Small-footprint highway deep neural networks for speech recognition

Citation for published version:
LU, L & Renals, S 2017, 'Small-footprint highway deep neural networks for speech recognition' IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 7, pp. 1502-1511. DOI: 10.1109/TASLP.2017.2698723

Digital Object Identifier (DOI):
10.1109/TASLP.2017.2698723

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
IEEE/ACM Transactions on Audio, Speech, and Language Processing

Small-footprint Highway Deep Neural Networks for Speech Recognition

Liang Lu *Member, IEEE*, Steve Renals *Fellow, IEEE*

Abstract—**State-of-the-art speech recognition systems typically employ neural network acoustic models. However, compared to Gaussian mixture models, deep neural network (DNN) based acoustic models often have many more model parameters, making it challenging for them to be deployed on resource-constrained platforms, such as mobile devices. In this paper, we study the application of the recently proposed highway deep neural network (HDNN) for training small-footprint acoustic models. HDNNs are a depth-gated feedforward neural network, which include two types of gate functions to facilitate the information flow through different layers. Our study demonstrates that HDNNs are more compact than regular DNNs for acoustic modeling, i.e., they can achieve comparable recognition accuracy with many fewer model parameters. Furthermore, HDNNs are more controllable than DNNs: the gate functions of an HDNN can control the behavior of the whole network using a very small number of model parameters. Finally, we show that HDNNs are more adaptable than DNNs. For example, simply updating the gate functions using adaptation data can result in considerable gains in accuracy. We demonstrate these aspects by experiments using the publicly available AMI corpus, which has around 80 hours of training data.**

Index Terms—**Deep learning, Highway networks, Small-footprint models, Speech recognition**

I. Introduction

Deep Learning has significantly advanced the state-of-the-art in speech recognition over the past few years [1]–[3]. Most speech recognisers now employ the neural network and hidden Markov model (NN/HMM) hybrid architecture, first investigated in the early 1990s [4], [5]. Compared to those models, current neural network acoustic models tend to be larger and deeper, made possible by faster computing such as general-purpose graphic processing units (GPGPUs). Furthermore, more complex neural architectures such as recurrent neural networks (RNNs) with long short-term memory (LSTM) units and convolutional neural networks (CNNs) have received intensive research, resulting in a range of flexible and powerful neural network architectures that have been applied to a range of tasks in speech, image and natural language processing.

Despite their success, neural network models have been criticized as lacking structure, being resistant to interpretation, and possessing limited adaptablity. Furthermore accurate neural network acoustic models reported in the research literature have tended to be much larger than conventional Gaussian mixture models, thus making it challenging to deploy them on resource constrained embedded or mobile platforms when cloud computing solutions are not appropriate (due to the unavailability of an internet connection or for privacy reasons). Recently, there has been considerable work to reduce the size of neural network acoustic models while limiting any reduction in recognition accuracy, such as the use of low-rank matrices [6], [7], teacher-student training [8]–[10], and structured linear layers [11]–[13]. Smaller footprint models may also bring advantages in requiring less training data, and in being potentially more adaptable to changing target domains, environments or speakers, owing to having fewer model parameters.

In this paper, we present a comprehensive study of small-footprint acoustic models using highway deep neural networks (HDNNs), building on our previous studies [14]–[16]. HDNNs are multi-layer networks which have shortcut connections between hidden layers [17]. Compared to regular multi-layer networks with skip connections, HDNNs are additionally equipped with two gate functions – *transform* and *carry* gates – which control and facilitate the information flow throughout the whole network. In particular, the transform gate scales the output of a hidden layer, and the carry gate is used to pass through a layer input directly after element-wise rescaling. These gate functions are central to training very deep networks [17] and to speeding up convergence [14]. We show that for speech recognition, recognition accuracy can be retained by increasing the depth of the network, while the number of hidden units in each hidden layer can be significantly reduced. As a result, HDNNs are much thinner and deeper with many fewer model parameters. Besides, in contrast to training regular multi-layer networks of the same depth and width, which typically requires careful pretraining, we demonstrate that HDNNs may be trained using standard stochastic gradient descent without any pretraining [14]. To further reduce the number of model parameters, we propose a variant of HDNN architecture by sharing the gate units across all the hidden layers. Furthermore, The authors in [17] only studied the constrained carry gate setting for HDNNs, while in this work we provide detailed comparisons of different gate functions in the context of speech recognition.

