IMPROVING DEEP NEURAL NETWORKS FOR LVCSR USING DROPOUT AND SHRINKING STRUCTURE

Shiliang Zhang[1], *Yebo Bao*[1], *Pan Zhou*[1], *Hui Jiang*[2], *Lirong Dai*[1]

[1]National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, Anhui, P. R. China
[2]Department of Electrical Engineering and Computer Science, York University, Toronto, Canada
Email: {zsl2008,bybillow,pan2005}@mail.ustc.edu.cn, hj@cse.yorku.ca, lrdai@ustc.edu.cn

ABSTRACT

Recently, the hybrid deep neural networks and hidden Markov models (DNN/HMMs) have achieved dramatic gains over the conventional GMM/HMMs method on various large vocabulary continuous speech recognition (LVCSR) tasks. In this paper, we propose two new methods to further improve the hybrid DNN/HMMs model: i) use dropout as pre-conditioner (DAP) to initialize DNN prior to back-propagation (BP) for better recognition accuracy; ii) employ a shrinking DNN structure (sDNN) with hidden layers decreasing in size from bottom to top for the purpose of reducing model size and expediting computation time. The proposed DAP method is evaluated in a 70-hour Mandarin transcription (PSC) task and the 309-hour Switchboard (SWB) task. Compared with the traditional greedy layer-wise pre-trained DNN, it can achieve about 10% and 6.8% relative recognition error reduction for PSC and SWB tasks respectively. In addition, we also evaluate sDNN as well as its combination with DAP on the SWB task. Experimental results show that these methods can reduce model size to 45% of original size and accelerate training and test time by 55%, without losing recognition accuracy.

Index Terms— dropout, dropout as pre-conditioner (DAP), shrinking hidden layer, deep neural networks, LVCSR, DNN-HMM

1. INTRODUCTION

Recently, deep neural networks (DNN) have become the state-of-the-art acoustic modelling in large vocabulary continuous speech recognition (LVCSR) systems. Results in [1, 2, 3, 4, 5, 6] show that the hybrid deep neural networks and hidden Markov models (DNN/HMMs) can lead to significant performance improvement over the conventional acoustic models using continuous density HMMs based on Gaussian mixture models (GMMs). Neural networks revived in speech recognition due to a few pre-training algorithms proposed in 2006, such as deep belief networks (DBN) [7] and stacked auto-encoders [8]. Generally speaking, in all instances of neural networks, the objective function is a highly non-convex function of the parameters, where many distinct local minima coexist in the model parameter space. The role of unsupervised pre-training is to guide learning towards basins of attraction of minima that may yield better generalization for the unknown test data [9]. The generalization of model may be poor if training is trapped in any bad local minima or over-fitted to training data. Even with unsupervised pre-training, deep neural networks with a huge number of parameters still suffer from the serious problem of over-fitting, especially when training sets are small. In traditional back-propagation (BP), weight decay regularization terms and early stopping are used to control over-fitting for models with many parameters. In [10], *Hinton at el.* present a *dropout* strategy by randomly omitting a fraction of the hidden units in all layers for each training sample to prevent the training from over-fitting, which gives big improvements in the TIMIT corpus. In [11, 12] have also investigated dropout training on small corpora. However, when dropout was used to LVCSR tasks, it was likely to only increase training time [13]. As opposed to other dropout work in the literature, we propose to use dropout as pre-conditioner (DAP), which guides the parameter values into an appropriate range for further supervised fine-tuning. In [14], it gave a preliminary trial of this idea but our study in this paper is more thorough and it also has some main differences from [14]. For example, we propose to eliminate the necessity to perform the time-consuming pre-training at the beginning and it only requires about 20 or even fewer iterations of dropout based training, which is far fewer than the previously reported work. As a result, the total training time is significantly reduced and the procedure can be applied to large scale speech recognition tasks within a reasonable amount of training time.

