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

Small-footprint Deep Neural Networks with Highway Connections for Speech Recognition

Liang Lu, Steve Renals

Centre for Speech Technology Research, The University of Edinburgh, Edinburgh, UK
{liang.lu, s.renals}@ed.ac.uk

Abstract

For speech recognition, deep neural networks (DNNs) have significantly improved the recognition accuracy in most of benchmark datasets and application domains. However, compared to the conventional Gaussian mixture models, DNN-based acoustic models usually have much larger number of model parameters, making it challenging for their applications in resource constrained platforms, e.g., mobile devices. In this paper, we study the application of the recently proposed highway network to train small-footprint DNNs, which are *thinner* and *deeper*, and have significantly smaller number of model parameters compared to conventional DNNs. We investigated this approach on the AMI meeting speech transcription corpus which has around 80 hours of audio data. The highway neural networks constantly outperformed their plain DNN counterparts, and the number of model parameters can be reduced significantly without sacrificing the recognition accuracy.
Index Terms: speech recognition, highway network, small-footprint deep learning.

1. Introduction

Modern state-of-the-art speech recognition systems are based on neural network acoustic models [1, 2, 3, 4, 5]. A typical architecture is the deep neural network (DNN) [1, 2], which is a feedforward neural network with multiple hidden layers (e.g., 4 $\sim$ 9), and each layer has a large number of hidden units (e.g., 512 $\sim$ 2048). Compared to the conventional Gaussian mixture models, DNN acoustic models usually have much larger number of model parameters, which explains their large statistical modelling capacities and high recognition accuracies. However, it becomes challenging for the applications of DNN-based speech recognition systems in resource constrained scenarios. For instance, it is highly desirable that the speech recognition system can still function in wearable computing and mobile devices when the internet connection is unavailable. This requires that smaller size of acoustic models can still achieve high recognition accuracy.

There have been a number of works on small footprint DNNs for this purpose. For instance, Xue et al. [6] and Sainath et al. [7] approximate the weight matrix between two hidden layers by a product of two low-rank matrices, which may be equivalent to insert a bottleneck layer in between without the nonlinear activation. Another branch of studies are based on the *teacher-student* architecture [8, 9, 10], which is also refereed to as model compression [11] and knowledge distillation [12]. In this approach, the *teacher* may be a large-size network or an ensemble of several different models, which is used to predict the soft targets for training the *student* model that is much smaller. As discussed in [12], the soft targets provided by the teacher encode the generalisation power of the teacher, and the student model trained using these labels is observed to perform better than a small model trained in the usual way [8, 9, 10]. Recently, [13] investigated the use of low rank displacement of structured matrices (e.g., Teoplitz matrix) for small-footprint neural networks. This work is in line with the argument that neural networks with dense connections are over-parameterised, and the linear layer may be replaced by structured efficient linear layers (SELLs) [14, 15, 16].

In this paper, we investigate the *thin* and *deep* architectures for small-footprint neural network acoustic models. However, as the depth increases, training DNNs by stochastic gradient decent (SGD) becomes increasingly difficult due to the highly non-convexity of the error surface. One approach is to pre-train the neural network by unsupervised [17] or greedy layer-wise fashion [18]. However, this approach cannot circumvent the difficulty arises in the fine tuning stage. Another approach is to rely on the *teacher-student* architecture, e.g. the FitNet [10], but it requires the additional effort to train the teacher model beforehand. Our work in this paper builds on the recently proposed highway networks [19], where the *transform* gate is used to scale the output of a hidden layer and the *carry* gate is used to pass through the input directly after elementwise rescaling. Similar idea has also been studied on long short-term memory recurrent neural networks (LSTM-RNN) for speech recognition [20]. In this work, we observe that the highway connections can be successfully applied to training *thinner* and *deeper* networks, while still retraining the recognition accuracy. Our experiments were performed on the AMI meeting speech transcription corpus, which contains around 70 hours of training data. Using highway neural networks, we managed to cut down the number of model parameters by over 80% with marginal accuracy loss compared to our baseline DNN acoustic models.