We also investigate the roles of the two gate functions in HDNNs using both cross-entropy (CE) training and sequence training, and We present a different way to investigate and understand the effect of gate units in neural networks from the point of view of regularization and adaptation. Our key observation is that the gate functions can manipulate the

Manuscript received -; revised -

Liang Lu is with Toyota Technological Institute at Chicago, and Steve Renals is with The University of Edinburgh, UK; email: llu@ttic.edu, s.renals@ed.ac.uk

The research was supported by EPSRC Programme Grant grant EP/I031022/1 *Natural Speech Technology* (NST) and the European Union under H2020 project *SUMMA*, grant agreement 688139.

behavior of all the hidden layers, and they are robust to overfitting. For instance, if we do not update the model parameters in the hidden layers and/or the softmax layer during sequence training, and only update the gate functions, then we are able to retain most of the improvement by sequence training. Moreover, the regularization term in the sequence training objective is not required when only updating the gate functions. Since the size of the gate functions are relatively small, we can achieve a considerable gain by only fine tuning these parameters for unsupervised speaker adaptation, which is a strong advantage of this model. Finally, we investigate teacher-student training, and its combination with sequence training, as well as speaker adaptation to further improve the accuracy of the small-size HDNN acoustic models. Our teacher-student training experiments also provide more results to understand this technique in the sequence training and adaptation setting.

Overall, a small-footprint HDNN acoustic model with 5 million model parameters achieved slightly better accuracy compared to a DNN system with 30 million parameters, while the HDNN model with 2 million parameters achieved only slightly lower accuracy compared to that DNN system. Finally, the recognition accuracy of a much smaller HDNN model (less than 0.8 million model parameters) can be significantly improved by teacher-student style training, narrowing the gap between this model and the much larger DNN system.

II. Highway Deep Neural Networks

A. Deep neural networks

We focus on feed-forward deep neural networks (DNNs) in this study. Although recurrent neural networks with long short-term memory units (LSTM-RNNs) and convolutional neural networks (CNNs) can obtain higher recognition accuracy with fewer model parameters compared to DNNs [18], [19], they are computationally more expensive for applications on resource constrained platforms. Moreover, their accuracy can be transferred to a DNN by teacher-student training [20]–[22].

A multi-layer network with L hidden layers is represented as

$$\boldsymbol{h}_1 = \sigma(\boldsymbol{x}, \theta_1) \tag{1}$$

$$\boldsymbol{h}_l = \sigma(\boldsymbol{h}_{l-1}, \theta_l), \quad \text{for} \quad l = 2, \ldots, L \tag{2}$$

$$\boldsymbol{y} = g(\boldsymbol{h}_L, \theta_c) \tag{3}$$

where: $\boldsymbol{x}$ is an input feature vector; $\sigma(\boldsymbol{h}_t^{(l-1)}, \theta_l)$ denotes the transformation of the input $\boldsymbol{h}_t^{(l-1)}$ with the parameter θ_l followed by a nonlinear activation function σ, e.g., `sigmoid`; $g(\cdot, \theta_c)$ is the output function that is parameterized by θ_c in the output layer, which usually uses the softmax to obtain the posterior probability of each class given the input feature. To facilitate our discussion later on, we denote θ_h=$\{\theta_1, \cdots, \theta_L\}$ as the set of neural network parameters.

Given target labels, the network is usually trained by gradient descent to minimize a loss function such as cross-entropy. However, as the number of hidden layers increases, the error surface becomes increasingly non-convex, and it becomes more likely to find a poor local minimum using gradient-based optimization algorithms with random initialization [23]. Furthermore the variance of the back-propagated gradients may become small in the lower layers if the model parameters are not initialized properly [24].

B. Highway networks

There have been a variety of training algorithms, and model architectures, proposed to enable very deep multi-layer networks including pre-training [25], [26], normalised initialisation [24], deeply-supervised networks [27], and batch normalisation [28]. Highway deep neural networks (HDNNs) [17] were proposed to enable very deep networks to be trained by augmenting the hidden layers with gate functions:

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

where: $\boldsymbol{h}_l$ denotes the hidden activations of l-th layer; $T(\cdot)$ 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; and $\circ$ denotes elementwise multiplication. The outputs of $T(\cdot)$ and $C(\cdot)$ are constrained to be within $[0, 1]$, and we use a sigmoid function for each, parameterized by $\mathbf{W}_T$ and $\mathbf{W}_c$ respectively. Following our previous work [14], we tie the parameters in the gate functions across all the hidden layers, which can significantly save model parameters. Untying the gate functions did not result in any gain in our preliminary experiments. In this work, we do not use any bias vector in the two gate functions. Since the parameters in $T(\cdot)$ and $C(\cdot)$ are layer-independent, we denote $\theta_g = (\boldsymbol{W}_T, \boldsymbol{W}_c)$, and we will look into the specific roles of these model parameters in sequence training and model adaptation experiments.