On the other hand, the unsupervised pre-training may act as a regularizer that increases the magnitude of weights [9]. But even with pre-training, the fully connected DNNs after back-propagation fine-tuning tend to be severely sparse, where a large portion of network connections usually have extremely small weights. Results in [15] show that the magnitude of 70% of weights is actually below 0.1 in a well-trained 7-hidden-layer DNN model. In [15], the sparseness of DNNs is explored to reduce model size but it fails to expedite computation time of DNNs because the resultant sparse matrices may leads to random memory accesses that dramatically slow down CPU/GPU execution. Along this line, low-rank weight matrix factorization in [16] and singular value decomposition of weight matrices in [17] have been proposed to take advantage of DNN sparseness for faster computation. In this work, we propose a more straightforward way to explore DNN sparseness that results in similar or even larger reduction in both model size and computation time. The motivation is that we have observed that connection weights in upper layers of DNN tend to be much sparser than those in the lower layers. The sparseness nature of DNN weights inspires us to adopt a novel network structure named shrinking hidden layer DNN (sDNN), where hidden layers gradually decrease in number of hidden nodes from bottom to upper layers, to pursue smaller model

This work was partially funded by the National Nature Science Foundation of China (Grant No. 61273264) and the National 973 program of China (Grant No. 2012CB326405).

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

size and faster computation.

In this paper, we evaluate effectiveness and efficiency of DAP and sDNN on a 70-hour Mandarin transcription PSC task and the 309-hour Switchboard (SWB) task. Firstly, we investigate suitable parameter configuration for DAP on the PSC task. Experiments show that it can yield relative error reduction of 11.3% over the traditional pre-trained DNNs. Secondly, we examine the proposed methods for the larger SWB task, where we obtain 6.8% relative performance improvement in WER. To the best of our knowledge, this is the first successful application of dropout methods to DNN training for hundreds of hours LVCSR tasks where significant performance improvement has been observed. Finally, we combine DAP and sDNN for the SWB task, which leads to model reduction to 45% of original size and computation speedup by more than 55%, without sacrificing recognition accuracy.

2. DROPOUT AS PRE-CONDITIONER

Dropout is a powerful technology introduced in [10] for improving generalization capability of neural networks. During training, dropout can reduce over-fitting by randomly omitting a fraction of the hidden units in all layers on each training case to prevent co-adaptation of hidden units. In our work, we investigate a new application of dropout that can be regarded as pre-conditioner, putting the parameter values in an appropriate range for further supervised fine-tuning. Compared with conventional unsupervised pre-training [7, 8, 18, 19], dropout as pre-conditioner (DAP) can be viewed as a method of supervised pre-training. In this case, the training procedure is composed of two stages: i) using dropout based fine-tuning for certain number of iterations to generate an initial DNN; ii) simple back-propagation (BP) to adjust the parameters of the DNN.

Considering a traditional feed-forward neural network with L hidden layers, where we use $l \in \{1, ..., L\}$ to index all hidden layers of the networks, $\mathrm{y}^{(l)}$ denote the output vector of hidden layer l, $\mathrm{W}^{(l)}$ and $\mathrm{b}^{(l)}$ for weights and biases of hidden layer l. The forward procedure of dropout fine-tuning takes the following form:

$$r^{(l)} \sim \text{Bernoulli}(p) \tag{1}$$

$$y^{(l+1)} = f(r^{(l)} * y^{(l)} W^{(l)} + b^{(l)}) \tag{2}$$

where $f(\cdot)$ is the sigmoid activation function and $r^{(l)}$ is a vector of Bernoulli random variables, each of which has probability of p being 0 and probability $1-p$ being 1, and we use $*$ to denote element-wise multiplication of vectors. For backward procedure, the derivatives of loss function are back-propagated through the networks with weights update taking the following form:

$$\Delta w^{t+1} = m^{t+1} \Delta w^{t} - \varepsilon(1 - m^{t+1}) \langle \nabla_w L \rangle \tag{3}$$

$$m^t = \begin{cases} \frac{t}{T} m_f + (1 - \frac{t}{T}) m_i & t < T \\ m_f & t \geq T \end{cases} \tag{4}$$

Here, $m_i = 0.5, m_f = 0.9, T = 10, \varepsilon = 2$, and $\langle \nabla_w L \rangle$ denotes the average gradient of the objective function to the parameters of current layer within one mini-batch, ε denotes the initial learning rate, which is set to be much larger than that of traditional BP. Using large learning rate allows a far more thorough search of the weight-space [9], and enables the model parameters to jump from one basin to another. But the randomly dropping out units (hidden and visible) in a neural network may add a particular type of noise to the hidden unit activations during the forward pass of training [13]. Unfortunately, adding so much noise may slow down learning or even make it difficult to converge to a local optimum. That is why the pure dropout-based fine-tuning seems to only increase training time for LVCSR tasks. As a result, we propose to further adjust the neural networks with traditional BP fine-tuning after a number of dropout fine-tuning iterations. Note that neither the time-consuming RBM-based pre-training nor the model averaging is used in our method.