2. Highway Deep Neural Network

2.1. Deep neural networks

A DNN is a feed-forward neural network with multiple hidden layers that performs cascaded layer-wise nonlinear transformations of the input. For a network with L hidden layers, the model may be represented as

$$\mathbf{h}_1 = f(\mathbf{x}, \theta_1) \tag{1}$$
$$\mathbf{h}_l = f(\mathbf{h}_{l-1}, \theta_l), \quad \text{for} \quad l = 2, \ldots, L \tag{2}$$
$$\mathbf{y} = g(\mathbf{h}_L, \varphi) \tag{3}$$

Funded by the EPSRC Programme Grant EP/I031022/1, Natural Speech Technology (NST). The NST research data collection may be accessed at http://datashare.is.ed.ac.uk/handle/10283/786. We thank Yu Zhang and Dong Yu for helpful discussions on using the CNTK toolkit.

http://dx.doi.org/10.21437/Interspeech.2016-39

where $\mathbf{x}$ is an input vector to the network; $f(\mathbf{h}_{l-1}, \theta_l)$ denotes the transformation of the input $\mathbf{h}_{l-1}$ with the parameter θ_l followed by a nonlinear activation function (e.g., `sigmoid` or `tanh`); $g(\cdot, \varphi)$ is the output function(e.g. `softmax`) which is parameterised by φ in the output layer. Given the ground truth target $\hat{\mathbf{y}}$, the network is usually trained by gradient decent to minimise a loss function $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ (e.g. cross-entropy). However, as the number of hidden layers increases, the error surface becomes increasingly non-convex, and it is more possible to find a poor local minima using gradient-based optimisation algorithms with random initialisation [21]. Furthermore, [22] showed that the variance of the back-propagated gradients may become small in the lower layers if the model parameters are not initialised properly.

2.2. Highway networks

There have been numerous studies on overcoming the difficulties in training very deep neural networks, including pre-training [17, 18], normalised initialisation [22], deeply-supervised networks [23], etc. Recently, Srivastav et al. [19] proposed the highway network and demonstrated good results to train very deep networks (e.g., up to 100 hidden layers). In the highway network, the hidden layers are augmented with two gating functions, which can be represented as

$$\begin{aligned}\mathbf{h}_l = f(\mathbf{h}_{l-1}, \theta_l) \circ T(\mathbf{h}_{l-1}, \mathbf{W}_T) \\ + \mathbf{h}_{l-1} \circ C(\mathbf{h}_{l-1}, \mathbf{W}_c)\end{aligned} \tag{4}$$

where $T(\cdot)$ is the *transform* gate that scales the original hidden activations; $C(\cdot)$ is the *carry* gate, which scales the input before passing it directly to the next hidden layer; $\circ$ denotes elementwise (Hadamad) product; The outputs of $T(\cdot)$ and $C(\cdot)$ are constrained to be $[0, 1]$, and we use sigmoid functions for both gates parameterised by $\mathbf{W}_T$ and $\mathbf{W}_c$ respectively. Unlike [19], in this work, we do not use any bias vector in the two gate functions. In [19], the carry gate is constrained to be $C(\cdot) = \mathbf{1} - T(\cdot)$, while in this work, we evaluate the generalisation ability of highway networks with and without this constraint.

Without the transform gate, i.e. $T(\cdot) = \mathbf{1}$, the highway network is similar to a network with skip connections – the main difference is that the input is firstly scaled by the carry gate. Without the carry gate, i.e. $C(\cdot) = \mathbf{0}$, the hidden layer is

$$\mathbf{h}_l = f(\mathbf{h}_{l-1}, \theta_l) \circ T(\mathbf{h}_{l-1}, \mathbf{W}_T). \tag{5}$$