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. If the carry gate is set to zero, i.e. $C(\cdot) = \mathbf{0}$, the second term in (4) is dropped,

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

resulting in a model that is similar to dropout regularization [29], which may be written as

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

where $p(\epsilon_i)$ is a Bernoulli distribution for the i-th element in $\boldsymbol{\epsilon}$ as originally proposed in [29]; it was shown later that using a continuous distribution with well designed mean and variance works as well or better [30]. From this perspective, the transform gate may work as a regularizer, but with the key difference that $T(\cdot)$ is a deterministic function, while ϵ_i is drawn stochastically from a predefined distribution in dropout. The network in (5) is also related to LHUC (Learning Hidden Unit Contribution) adaptation for multilayer acoustic models [31], [32], which may be represented as

$$\boldsymbol{h}_l^s = a(\boldsymbol{r}_l^s) \circ \sigma(\boldsymbol{h}_{l-1}^s, \theta_l) \tag{7}$$

where: $\boldsymbol{r}_l^s$ is a speaker dependent vector for l-th hidden layer, and $\boldsymbol{h}_l^s$ is the speaker adapted hidden activations; s is the speaker index; and $a(\cdot)$ is a nonlinear function. The model in (5) can be seen as an extension of LHUC in which $\boldsymbol{r}_l^s$ is parameterized as $\boldsymbol{W}_T\boldsymbol{h}_{l-1}$. We shall investigate the update of $\boldsymbol{W}_T$ for speaker adaptation in the experimental section.

Although there are more computational steps for each hidden layer compared to regular DNNs due to the gate functions, the training speed will be improved if the size of the weight matrices are smaller. Furthermore, the matrices can be packed together as

$$\tilde{\boldsymbol{W}}_l = \left[\boldsymbol{W}_l^\top, \boldsymbol{W}_T^\top, \boldsymbol{W}_c^\top\right]^\top, \tag{8}$$

where $\boldsymbol{W}_l^\top$ is the weight matrix in the l-th layer, and we then compute $\tilde{\boldsymbol{W}}_l\boldsymbol{h}_{l-1}$. This approach, applied at the minibatch level, allows more efficient matrix computation when using GPUs.

C. Related models

Both HDNNs and LSTM-RNNs [33] employ gate functions. However, the gates in LSTMs are designed to control the information flow through time and to model along temporal dependencies; for HDNNs, the gates are used to facilitate the information flow through the depth of the model. Combinations of the two architectures have been explored recently: highway LSTMs [34] employ highway connections to train a stacked LSTM with multiple layers; recurrent highway networks [35] share gate functions to control the information flow in both time and model depth. On the other hand, the residual network (ResNet) [36] was recently proposed to train very deep networks, advancing the state-of-the-art in computer vision. ResNets are closely related to highway networks in the sense that they also rely on skip connections for training very deep networks; however, gate functions are not employed in ResNets (which can save some computational cost). Finally, adapting approaches developed for visual object recognition [37], very deep CNN architectures have been investigated for speech recognition [38].

III. Training

A. Cross-entropy training

The most common criterion used to train neural networks for classification is the cross-entropy (CE) loss function,

$$\mathcal{L}^{(CE)}(\theta) = -\sum_j \hat{y}_{jt} \log y_{jt}, \tag{9}$$

where j is the index of the hidden Markov model (HMM) state, $\boldsymbol{y}_t$ is the output of the neural network (3) at time t, and $\hat{\boldsymbol{y}}_t = \{y_{1t}, \cdots, y_{Jt}\}$ denotes the ground truth label that is a one-hot vector, where J is the number of HMM states. Note that the loss function is defined for one training example here for simplicity of notation. Supposing that $\hat{y}_{jt} = \delta_{ij}$, where δ_{ij} is the Kronecker delta function and i is the ground truth class at the time step t, the CE loss becomes

$$\mathcal{L}^{(CE)}(\theta) = -\log y_{it}. \tag{10}$$

In this case, minimizing $\mathcal{L}^{(CE)}(\theta)$ corresponds to minimizing the negative log posterior probability of the correct class, and is equal to maximizing the probability y_{it}; this will also result in minimizing the posterior probabilities of other classes since they sum to one.