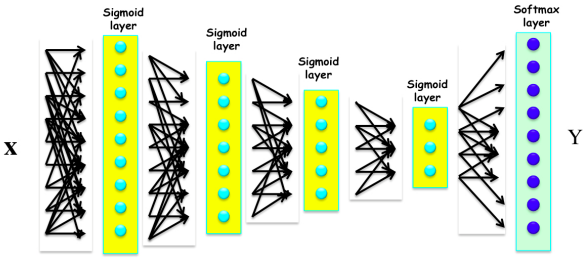

Fig. 1. Illustration of sDNN with four hidden layers.

3. DNN WITH SHRINKING HIDDEN LAYERS

The state-of-the-art DNN/HMMs systems often use feed-forward neural networks where all hidden layers have the same number of hidden nodes. In our work, we have observed that when DNN is trained with many hidden layers, weights in the networks are very sparse, which may become more apparent as we increase the number of hidden layers and hidden units. Results in [15] show that magnitude of 70% of network connection weights in a well-trained 7-hidden-layer DNN is below 0.1. Meanwhile, small weights in DNN make fairly weak contribution to the outputs of hidden layers that are the weighted sums of the previous layer outputs followed by monotonic nonlinear transformation. From this viewpoint, it can be regarded as redundancies and thus can be abandoned for the purpose of reducing model size and expediting computation.

On the other hand, DNN can be considered as a highly complex nonlinear feature extractor, and all units are learnt to represent features that capture higher order correlations in the original input data. The lower layer units can be regarded as feature detectors to capture raw features in training data. We can achieve more discriminative and invariant features through many layers of nonlinear transformation. During this process, DNN has the ability to capture the useful discriminative features and filter irrelevant information. Therefore, we should be able to use fewer units in upper layers to simply model structure and reserve critical information at the same time.

Based on the above reasons, we propose to use a special network structure with shrinking hidden layers (sDNN), where hidden layers gradually decrease in size from bottom to top. For example, Fig. 1 illustrates a structure of 4-hidden-layer sDNN. By removing redundancies in the weights, we can boost training efficiency by using smaller model size and less multiplications. For example, in our work, it typically takes about 10 hours per epoch to train a 6-hidden-layer DNN with 2,048 nodes per layer using a single GeForce 690 GPU. While using sDNN with shrinking structure of 2048-1792-1536-1280-1024-768 can reduce the training time to 5.5 hours per epoch with no loss of performance at all. Moreover, it can also speed up computation of the posterior probabilities for decoding, which is important for real-time speech recognition.

Table 1. Performance comparison (CER in %) of various baseline models in Mandarin PSC task.

	MLE	MPE
GMM-HMM	18.2	16.7
hidden layers	3	5
DNN (pre-train)	14.0	13.2
DNN (random)	14.8	14.2

4. EXPERIMENTS

In this paper, we evaluate the proposed DAP and sDNN on two LVCSR tasks, namely 70-hour Mandarin transcription task and the 309-hour Switchboard task. Moreover, we also evaluate the combination of these two methods on the Switchboard task.

4.1. Mandarin transcription task

For the Mandarin transcription task, the training set contains 76,858 utterances (about 70 hours) from 1,539 speakers. Evaluation is measured in terms of character error rate (CER) on a separate 3-hour test set, consisting of 3,720 utterances from other 50 speakers. In this task, we investigate the appropriate parameters settings for DAP.

