this paper, we propose the deep activation mixture model for speech recognition. A mixture model and a residual model are introduced to jointly form the hidden activations. The mixture model defines a smooth activation contour and the residual model describes fluctuations around this contour. Meanwhile, this model can also be applied to rapid adaptation. The experiments were conducted on a large-vocabulary U.S. English broadcast news transcription task. The DAMM yields a slightly better performance than the DNN baseline on both the cross-entropy and MPE criteria. The utterance-level unsupervised adaptation on the DAMM can further acquire a lower-error performance.

5. References

[1] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[2] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[3] X. Chen, A. Eversole, G. Li, D. Yu, and F. Seide, "Pipelined back-propagation for context-dependent deep neural networks." in *INTERSPEECH*, 2012.

[4] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[5] S. Tan, K. C. Sim, and M. Gales, "Improving the interpretability of deep neural networks with stimulated learning," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, 2015, pp. 617–623.

[6] C. Wu, P. Karanasou, M. J. Gales, and K. C. Sim, "Stimulated deep neural network for speech recognition," in *Interspeech 2016*, 2016, pp. 400–404. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-580

[7] C. M. Bishop, "Mixture density networks," 1994.

[8] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Interspeech*, 2006.

[9] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3844–3848.

[10] E. Variani, E. McDermott, and G. Heigold, "A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4270–4274.

[11] S. Zhang, H. Jiang, and L. Dai, "Hybrid orthogonal projection and estimation (HOPE): a new framework to learn neural networks," *Journal of Machine Learning Research*, vol. 17, no. 37, pp. 1–33, 2016.

[12] A. van den Oord and B. Schrauwen, "Factoring variations in natural images with deep Gaussian mixture models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3518–3526.

[13] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[14] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6965–6969.

[15] P. Bell and S. Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task training," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4290–4294.

[16] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[17] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.

[18] 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.

[19] O. Abdel-Hamid and H. Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition." in *Interspeech*, 2013, pp. 1248–1252.

[20] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 171–176.

[21] C. Zhang and P. Woodland, "DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5300–5304.

[22] C. Wu and M. J. F. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4315–4319.

[23] C. Wu, P. Karanasou, and M. J. F. Gales, "Combining i-vector representation and structured neural networks for rapid adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5000–5004.

[24] T. Tan, Y. Qian, M. Yin, Y. Zhuang, and K. Yu, "Cluster adaptive training for deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4325–4329.

[25] M. Delcroix, K. Kinoshita, T. Hori, and T. Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4535–4539.

[26] P. Swietojanski and S. Renals, "Differentiable pooling for unsupervised speaker adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4305–4309.

[27] S. Tranter, M. Gales, R. Sinha, S. Umesh, and P. Woodland, "The development of the Cambridge University RT-04 diarisation system," in *Proc. Fall 2004 Rich Transcription Workshop (RT-04)*, 2004.

[28] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, A. Ragni, V. Valtchev, P. Woodland, and C. Zhang, "The HTK book (for HTK version 3.5)," 2015.

[29] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.