B. Teacher-Student training

Instead of using the ground truth labels, the teacher-student training approach defines the loss function as

$$\mathcal{L}^{(KL)}(\theta) = -\sum_j \tilde{y}_{jt} \log y_{jt}, \tag{11}$$

where $\tilde{y}_{jt}$ is the output of the teacher model, which works as a pseudo-label. Minimizing this loss function is equivalent to minimizing the Kullback-Leibler (KL) divergence between the posterior probabilities of each class from the teacher and student models [8]. Here, $\tilde{y}_{jt}$ is no longer a one-hot vector; instead, the competing classes will have small but nonzero posterior probabilities for each training example. Hinton et al. [39] suggested that the small posterior probabilities are valuable information that encode correlations among different classes. However, their roles may be very small in the loss function as these probabilities are close to zero due to the softmax function. To address this problem, a temperature parameter, $T \in \mathbb{R}^+$, may be used to flatten the posterior distribution,

$$y_{jt} = \frac{\exp\left(z_{jt}/T\right)}{\sum_i \exp\left(z_{it}/T\right)}, \tag{12}$$

$$\boldsymbol{z}_t = \boldsymbol{W}_{L+1}\boldsymbol{h}_{Lt} + \boldsymbol{b}_{L+1}, \tag{13}$$

where $\boldsymbol{W}_{L+1}, \boldsymbol{b}_{L+1}$ are parameters in the softmax layer. Following [39], we applied the same temperature to the softmax functions in both the teacher and student networks in our experiments.[1]

A particular advantage of teacher-student training is that unlabelled data can be used easily. However, when ground truth labels are available, the two loss functions can be interpolated to give a hybrid loss parametrised by $q \in \mathbb{R}^+$

$$\widetilde{\mathcal{L}(\theta)} = \mathcal{L}^{(KL)}(\theta) + q\mathcal{L}^{(CE)}(\theta). \tag{14}$$

C. Sequence training

While the previous two loss functions are defined at the frame level, sequence training defines the loss at the sequence level, which usually yields a significant improvement in speech recognition accuracy [40]–[42]. Given a sequence of acoustic frames, $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, of length T, and a sequence of labels, $\boldsymbol{Y}$, then the loss function from the state-level minimum Bayesian risk criterion (sMBR) [43], [44] is defined as

$$\mathcal{L}^{(sMBR)}(\theta) = \frac{\sum_{\mathcal{W}\in\Phi} p(\boldsymbol{X} \mid \mathcal{W})^k P(\mathcal{W}) A(\boldsymbol{Y}, \hat{\boldsymbol{Y}})}{\sum_{\mathcal{W}\in\Phi} p(\boldsymbol{X} \mid \mathcal{W})^k P(\mathcal{W})}, \tag{15}$$

where: $A(\boldsymbol{Y}, \hat{\boldsymbol{Y}})$ measures the state-level distance between the ground truth and predicted labels; Φ denotes the hypothesis space represented by a denominator lattice; $\mathcal{W}$ is the word-level transcription; and k is the acoustic score scaling parameter. In this paper, we focus on the sMBR criterion for sequence training since it can achieve comparable or slightly better results than training using the maximum mutual information (MMI) or minimum phone error (MPE) criteria [41].

[1] Only increasing the temperature in the teacher network resulted in much higher error rates in pilot experiments.

Only applying the sequence training criterion without regularization may lead to overfitting [41], [42]. To address this problem, we interpolate the sMBR loss function with the CE loss [42], with smoothing parameter $p \in \mathbb{R}^{+}$,

$$\mathcal{L}(\theta) = \mathcal{L}^{(sMBR)}(\theta) + p\mathcal{L}^{(CE)}(\theta). \tag{16}$$

A motivation for this interpolation is that the acoustic model is usually first trained using CE, and then fine tuned using sMBR for a few iterations. However, in the case of teacher-student training for knowledge distillation, the model is first trained with the KL loss function (11). Hence, we apply the following interpolation when switching from the KL loss function (11) to the sequence-level loss function in the case of teacher-student training:

$$\widehat{\mathcal{L}(\theta)} = \mathcal{L}^{(sMBR)}(\theta) + p\mathcal{L}^{(KL)}(\theta). \tag{17}$$

Again, $p \in \mathbb{R}^{+}$ is the smoothing parameter, and we have used the same ground truth labels $\boldsymbol{Y}$ when measure the sMBR loss as in the the standard sequence training .