4.1.1. Baseline Systems

First of all, we build the baseline GMM/HMMs system based on the standard tied-state cross-word tri-phone models. We use the regular 43-dimension features as input, including 39-dimension MFCC features (static, first and second derivatives) and 4-dimension pitch features. The features are pre-processed with cepstral mean normalization (CMN) algorithm. The baseline models, including 3,969 tied HMM states of 30 Gaussian components, are first trained based on maximum likelihood estimation (MLE) and then discriminatively trained using minimum phone error (MPE) criterion. In the second row of Table 1, we give performances of these baseline GMM/HMMs systems. For the hybrid DNN/HMMs baseline system, we use the best training configurations in [20]. The inputs are concatenated from all consecutive frames within a long context window of (5+1+5). DNN is first pre-trained using RBM-based greedy layer-wise pre-training and then fine-tuned using state labels obtained through forced alignment of MLE trained GMM/HMMs. For fine-tuning, we use 10 training epochs of the whole training set. An initial learning rate of 0.2 is kept constant for the first four epochs and is halved for each of the remaining epochs. We have trained DNN with 3 and 5 hidden layers with 2,048 nodes per layer. The performances are listed in the fourth row of Table 1. For comparison, we also trained the same DNN with Gaussian random initialization. Results in Table 1 show that pre-trained DNNs can achieve significant performance improvement over Gaussian random initialization in this task.

4.1.2. Dropout as pre-conditioner

In this section, we study how to use DAP for LVCSR from three different aspects.

1) *Varying dropout probability and learning rate*: We first use different dropout probabilities and initial learning rates to fine-tune a 3-hidden-layer Gaussian random initialized DNN, which contains 2,048 units in each hidden layer. For all experiments, the dropout probability of input layer is 20%. Fig.2 illustrates that the classification error curve converges faster as dropout probability decreases and initial learning rate increases. Table 2 shows that after 20 epochs of dropout epochs and 10 epochs of standard BP, we can achieve the lowest CER of 12.3% with hidden layer dropout probability of 30% and initial learning rate of 2.

Table 2. Performance comparison (CER in %) of DAP using different dropout probabilities and initial learning rates in Mandarin PSC task. (DNN contains 3 hidden layers and 2,048 nodes per layer)

Method	dropout epochs			
Hid_omit=0.5	epoch	10	20	60
Learning rate=1	CER	13.4	**13.1**	12.4
Hid_omit=0.3	epoch	10	20	40
Learning rate=1	CER	13.0	**12.7**	12.5
Hid_omit=0.3	epoch	5	10	20
Learning rate=2	CER	13.2	12.9	**12.3**
Hid_omit=0.1	epoch	5	10	20
Learning rate=1	CER	13.4	13.1	**12.7**

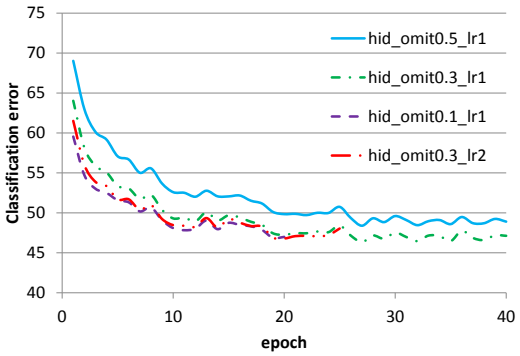

Fig. 2. Held-out set frame classification error (%) curves of different hidden layer dropout probabilities and initial learning rate. "hid_omit0.3_lr1" denotes the dropout probability of all hidden layers is 30% and initial learning rate is 1.

2) *Different initial methods* : Secondly, we investigate different initialization methods for dropout: i) model average; ii) pre-training. In the original dropout work, DNN weights need to be multiplied by (1-p) at test time, which can be treated as an effective way to perform model averaging. Therefore, we examine whether it needs to perform model averaging before using the dropout fine-tuned DNNs for BP. As seen from Table 3, without using model averaging we obtain CER of 12.3%, which is slightly better than 12.4% of performing model averaging before standard BP. Moreover, it is not a good idea to use the RBM-based pre-training prior to dropout based preconditioner. These results suggest that it is not essential to use model averaging after dropout and meanwhile the time-consuming RBM-based pre-training process can be completely eliminated. From this point, we may view dropout fine-tuning as a pre-conditioner to generate a good initialization for standard BP.

