INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Adaptation of neural networks constrained by prior statistics of node co-activations

Tasha Nagamine, Zhuo Chen, Nima Mesgarani

Department of Electrical Engineering, Columbia University, New York, USA
tasha.nagamine@columbia.edu, zc2204@columbia.edu, nima@ee.columbia.edu

Abstract

We propose a novel unsupervised model adaptation framework in which a neural network uses prior knowledge of the statistics of its output and hidden layer activations to update its parameters online to improve performance in mismatched environments. This idea is inspired by biological neural networks, which use feedback to dynamically adapt their computation when faced with unexpected inputs. Here, we introduce an adaptation criterion for deep neural networks based on the observation that in matched testing and training conditions, the node co-activation statistics of each layer in a neural network are relatively stable over time. The proposed method thus adapts the model layer by layer to minimize the distance between the co-activation statistics of nodes in matched versus mismatched conditions. In phoneme classification experiments, we show that such node co-activation constrained adaptation in a deep neural network model significantly improves the recognition accuracy over baseline performance when the system is tested in various novel noises not included in the training.
Index Terms: online adaptation, deep neural networks, speech recognition, mismatched conditions

1. Introduction

One common issue that plagues practical applications utilizing deep neural network (DNN) systems is the mismatch between the training and real-world testing conditions. For example, in the application of automatic speech recognition, since the testing acoustic environment is difficult to pre-estimate, the performance generally suffers in the presence of unexpected noise. To improve model robustness, one popular scheme is to implement multi-condition training, in which improved generalization is achieved by training on various environments [1]. However, such schemes usually require a large amount of training data, which can be difficult to find for low-resource tasks. Another limitation of such an approach is that there is no guarantee of generalization to any particular unseen condition, even when networks are trained with largely enhanced data [2].

Model adaptation is another common technique for correcting the mismatch between training and test conditions. Currently in automatic speech recognition, model adaptation techniques for DNN-HMM acoustic models fall into three main categories [3]. The first is to apply a linear transformation to either the input features, softmax output, or hidden layer activations [4, 5, 6]. Adaptation may also be applied through conservative training, in which regularization to model weights or outputs is added to the adaptation criterion [7, 8]. Finally, subspace models construct a subspace for speaker or noise information, then adapt the network weights as a point in the subspace [1, 9, 10]. Such techniques can be applied in noise-robust speech recognition [1, 11], where it is common to implement noise-aware training. Here, features are appended to the standard Mel spectral or cepstral features that characterize the noise present in an utterance [1, 12, 13].

Network adaptation is also found ubiquitously in biological neural networks, which have the ability to quickly adapt to implement novel, task-related computations. For example, in the human auditory cortex, it has been shown that top-down, knowledge-driven global plasticity can facilitate the extraction of acoustic parameters relevant for a given task [14], such as the separation of a target signal from background noise [15]. Furthermore, it has been postulated that the correlational structure of neural activity aids in the separation of signals with temporal structure [16, 17].

Inspired by these biological mechanisms, in this study we propose a novel unsupervised neural network adaptation technique incorporating similar top-down feedback, which allows the network to compensate for the mismatch between training and testing conditions. Our technique is based on the notion that through initial training, the statistical properties of the output and hidden activations of a neural network can be calculated in ideal conditions. Consequently, in mismatched testing conditions, the network can use this prior knowledge of its expected hidden layer and output node co-activation patterns to update the transformation of an input signal by adapting the network weights to restore the expected co-activation patterns. This is accomplished by maximizing the similarity of the co-activation patterns of nodes in a layer to the reference calculated during training. Because this particular error metric is differentiable with respect to the network weights, optimization can be performed through backpropagation. We demonstrate the feasibility of node co-activation constrained adaptation (NCCA) on the task of phone recognition in neural networks trained on the TIMIT benchmark. Our results demonstrate the superior generalization power of networks adapted with NCCA in noisy conditions that were not included in the training of the network.