At first glance, it looks similar to the dropout regularisation for neural networks [24], which may be represented as

$$\mathbf{h}_l = f(\mathbf{h}_{l-1}, \theta_l) \circ \boldsymbol{\epsilon}, \quad \epsilon_i \sim p(\epsilon_i), \tag{6}$$

where $p(\epsilon_i)$ is a Bernoulli distribution for each element in $\boldsymbol{\epsilon}$ as originally proposed in [24], while it was shown later that using a continuous distribution with well designed mean and variance works as well or better [25]. From this perspective, the transform gate may work as a regulariser, but with the key difference that $T(\cdot)$ is a deterministic function, while ϵ_i is drawn stochastically from a predefined distribution in dropout. Nevertheless, our empirical results (cf. Section 3.3) indicate that the transform gate and the carry gate can speed up the convergence rate. In addition, the highway networks also generalise better when measured in terms of recognition accuracy, which is presumably due to the regularisation effect of the two gating functions.

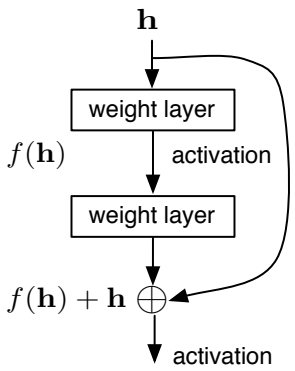

Figure 1: The building block of residual networks [26]

2.3. Small-footprint networks

The aim of this paper is to train small-footprint neural networks for resource constrained speech recognition. From Eq. (4), the highway network is not directly suitable for this purpose, because it introduces additional computational cost and model parameters in the two gating functions. The rationale is that the computational complexity and the number of model parameters for each layer in a densely connected network are in the order of $O(n^2)$, where n is the size of hidden units. Increasing the depth of the network only linearly increases the computational cost and the model size, while reducing the width can yield the quadratic reduction in the two metrics. Highway connections make it feasible to train very *thin* and *deep* networks, and therefore the overall model sizes are much smaller. To further save the model parameters, in this work, we shared the two gates for all hidden layers so that the additional number of model parameters for $T(\cdot)$ and $C(\cdot)$ is relatively small.

2.4. Comparison to residual networks

Residual network is a type of very deep network using skip connections, which has achieved state-of-the-art results in image recognition [26]. The building block for residual networks is shown in Figure 1. In fact, residual networks are similar to highway networks without the two additional gate functions, which can significantly reduce the computational cost. It also reduces the number of model parameters, albeit the reduction is marginal because the two gating functions are tied for all the hidden layers in our configuration. However, without the gating functions, training residual networks may be more difficult compared to highway networks, which will be empirically studied in the following experimental section.

3. Experiments

3.1. System setup

Our experiments were performed on the individual headset microphone (IHM) subset of the AMI meeting speech transcription corpus [27]. The amount of training data is around 80 hours, corresponding to roughly 28 million frames. This dataset is much larger than most of the datasets (e.g. MNIST, CIFAR, etc.) where other types of thin and deep networks were evaluated [10, 19]. We used 40-dimensional fMLLR adapted features vectors normalised on per-speaker level, which were then spliced by a context window of 15 frames (i.e. ± 7) for all the systems. The number of tied HMM states is 3972,

Table 1: Comparison of depth and width between plain DNNs and HDNNs. *indicates that the models were trained using the Kaldi toolkit, where the networks were initialised with restricted Boltzmann machine (RBM) based pre-training because random initialisation did not yield convergence.

System	#Layer	Dim	#Parm (M)	WER
GMM+SAT+bMMI	-	-	6.48	31.7
DNN	6	2048	30.3	26.8
DNN	6	1024	9.9	27.2
DNN	10	2048	47.1	27.7
DNN	10	1024	14.1	27.9
DNN*	10	512	4.7	28.8
DNN*	10	256	1.8	31.5
DNN*	15	1024	19.4	27.6
DNN*	15	512	6.0	29.1
DNN*	15	256	2.1	31.5
HDNN	10	2048	55.5	26.8
HDNN	10	1024	**16.2**	**26.8**
HDNN	10	512	5.2	27.2
HDNN	10	256	1.9	28.8
HDNN	10	128	0.77	32.0
HDNN	15	1024	21.5	26.8
HDNN	15	512	**6.5**	**27.1**
HDNN	15	256	2.2	28.5
HDNN	15	128	**0.85**	**31.4**

and all the DNN systems were trained with the same alignment. The results reported in this paper were obtained using the CNTK toolkit [28] with the Kaldi decoder [29], and the networks were trained using the cross-entropy (CE) criterion without pre-training unless specified otherwise. We set the momentum to be 0.9 after the 1st epoch, and we used the sigmoid activation for the hidden layers. The weights in each hidden layer were randomly initialised with a uniform distribution in the range of $[-0.5, 0.5]$ and the bias parameters were initialised to be 0 for CNTK systems. We used a trigram language model for decoding.