D. Adaptation

Adaption of deep neural networks is challenging due to the large number of unstructured model parameters and the small amount of adaptation data. However, the HDNN architecture is more structured as the parameters in the gate functions are layer-independent, and can control the behavior of all the hidden layers. This motivates the investigation of the adaptation of highway gates by only fine tuning these model parameters. Although the number of parameters in the gate functions is still large compared to the amount of per-speaker adaptation data, the size of the gate functions may be controlled by reducing the number of hidden units, but maintaining the accuracy by increasing the depth [14]. Moreover, speaker adaptation can be applied to teacher-student training to further improve the accuracy of the compact HDNN acoustic models.

IV. Experiments

A. System setup

Our experiments were performed on the individual headset microphone (IHM) subset of the AMI meeting speech transcription corpus [45], [46].[2] The amount of training data is around 80 hours, corresponding to roughly 28 million frames. We used 40-dimensional fMLLR adapted features vectors normalised at a 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, and all the DNN systems were trained using the same alignment. The results reported in this paper were obtained using the CNTK toolkit [47] with the Kaldi decoder [48], 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 initialized with a uniform distribution in the range of $[-0.5, 0.5]$ and the bias parameters were initialized to be 0 for CNTK systems. We used a trigram language model for decoding.

TABLE I
Comparison of DNN and HDNN system with CE and sMBR training. The DNN systems were built using Kaldi, where the networks were pretrained using stacked restricted Boltzmann machines. Results are shown in terms of word error rates (WERs). We use H to denote the size of hidden units, and L the number of layers. M indicates million model parameters.

			`dev`		`eval`	
ID	Model	Size	CE	sMBR	CE	sMBR
1	DNN-$H_{2048}L_6$	$30M$	26.0	24.3	26.8	24.6
2	DNN-$H_{512}L_{10}$	$4.6M$	26.8	25.1	28.0	25.6
3	DNN-$H_{256}L_{10}$	$1.7M$	28.4	26.5	30.4	27.5
4	DNN-$H_{128}L_{10}$	$0.71M$	31.5	29.3	34.1	30.8
5	HDNN-$H_{512}L_{15}$	$6.4M$	25.8	24.3	27.1	24.7
6	HDNN-$H_{512}L_{10}$	$5.1M$	26.0	24.5	27.2	24.9
7	HDNN-$H_{256}L_{15}$	$2.1M$	26.9	25.2	28.4	25.9
8	HDNN-$H_{256}L_{10}$	$1.8M$	27.2	25.2	28.6	26.0
9	HDNN-$H_{128}L_{10}$	$0.74M$	29.9	28.1	32.0	29.4

B. Baseline results

Table I shows the CE and sequence training results for baseline DNN and HDNN models of various size. The DNN systems were all trained using Kaldi with RBM pretraining (without pretraining, training thin and deep DNN models did not converge using CNTK). However, we were able to train HDNNs with random initialization without pretraining, demonstrating that the gate functions in HDNNs facilitate the information flow through the layers. For sequence training, we performed the sMBR update for 4 iterations, and set $p = 0.2$ in Eq. (16) to avoid overfitting. Table I shows that the HDNNs achieved consistently lower WERs compared to the DNNs; the margin of the gain also increases as the number of hidden units becomes smaller. As the number of hidden units decreases, the accuracy of DNNs degrades rapidly, and the accuracy loss cannot be compensated by increasing the depth of the network. The results also show that sequence training improves the recognition accuracy comparably for both DNN and HDNN systems, and the improvements are consistent for both `dev` and `eval` sets. Overall, the HDNN model with around 6 million model parameters has a similar accuracy to the regular DNN system with 30 million model parameters.

C. Transform and Carry gates

We then evaluated the specific role of the transform and carry gates in the highway architectures. The results are shown in Table II, where we disabled each of the gates in turn. We can see that using only one of the two gates, the HDNN can still achieve lower WER compared to the regular DNN baseline, but the best results are obtained when both gates are active, indicating that the two gating functions are complementary. Figure 1 shows the convergence curves of training HDNNs with and without the transform and carry gates. We observe faster convergence when both gates are active, with considerably slower convergence when using only the transform gate. This indicates that the carry gate, which controls the skip connections, is more important to the