2. Proposed Model

Implementing NCCA in a neural network model requires three components: an unsupervised measure of network performance, an error metric that quantifies deviation from desired performance, and a method to adapt the network weights to minimize this error signal.

2.1. A statistical model of network activations

We create a statistical model of nodes in any layer ℓ of a neural network by defining a co-activation matrix of node responses Y over time, normalized over utterance duration T:

http://dx.doi.org/10.21437/Interspeech.2016-600

$$C^\ell = \frac{1}{T} Y Y^T, c_{ij}^\ell = \frac{1}{T} \sum_{\tau=1}^{T} \sum_{i,j=1}^{N} y_{i\tau}^\ell y_{j\tau}^\ell. \quad (1)$$

Here, i and j index over N nodes in a layer. This metric may have a particular interpretation depending on the task the network is designed for. In the example of a neural network acoustic model for speech recognition, the co-activation matrix of the output layer typically reflects monophone or triphone confusion patterns in the network, since classes that are more frequently jointly assigned weight in the posterior distribution are more likely to be confused. Likewise, because it has been shown that individual nodes in a neural network become selective to particular phonetic features [18], the co-activation of the hidden layers should encode information about the distributions of phones present in a speech signal. We quantify in section 3.2.2 the dependence of this statistic on signal duration T.

2.2. An unsupervised error signal to measure network performance

We define the error signal of the network E^ℓ in layer ℓ as the square of the Frobenius norm of the difference between the co-activation matrix C^ℓ and the expected co-activation matrix C_R^ℓ, which is computed during the training phase. This can be written as:

$$E^\ell = \sum_{i,j=1}^{N} \left(\frac{1}{T} \sum_{\tau=1}^{T} y_{i\tau}^\ell y_{j\tau}^\ell - c_{Rij}^\ell \right)^2. \quad (2)$$

The objective can be applied to the output distribution or the activations of any hidden layer.

2.3. Adapting the network weights to minimize the objective

An important property of the proposed error metric in (2) is that it is differentiable with respect to each node activation at all time points. Therefore, we can use the chain rule to compute the error derivative with respect to each weight in the network, then use gradient descent to backpropagate this unsupervised error to optimize the network parameters. At any time t, the partial derivative of the error E^ℓ in layer ℓ containing N nodes with respect to the activation y_k^ℓ of node k can be written:

$$\begin{aligned} \frac{\partial E^\ell}{\partial y_{kt}^\ell} &= \sum_{i,j=1}^{N} \frac{\partial}{\partial y_{kt}^\ell} \left(\frac{1}{T} \sum_{\tau=1}^{T} y_{i\tau}^\ell y_{j\tau}^\ell - c_{Rij}^\ell \right)^2 \\ &= 2 \sum_{i,j=1}^{N} \left(\frac{1}{T} \sum_{\tau=1}^{T} y_{it}^\ell y_{jt}^\ell - c_{Rij}^\ell \right) \frac{\partial}{\partial y_{kt}^\ell} \left(\frac{1}{T} \sum_{\tau=1}^{T} y_{i\tau}^\ell y_{j\tau}^\ell - c_{Rij}^\ell \right). \end{aligned} \quad (3)$$

Any matrix element c_{Rij}^ℓ of the reference co-activation matrix C_R^ℓ is invariant with respect to the node activation y_{kt}^ℓ. Additionally, the derivative of the first term is only nonzero when $t = \tau$ and $i \cap j = k$. Thus,the above expression can be written in terms of the Kronecker delta function:

$$\begin{aligned} \frac{\partial E^\ell}{\partial y_{kt}^\ell} &= \frac{2}{T} \sum_{i,j=1}^{N} \left(\frac{1}{T} \sum_{\tau=1}^{T} y_{i\tau}^\ell y_{j\tau}^\ell - c_{Rij}^\ell \right) \left(y_{jt}^\ell \delta_{ik} + y_{it}^\ell \delta_{jk} \right) \\ &= \frac{2}{T} \sum_{i,j=1}^{N} \Delta_{ij}^\ell \left(y_{jt}^\ell \delta_{ik} + y_{it}^\ell \delta_{jk} \right), \end{aligned} \quad (4)$$