3.2. Depth vs. Width

Table 1 shows the word error rates (WERs) of plain DNNs and highway networks (HDNNs) with different configurations. As the number of hidden units decreases, the accuracy of plain DNNs degrade rapidly, and the accuracy loss cannot be compensated by increasing the depth of the network. We faced the difficulty to train thin and deep networks directly without RBM pre-training (the CE loss did not decrease at all after many epochs). However, with highway connections we did not have this difficulty. The HDNNs achieved consistent lower WERs compared to the plain DNN counterparts, and the margin of the gain also increases as the number of hidden units becomes smaller as shown in Figure 2. With highway connections, we can cut down the number of model parameters by around 80% with marginal accuracy loss, and with less than 1 million model parameters, the CE trained HDNN can achieve comparable or slight higher accuracy compared to a strong GMM baseline with speaker adaptive training (SAT) and bMMI-based discriminative training. The accuracy of smaller-size HDNN models may be further improved by the teacher-student style training, which will be investigated in the future.

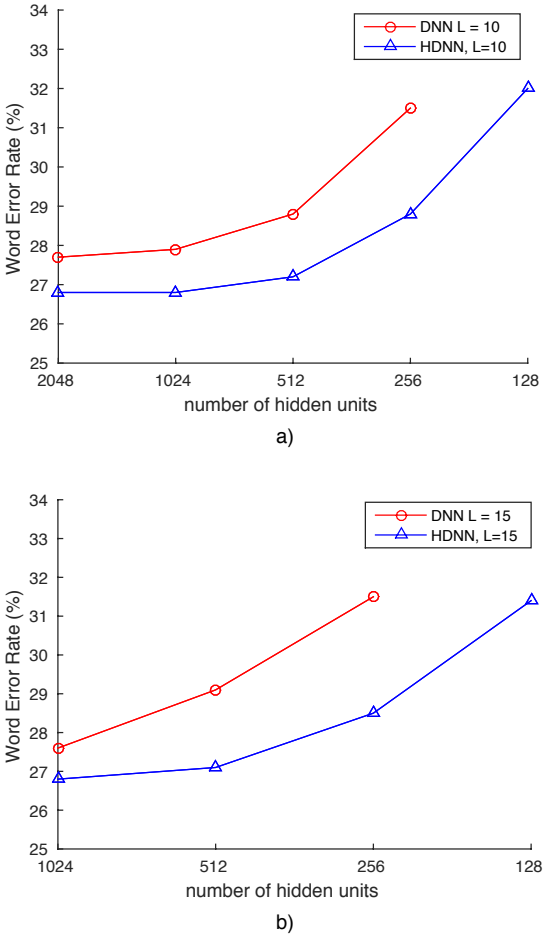

Figure 2: Comparison between plain DNNs and HDNNs with different number of hidden units. Thin and deep HDNNs achieved consistent lower WERs than their plain DNN counterparts.

3.3. Transform gate vs. Carry gate

We then evaluated the specific role of the transform and carry gate in the highway architectures. The results are shown in Table 2, where we disabled one of both of the gates. We observed that using only one of the two gates, the HDNN can still achieved lower WER compared to the plain DNN, but the best results were obtained when both of the gates were active, which indicates that the two gating functions are complementary to each other. Figure 3 shows the convergence curve of training HDNNs with and without the transform and carry gate. We observed that it converged faster when both of the gates were turned on. With only the transform gate, the convergence rate was much slower. As discussed before, the carry gate can be viewed as a particular type of skip connection, and it was more important to speed up the convergence compared to the transform gate in our experiments.