3) *Varying hidden layer number*: We use the best experimental configuration obtained from above experiments to train DNN with 3-hidden-layer and 5-hidden-layer, which both contain 2,048 nodes in each layer. Table 4 depicts that if 10 epochs of traditional BP are executed after 20 iterations of dropout fine-tuning, we can achieve 9.6% relative error reduction over pre-trained DNN. Moreover, even with only 10 epochs of dropout fine-tuning, we can still achieve 7.2% relative error reduction.

Table 3. Character error rates (CER in %) of DAP using different initial methods in Mandarin PSC task. (DNN contains 3 hidden layers and 2,048 nodes per layer; PT for pre-training)

dropout epochs	5	10	20
no PT, no model average	13.2	12.9	**12.3**
no PT, model average	13.2	12.9	**12.4**
PT, no model average	13.2	12.8	**12.6**

Table 4. Character error rates (CER in %) of DAP with varying number of hidden layers in Mandarin PSC task.

hidden layers	3		5	
	CER	gain	CER	gain
Baseline: Pre-DNN	14.0	-	13.2	-
DAP epoch5 +BP	13.2	5.4%	12.7	3.7%
DAP epoch10+BP	12.9	8.1%	12.2	7.2%
DAP epoch20+BP	**12.3**	**11.3%**	**11.9**	**9.6%**

4.2. Switchboard task

For the Switchboard (SWB) task, the training data consists of 309-hour Switchboard-I training set and 20-hour Call Home English data. Evaluation is measured in terms of word error rate (WER) on the NIST 2000 Hub5 evaluation set, denoted as Hub5e00. In the SWB task, we first evaluate DAP for better accuracy and then consider to combine DAP with sDNN for better efficiency.

4.2.1. Baseline systems

The baseline GMM/HMMs systems are standard tied-state cross-word triphone system estimated with MLE and MPE criteria using 39-dimension PLP features (static, first and second derivatives) that are pre-processed with cepstral mean and variance normalization (CMVN) per conversation side. The HMM consists of 8,991 tied states and 40 Gaussians per state. In decoding, we use a trigram LM trained with all training transcripts. On the other hand, the baseline hybrid DNN/HMMs system uses the same PLP features concatenated from 11-frame context window. The baseline DNN is composed of six hidden layers and 2,048 units per layer, which is either pre-trained by RBM or randomly initialized using Gaussian distribution. After that, The DNN hybrid system is fine-tuned by 10 epochs of frame-level cross-entropy (CE) training, and followed by 2 epochs of MMI-based sequence training. The baseline performance in Table 5 shows the CE-trained hybrid DNN/HMMs can give 34.4% relative error reduction over the discriminatively trained GMM/HMMs on Hub5e00 test set. The MMI sequence training yields word error rate of 14.0%, indicating additional 13.6% relative error reduction.

4.2.2. Dropout as pre-conditioner and shrinking hidden layer

In this section, we first apply dropout as pre-conditioner (DAP) based on the best configuration from the previous PSC task to initialize the baseline DNN for SWB. As shown in the first column in Table 6, DAP significantly improves accuracy of the baseline DNN. For example, if we run 20 epochs of DAP plus 10 epochs of BP for the cross-entropy training, word error rate is improved from 16.2% to 15.1%, about 6.8% relative error reduction. After the MMI sequence training[21], DAP yields the best accuracy of 13.4% in WER, about 4.5% improvement from baseline 14.0% in WER.

Table 5. Word error rates (WER in %) of various baseline models in Switchboard. (DNN: 6*2048)

model	method	Hub5e00
GMM-HMM	MLE	28.7
	MPE	24.7
DNN	CE (random)	16.6
	CE (pretrain)	16.2
	+ Full Seq. Training	**14.0**

Table 6. WER (in %) of DAP and sDNN in Switchboard (Hub5e00). (DNN:6*2048; sDNN1: 2*3072-2*2048-2*1024; sDNN2:2048-1792-1536-1280-1024-768; sDNN3: 2*2048-2*1024-2*512)