where Δ_{ik}^ℓ is defined as $\frac{1}{T} \sum_{\tau=1}^{T} y_{i\tau}^\ell y_{j\tau}^\ell - c_{Rij}^\ell$. Because the double sum is only nonzero for $i \cap j = k$, and using the symmetric properties of Δ_{ik}^ℓ, we can combine these sums into one simplified expression, $\frac{\partial E^\ell}{\partial y_{kt}^\ell} = \frac{4}{T} \sum_{j=1}^{N} \Delta_{kj}^\ell y_{jt}^\ell$, which defines the error with respect to the activation of node k at one particular time point. Averaging this value over the duration of an utterance yields the expected error over time $\frac{\partial E^\ell}{\partial y_k^\ell}$. Using traditional backpropagation with learning rate α, we can adjust the weight w between node k in layer ℓ and node i in layer $\ell - 1$ in the network using:

$$w_{ik} = w_{ik} - \alpha \frac{\partial E^\ell}{\partial y_k^\ell} y_k^\ell \left(1 - y_k^\ell \right) y_i^{\ell-1}. \quad (5)$$

3. Experiments and results

To provide an intuitive account of how node co-activation constrained adaptation (NCCA) works in a neural network model, we first incorporate the proposed feedback in a basic autoencoder network [19]. We then demonstrate the efficacy of node co-activation constrained adaptation in deep neural network acoustic models trained for phoneme recognition.

3.1. Autoencoder

We incorporate our adaptation scheme into a feed-forward autoencoder network on the TIMIT speech corpus [20] to reconstruct an input spectrogram X to a reconstructed output Z using the loss function $L = ||X - Z||_F^2$. The input and output of the autoencoder network are 257 frequency channels, and the network has one hidden layer consisting of 128 nodes. All nonlinearities consisted of the hyperbolic tangent function. Features were extracted using log-scale spectrograms, then scaled in the range $[-1, 1]$ to fall within the operating range of the nonlinearity. The reference statistic of the hidden layer C_R^{HL} was computed during training.

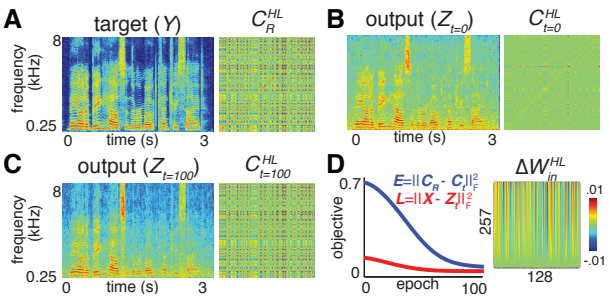

Figure 1: *An autoencoder adapted with NCCA. (A-C) Spectrograms and co-activation matrices C_t^{HL} for NCCA epoch t for the clean target spectrogram Y, noisy network output $Z_{t=0}$, and network output $Z_{t=100}$ after adaptation. (D) Training objective L and NCCA objective E as a function of adaptation epoch. Shown at right is the change in the weights ΔW.*

The autoencoder was trained only on clean speech, with the result that distortions in the input space are faithfully mapped to the output. Figure 1 shows the result of introducing NCCA into the autoencoder using additive noise from the NOISEX-92 database [24] (white noise at an SNR = 10dB, with C^{HL} calculated using 64 random utterances from the TIMIT test set). By applying NCCA on the weight matrix W_{in}^{HL} connecting the input and hidden layer of the network, we show that distor-

tions can be significantly suppressed in the reconstructed outputs. Before adaptation ($t = 0$), the co-activation matrix of the network output $C^{HL}_{t=0}$ in noise differs significantly from the reference C^{HL}_R calculated from a clean utterance (Figure 1A-B). Here, feedback is applied for 100 epochs of backpropagation using equation (5). After each epoch, weights are adjusted and the co-activation matrix C^{HL}_t is recomputed. Through feedback, the network is able to recover the statistical structure of its activations in the clean condition (Figure 1C). We show in Figure 1D that the typical training loss function $L = ||X - Z_t||^2_F$ is closely related to the true objective E in equation (2). It is worth emphasizing that the network has no knowledge of the noise; it only uses the expected statistics of the output in clean conditions to re-wire itself, which results in suppression of unwanted variability. The change in the weight matrix from hidden to output layer (Δw_{ik}) is shown in Figure 1D.