3.4. Constrained carry gate

We also evaluated using the constrained carry gate in our experiments, where $C(\cdot) = \mathbf{1} - T(\cdot)$ as studied in [19]. In this approach, the computational cost is reduced since the matrix-vector multiplication for the carry gate is not required. We eval-

Table 2: Results of highway networks with and without the transform and carry gate.

System	#Layer	Dim	Transform	Carry	WER
DNN*	10	512	×	×	28.8
HDNN	10	512	√	√	27.2
HDNN	10	512	√	×	27.6
HDNN	10	512	×	√	27.5

Table 3: Results of using constrained carry gate, where $C(\cdot) = \mathbf{1} - T(\cdot)$.

System	#Layer	Dim	Constrained	WER
DNN	10	1024	-	27.9
HDNN	10	1024	×	26.8
HDNN	10	1024	√	28.0
DNN*	10	512	-	28.8
HDNN	10	512	×	27.2
HDNN	10	512	√	27.4
DNN*	10	256	-	31.5
HDNN	10	256	×	28.8
HDNN	10	256	√	29.6

uated this configuration with 10-layer neural networks, and the results are shown in Table 3. Contrary to our expectations, with the constrained carry gate only we obtained worse results when the networks were relatively wide, while the accuracy gap was reduced when the number of hidden units was smaller. The reason may be that in the constrained setting, the transform gate $T(\cdot)$ learns the scaling function for both the input and output at the same time. As regularisation is expected to be more important for training wide and deep networks, this may not be achieved by using a single gating function. For instance, both the input and output of one hidden layer may require larger or smaller scaling weights at the same time, which is impossible in the constrained setting. In the future, we shall look into the regularisation and generalisation properties of the two gating functions more closely.

3.5. Comparison to residual networks

Finally, we compare highway networks to residual networks, and the results are given in Table 4. Our experiments showed that without the two gating functions, training the residual networks was comparably more challenging. For instance, with 10 hidden layers and using sigmoid activations, residual networks achieved higher WER compared to highway networks. However, the differences in terms of the accuracy were smaller when using ReLU (rectified linear unit) activations for residual networks, because training ReLU networks are relatively less difficult. Furthermore, we experienced difficulty to train residual networks with 15 hidden layers using sigmoid activations instead of ReLU (The CE cost did not come down after over 20 epochs), although with ReLU activations, residual networks slightly outperformed highway networks in this case. Note that, residual networks still performed better compared to the plain networks with RBM pre-training, e.g., when the depth was 10. From our experiments, we may draw the conclusion that residual networks are more powerful to train deeper networks compared to plain DNNs, particular with ReLU activation functions which reduce the optimisation difficulty. However, highway networks are more flexible with the activation functions due to the two gating functions that control the follow of information.

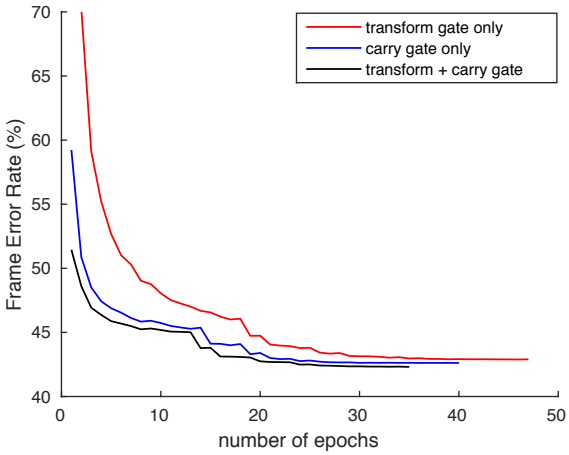

Figure 3: The convergence curve of training HDNNs with and without the transform and carry gate. The Frame Error Rates (FERs) were obtained from the validation dataset.

Table 4: Comparison to residual networks (ResNets).