	DNN	sDNN1	sDNN2	sDNN3
DNN (random)	16.6	16.5	16.8	16.8
Parameters (M)	38.4	32.0	17.4	13.0
Total Time (hr)	100	90	55	45
speedup	-	x1.1	**x1.8**	**x2.2**
DNN (pretrain)	**16.2**	16.1	**16.3**	16.4
DAP epoch5+BP	15.8	15.8	15.9	16.0
DAP epoch10+BP	15.4	15.4	15.4	15.8
DAP epoch20+BP	**15.1**	15.1	**15.2**	15.6
+Full Seq.Training	**13.4**	13.4	**13.5**	13.7

Next, we consider to combine DAP with shrinking DNN (sDNN) structure for better computation efficiency. In this experiment, we evaluate three different 6-layer sDNN structures: i) sDNN1 (2*3072-2*2048-2*1024): containing 32 millions of weights (about 83% of baseline DNN); ii) sDNN2 (2048-1792-1536-1280-1024-768): containing 17.4 millions of weights (about 45.3% of baseline DNN); iii) sDNN3 (2*2048-2*1024-2*512): containing 13 millions of weights (about 34% of baseline DNN). Moreover, sDNN results in much more efficient computation for both training and decoding due to smaller number of matrix multiplications. For example, sDNN2 (or sDNN3) leads to about 1.8 (or 2.2) times faster in both training and testing. Meanwhile, as shown in Table 6, we can see that sDNN structure can yield comparable recognition performance as baseline DNN. For instance, sDNN2 accelerates computation time by 1.8 times but it maintains similar recognition performance as the baseline DNN, 16.3% in WER with pre-training and CE and 15.2% in WER with DAP and 13.5% in WER after sequence training. These numbers are pretty much the same (only 0.1% degradation) as the baseline DNN, which is about twice bigger in size and also twice slower in training and testing.

5. CONCLUSIONS

In this paper, we proposed to use dropout as pre-conditioner (DAP) for better recognition performance in LVCSR. Experimental results show that executing standard BP after a number of epochs of DAP can lead to significant performance improvement compared with traditional pre-trained DNN. Meanwhile, we also investigated a new network structure, called shrinking hidden layers, for DNNs as a new way to explore DNN sparseness for better efficiency. Experiments show that shrinking DNN structure can significantly reduce model size and computation time without losing performance.

6. REFERENCES

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks.," in *INTERSPEECH*, 2011, pp. 437–440.

[2] 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.

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*. IEEE, 2011, pp. 24–29.

[4] J. Pan, C. Liu, Z.G. Wang, Y. Hu, and H. Jiang, "Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling," in *8th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2012, pp. 301–305.

[5] T.N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A-R Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *ASRU*. IEEE, 2011, pp. 30–35.

[6] Pan Zhou, Cong Liu, Qingfeng Liu, Lirong Dai, and Hui Jiang, "A cluster-based multiple deep neural networks method for large vocabulary continuous speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6650–6654.

[7] G.E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[8] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, pp. 153, 2007.

[9] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

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

[11] Michael L Seltzer, Dong Yu, and Yongqiang Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7398–7402.

[12] Yajie Miao and Florian Metze, "Improving low-resource cd-dnnhmm using dropout and multilingual dnn training," in *Proc. Interspeech*, 2013, pp. 2237–2241.

[13] G.E. Dahl, T.N. Sainath, and G.E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *ICASSP*, 2013.

[14] J. Li, X. Wang, and B. Xu, "Understanding the dropout strategy and analyzing its effectiveness on LVCSR," in *ICASSP*, 2013.

[15] D. Yu, F. Seide, G. Li, and L. Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," in *ICASSP*. IEEE, 2012, pp. 4409–4412.

[16] 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 *ICASSP*. IEEE, 2013.

[17] J. Xue, J.Y. Li, and Y.F. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," in *INTERSPEECH*, 2013.

[18] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 9999, pp. 3371–3408, 2010.

[20] Y.B. Bao, H. Jiang, C. Liu, Y. Hu, and L.R. Dai, "Investigation on dimensionality reduction of concatenated features with deep neural network for LVCSR systems," in *11th International Conference on Signal Processing (ICSP)*, 2012, vol. 1, pp. 562–566.

[21] Pan Zhou, Lirong Dai, and Hui Jiang, "Sequence training of multiple deep neural networks for better performance and faster training speed," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014.