3.2. Neural network acoustic models

3.2.1. Neural network architecture

We incorporate our proposed model into feed-forward neural networks trained for phone recognition on the TIMIT speech corpus [20] with utterances common to all speakers (SA utterances) excluded. Networks were trained on two targets: monophone outputs (144 nodes corresponding to the beginning, middle, and end of 48 phone labels) and triphones (1888 context-dependent outputs). The phone labels used were mapped from the original 61 labels of TIMIT to a subset of 48 for the softmax posterior distribution, then to 39 labels for scoring [25]. Forced alignments were obtained with the TIMIT s5 mono and tri1 recipes from the Kaldi speech recognition toolkit [26]. Experiments were performed on three acoustic models: two context-independent monophone output networks with five hidden layers each and 256 and 512 nodes per hidden layer, respectively, and one context-dependent triphone output network with four hidden layers and 2048 nodes per hidden layer.

Models were trained using Theano [23] using a sigmoid nonlinearity and a 20% and 50% dropout rate on inputs and hidden layers, respectively [27, 28]. The reference statistic C^{ℓ}_R for the output and each hidden layer ℓ was computed during training. The input features to all networks were 11 shifted frames of 13-dimensional log Mel filter-bank coefficients with appended deltas and double-deltas (429 dimensions total) with applied cepstral mean and variance normalization. Decoding to obtain phone error rate (PER) was performed using bigram language models. Feature extraction and decoding were performed with Kaldi. All reports of frame-wise phone accuracy and PER are reported for the core TIMIT test set (24 speakers, 192 utterances). An additional 50 speakers (400 utterances) were excluded from training and used as a validation set.

To determine the efficacy of the adaptation scheme, noise was artificially added to the test and validation sets from samples of white, pink, babble, and destroyer engine noise from the NOISEX-92 database [24] at SNRs of 0, 10, and 20 dB.

3.2.2. Adaptation utterance batch size estimation

Node co-activation constrained adaptation (NCCA) is successively implemented in each layer of the network (Figure 2), where each layer regulates itself by changing the weight parameters connected to the preceding layer. Because the co-activation matrix C^{ℓ} of node activations in response to speech is used in our error metric, it is key to evaluate the dependence of this statistic on the duration of the sample of speech used to generate it. To determine what signal duration is needed to obtain a good estimate of C^{ℓ}, we calculated the average frame-wise classification accuracy for all noise types and SNR as a function of NCCA epoch for test co-activation C^{ℓ} calculated from utterance batch sizes $n \in [2, 4, 16, 32, 64]$. The learning rate was kept fixed at 0.001. Figure 3A shows classification accuracy as a function of batch size after adaptation in the first hidden layer of a monophone network with 256 nodes per hidden layer. We can see that performance is improved for larger batch sizes, and that for small batch sizes ($n < 4$), the computed $C^{\ell=1}$ is not representative of the training statistics, resulting in severe overfitting. However, with enough data ($n > 4$), we observe that minimizing the unsupervised error E does indeed result in improved frame accuracy. Figure 3B confirms that increasing error in our objective in equation (2) reliably correlates with lower classification accuracy for noises at various SNRs.

It is intuitive that incorporating adaptation into the output layer of the network will improve classification accuracy, since the objective at the output forces the output probability distribution to be closer to that of the unadapted model. However, the adaptation scheme outlined in Figure 2 also places the additional constraint on the system that the correlation within hidden layer activations should be restored. We demonstrate that performing adaptation layer-wise starting from the first hidden layer also improves performance (Figure 3C).