System	#Layer	Dim	Activation	WER
DNN*	10	1024	Sigmoid	27.9
DNN*	10	512	Sigmoid	28.8
DNN*	10	256	Sigmoid	31.5
ResNet	10	1024	Sigmoid	27.6
ResNet	10	512	Sigmoid	27.8
ResNet	10	256	Sigmoid	29.5
HDNN	10	1024	Sigmoid	26.8
HDNN	10	512	Sigmoid	27.2
HDNN	10	256	Sigmoid	28.8
ResNet	10	1024	ReLU	27.2
ResNet	10	512	ReLU	27.3
ResNet	10	256	ReLU	28.6
ResNet	15	1024	ReLU	26.9
ResNet	15	512	ReLU	27.0
ResNet	15	256	ReLU	28.2
HDNN	15	1024	ReLU	27.1
HDNN	15	512	ReLU	27.3
HDNN	15	256	ReLU	28.7

4. Conclusions

In this paper, we investigate thin and deep neural networks for small-footprint acoustic models. Our study is build on the recently proposed highway neural network, which introduces an additional transform and carry gate for each hidden layer. Our experiments indicate that the highway connections can facilitate the information flow and mitigate the difficulty in training very deep feedforward networks. The thin and deep architecture with highway connections achieved consistently lower WERs compared to plain DNNs, and by reducing the number of hidden units, we can significantly cut down the total number of model parameters with negligible accuracy loss. We also evaluated the specific role of the transform and carry gate, and we found that the carry gate was more important to speed up the convergence in our experiment. The small-footprint highway networks may be further improved by the teacher-student style training, which will be investigated in our future work.

5. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "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.

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

[3] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer, 1994, vol. 247.

[4] N. Morgan and H. A. Bourlard, "Neural networks for statistical recognition of continuous speech," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 742–772, 1995.

[5] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, 1994.

[6] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Proc. INTERSPEECH*, 2013, pp. 2365–2369.

[7] 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 *Proc. ICASSP*. IEEE, 2013, pp. 6655–6659.

[8] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Proc. INTERSPEECH*, 2014.

[9] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. NIPS*, 2014, pp. 2654–2662.

[10] R. Adriana, B. Nicolas, K. Samira Ebrahimi, C. Antoine, G. Carlo, and B. Yoshua, "Fitnets: Hints for thin deep nets," in *Proc. ICLR*, 2015.

[11] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. ACM SIGKDD*, 2006.

[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learning and Representation Learning Workshop*, 2015.

[13] V. Sindhwani, T. N. Sainath, and S. Kumar, "Structured transforms for small-footprint deep learning," in *Proc. NIPS*, 2015.

[14] Q. Le, T. Sarlós, and A. Smola, "Fastfood-approximating kernel expansions in loglinear time," in *Proc. ICML*, 2013.

[15] Z. Yang, M. Moczulski, M. Denil, N. de Freitas, A. Smola, L. Song, and Z. Wang, "Deep fried convnets," in *Proc. ICCV*, 2015.

[16] M. Moczulski, M. Denil, J. Appleyard, and N. de Freitas, "ACDC: A Structured Efficient Linear Layer," *arXiv preprint arXiv:1511.05946*, 2015.

[17] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," in *Proc. NIPS*, vol. 19, 2007, p. 153.

[19] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. NIPS*, 2015.

[20] Y. Zhang, G. Chen, D. Yu, K. Yao, S. Khudanpur, and J. Glass, "Highway Long Short-Term Memory RNNs for Distant Speech Recognition," *arXiv preprint arXiv:1510.08983*, 2015.

[21] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in *International Conference on artificial intelligence and statistics*, 2009, pp. 153–160.

[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[23] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," *arXiv preprint arXiv:1409.5185*, 2014.

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

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

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[27] S. Renals, T. Hain, and H. Bourlard, "Recognition and understanding of meetings the AMI and AMIDA projects," in *Proc. ASRU*. IEEE, 2007, pp. 238–247.

[28] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang *et al.*, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR, Microsoft Research, Tech. Rep., 2014.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlıcek, Y. Qian, P. Schwarz, J. Silovský, G. Semmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.