Finally, we wanted to determine if output layer size affected the signal length needed to obtain a good estimate of the output layer statistics. To do this, for the 512 node per hidden layer monophone model and the triphone output model, we computed C^{out} and calculated the correlation of this co-activation matrix to the reference matrix C^{out}_R for various batch sizes, with batches randomly sampled from the test set. Figure 3D shows that with 64 utterances, an accurate estimation can reliably be obtained for both a monophone and triphone model at the output layer. The number of utterances needed for both models is very similar, especially with increasing batch size n; this may be due to the fact that many of the triphone output states occur infrequently. Thus, for the remainder of the experiments in this study, we utilized a batch size of $n = 64$, which corresponds to approximately three minutes of speech (Figure 3E).

3.2.3. Experimental evaluation

We evaluated our adaptation scheme for the task of improving phoneme recognition in unseen noises. We did so in two neural network models: the monophone model with 512 nodes per hidden layer (100 adaptation epochs, base learning rate $\alpha_0 = 0.005$) and the triphone model (50 adaptation epochs, base learning rate $\alpha_0 = 0.0025$). Because the gradient of the

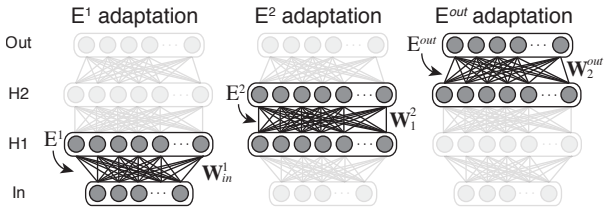

Figure 2: *The sequence of operations for implementing node co-activation constrained adaptation (NCCA) in a neural network with two hidden layers. Adaptation is applied sequentially in all hidden layers, then in the output layer.*

Network	SNR	BL	White +NCCA	BL	Pink +NCCA	BL	Babble +NCCA	BL	Destroyer +NCCA	Average relative PER reduction (all noises)
	∞	26.7%								
512 nodes, 5 layers	20 dB	38.8%	33.8% (12.9%)	34.9%	32.3% (7.4%)	34.4%	32.7% (4.9%)	33.6%	31.9% (5.1%)	7.6%
monophone output	10 dB	55.1%	45.4% (17.6%)	50.3%	43.4% (13.7%)	49.0%	46.6% (4.9%)	47.3%	42.4% (10.4%)	11.7%
	0 dB	69.0%	61.2% (11.3%)	68.1%	61.3% (10.0%)	69.5%	65.9% (5.2%)	66.5%	62.3% (6.3%)	8.2%
										9.2%
	∞	22.9%								
2048 nodes, 4 layers	20 dB	37.2%	33.1% (11.0%)	32.5%	30.0% (7.7%)	31.0%	30.1% (2.9%)	31.0%	28.8% (7.1%)	7.2%
triphone output	10 dB	57.9%	50.4% (13.0%)	53.5%	45.4% (15.1%)	49.4%	45.6% (7.7%)	47.9%	43.2% (9.8%)	11.4%
	0 dB	74.1%	67.1% (9.4%)	74.6%	66.0% (11.5%)	73.6%	68.2% (7.3%)	70.8%	65.3% (7.7%)	9.0%
										9.2%

Table 1: *Model performance measured in phone error rate (PER) for model baseline performance in noise (BL) and after incorporation of node co-activation constrained adaptation (+NCCA). PER for the clean condition is shown by $SNR = \infty$. Relative PER reduction is shown in parentheses for each condition; grand average relative PER reduction over all conditions is shown in bold.*

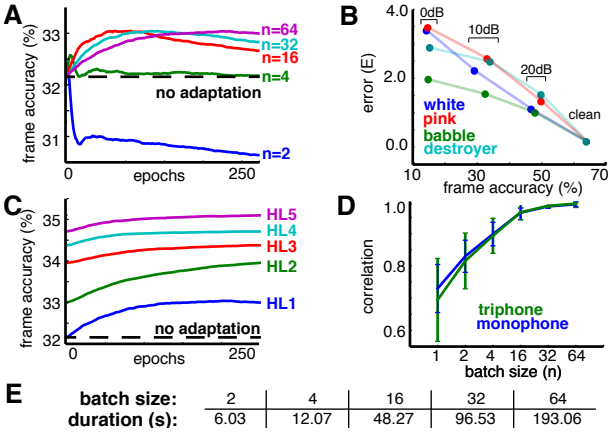

Figure 3: *(A-C) Monophone DNN (256 nodes). (A) Frame accuracy after adaptation in hidden layer 1 for various utterance batch sizes n as a function of NCCA epoch, averaged over noise types and SNR. (B) Relationship between error E and frame accuracy for various noise types and SNRs (test set). (C) Frame accuracy with adaptation applied sequentially in all hidden layers. (D) Correlation between C_R^{out} and C^{out} for batch size n for monophone (512 nodes) and triphone outputs. Error bars show standard deviation. (E) Corresponding duration (s) for batch size n.*

error E^ℓ tends to drop in magnitude for deeper layers, leading to reduced error gradients and smaller adaptation effects, the learning rate for each layer was set to $\alpha^\ell = \alpha_0 / E^\ell_{t=0}$ with the constraint $\alpha^\ell < 250$. Doing so was generally found to improve performance over a fixed learning rate. To prevent overfitting, we also used a validation set with batch size matched to the adaptation utterances and stopped adaptation in any layer if the validation error did not decrease for 10 consecutive epochs.

Table 1 shows the results after incorporating NCCA into both neural network models. Baseline phone error rate (PER) in clean conditions (SNR = ∞) for the monophone and triphone models is 26.7% and 22.9%, respectively. For the triphone model, this is comparable to the performance of the Kaldi s5 DNN implementation (23.0% PER on the test set, Dan's DNN). For each noise type and SNR, baseline performance (BL) is compared with results from the adapted model (+NCCA), with relative PER reduction shown in parenthesis. Over all noise types and SNRs, the average relative PER reduction was 9.2% in both models. We should also note that PER reduction was greater for more stationary noises such as white and pink noise, although improvements were seen for all noise types.

4. Conclusions

In this study we introduced a novel unsupervised neural network adaptation technique, which we have termed node co-activation constrained adaptation (NCCA). Our method works by placing a constraint on the co-activation pattern of nodes in the hidden and output layers of the network, which forces activations in each layer to be similar to those of the unadapted model. We show that this technique is effective in an autoencoder model and in a neural network phone recognition task, where in a large, context-dependent DNN we obtained an average relative PER reduction of 9.2% across a variety of artificial noisy conditions.

This objective differs from other adaptation techniques such as unsupervised KL-divergence regularization [8] because while the interpretation of the adaptation criterion at the output layer is similar, the way in which it is accomplished is fundamentally different. Our model places constraints on co-activation statistics rather than probability distributions, meaning that it can be applied in an unsupervised manner to output and any hidden layers. Additionally, our technique utilizes second-order moments, meaning that it is possible to recover corrupted or missing activation patterns as long as they are correlated with other, less noisy node activations.

Future directions for this work in DNN acoustic models include extension into larger datasets with various mismatched train and test conditions while implementing hyper-parameter optimization [29, 30] or low-rank matrix factorization techniques [10, 31]. It is also possible to loosen the statistical constraint on hidden layer activations by allowing the complete backpropagation of error as adaptation is applied sequentially in the network, or to apply NCCA in other DNN architectures. We would also like to emphasize that although we have demonstrated this technique in noisy phone recognition, NCCA is generalizable to any neural network where node activations of any hidden or output layer encode predictable temporal statistics. This applies to both discriminative and generative models in a variety of applications.

5. Acknowledgements

We thank Winston Mann and Prof. John Wright for productive discussions. This work was funded by a grant from the National Institute of Health, NIDCD, DC014279, National Science Foundation CAREER Award, and the Pew Charitable Trusts.

6. References

[1] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 26–31, Vancouver, Canada, Proceedings*, 2013, pp. 7398–7402.

[2] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in English and Mandarin," *arXiv:1512.02595*, 2015.

[3] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. New York: Springer, 2015.

[4] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," in *Fourth European Conference on Speech Communication and Technology (EUROSPEECH), September 18–21, Madrid, Spain, Procedings*, 1995, pp. 2171–2174.

[5] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, pp. 827–835, 2007.

[6] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *IEEE Spoken Language Technology Workshop (SLT), December 2–5, Miami, Florida, Proceedings*, 2012, pp. 366–369.

[7] X. Li and J. Bilmes, "Regualized adaptation of discriminative classifiers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 14–19, Toulouse, France, Proceedings*, 2006, pp. I–237–240.

[8] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 26–31, Vancouver, Canada, Proceedings*, 2013, pp. 7893–7897.

[9] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding Workshop (ASRU), December 8–12, Olomouc, Czech Republic, Proceedings*, 2013.

[10] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 4–9, Florence, Italy, Proceedings*, 2014, pp. 6359–6363.

[11] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 745–777, 2014.

[12] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 19–24, Brisbane, Australia, Proceedings*, 2015, pp. 5014–5018.

[13] S. Kim, B. Raj, and I. Lane, "Environmental noise embeddings for robust speech recognition," *arXiv:1601.02553*, 2016.

[14] J. Fritz, S. Shamma, M. Elhilali, and D. Klein, "Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex," *Nature Neuroscience*, vol. 6, no. 11, pp. 1216–1223, 2003.

[15] N. Mesgarani and E. F. Chang, "Selective cortical representation of attended speaker in multi-talker speech perception," *Nature*, vol. 485, pp. 233–236, 2012.

[16] A. K. Barros and A. Cichoki, "Extraction of specific signals with temporal structure," *Neural Computation*, vol. 13, no. 9, pp. 1995–2003, 2001.

[17] M. Elhilali, L. Ma, C. Micheyl, A. J. Oxenham, and S. A. Shamma, "Temporal coherence in the perceptual organization and cortical representation of auditory scenes," *Neuron*, vol. 61, no. 2, pp. 317–329, 2009.

[18] T. Nagamine, M. L. Seltzer, and N. Mesgarani, "Exploring how deep neural networks form phonemic categories," in *INTERSPEECH 16th Annual Conference of the International Speech Communication Association, September 6–10, Dresden, Germany, Proceedings*, 2015, pp. 1912–1916.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural netowrks," *Science*, vol. 313, pp. 504–507, 2006.

[20] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallet, and N. L. Dahlgren, "TIMIT acoustic-phonetic continuous speech corpus," *LDC93S1*, 1993.

[21] P. Vincent, H. Larochelle, Y. Bengion, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *International Conference on Machine Learning (ICML), July 5–9, Helsinki, Finland, Proceedings*, 2008, pp. 1096–1103.

[22] X. Feng, Y. Zhang, and J. Glass, "Ieee international conference on acoustics, speech, and signal processing (icassp), may 4–9, florence, italy, proceedings," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 4–9, Florence, Italy, Proceedings*, 2014.

[23] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation.

[24] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, pp. 247–251, 1993.

[25] K. fu Lee and H. wuen Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 10, pp. 1641–1648, 1989.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembeck, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz, "The Kaldi speech recognition toolkit," in *Automatic Speech Recognition and Understanding Workshop (ASRU), December 11–15, Waikoloa, Hawaii, Proceedings*, 2011, pp. 1912–1916.

[27] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by prevent co-adaptation of feature detectors," *arXiv:1207.0580*, 2012.

[28] N. Srivastava, G. 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, pp. 1929–1958, 2014.

[29] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[30] J. Snoek and H. Larochelle, "Practical bayesian optimization of machine learning algorithms," *arXiv:1206.2944*, 2012.

[31] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 26–31, Vancouver, Canada, Proceedings*, 2013, pp. 6655–6659.