[2] http://corpus.amiproject.org

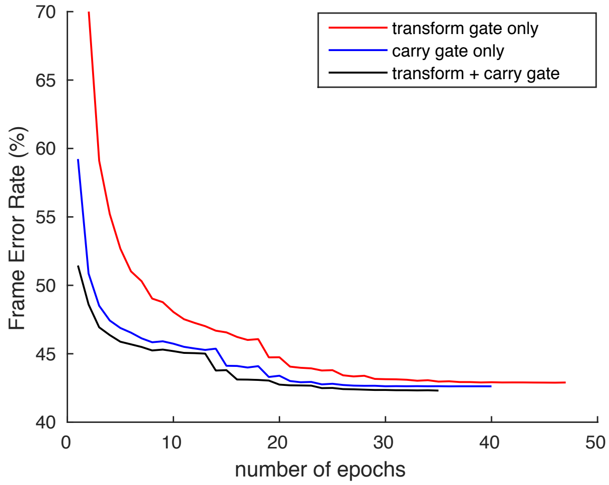


Fig. 1. Convergence curves for training HDNNs with and without the transform and the carry gate. The frame error rates (FERs) were obtained using the validation dataset.

TABLE II
RESULTS OF HIGHWAY NETWORKS WITH AND WITHOUT THE TRANSFORM AND THE CARRY GATE. THE HDNN-$H_{512}L_{10}$ WITH BOTH GATES ACTIVE CORRESPONDS TO THE CE BASELINE IN TABLE I

Model	Transform	Carry	Constrained	WER
HDNN-$H_{512}L_{10}$	$\surd$	$\surd$	$\times$	27.2
	$\surd$	$\times$	$\times$	27.6
	$\times$	$\surd$	$\times$	27.5
	$\surd$	$\surd$	$\surd$	27.4

convergence rate. We also investigated constrained gates, in which $C(\cdot) = \mathbf{1} - T(\cdot)$ [17], which reduces the computational cost since the matrix-vector multiplication for the carry gate is not required. We evaluated this configuration with 10-layer neural networks, and the results are also shown in Table II: this approach does not improve recognition accuracy in our experiments.

To look into the relative importance of the gate functions to other type of model parameters in the feature extractor and classification layer, we also performed a set of ablation experiments with sequence training, where we removed the update of different sets of model parameters (after CE training). These results are given in Table III, which shows that only updating the parameters in the gates θ_g can retain most of the improvement given by sequence training, while updating θ_g and θ_c can achieve the accuracies close to the optimum. Although θ_g only accounts for a small fraction of the total number of parameters (e.g., $\sim$ 10% for the HDNN-$H_{512}L_{10}$ system and $\sim$ 7% for the HDNN-$H_{256}L_{10}$ system), the results demonstrate that it plays an important role in manipulating the behavior of the neural network feature extractor.

Complementary to the above experiments, we then investigated the effect of the regularization term for sequence training of HDNNs (16). We performed the experiments with and without the CE regularization for two system settings, i.e.: i) update all the model parameters; ii) update only the gate functions. Our motivation was to validate if only updating the gate parameters is more resistant to overfitting. The results are given in Table IV, from which we see that by removing the CE regularization term, we achieved slightly lower WER when updating the gate functions only. However, when updating all model parameters, the regularization term was an important stabilizer for the convergence. Figure 2 shows the convergence curves for the two system settings. Overall, although the gate functions can largely control the behavior of the highway networks, they are not prone to overfitting when other model parameters are switched off.

TABLE III
RESULTS OF UPDATING OF SPECIFIC SETS OF MODEL PARAMETERS IN SEQUENCE TRAINING (AFTER CE TRAINING). θ_h DENOTES THE HIDDEN LAYER WEIGHTS, θ_g DENOTES THE GATING PARAMETERS, AND θ_c DENOTES THE PARAMETERS IN THE OUTPUT SOFTMAX LAYER. CE REGULARIZATION WAS USED IN THESE EXPERIMENTS.

	sMBR Update			WER
Model	θ_h	θ_g	θ_c	(eval)
HDNN-$H_{512}L_{10}$	$\times$	$\times$	$\times$	27.2
	$\surd$	$\surd$	$\surd$	24.9
	$\times$	$\surd$	$\surd$	25.2
	$\times$	$\surd$	$\times$	25.8
HDNN-$H_{256}L_{10}$	$\times$	$\times$	$\times$	28.6
	$\surd$	$\surd$	$\surd$	26.0
	$\times$	$\surd$	$\surd$	26.6
	$\times$	$\surd$	$\times$	27.0
HDNN-$H_{512}L_{15}$	$\times$	$\times$	$\times$	27.1
	$\surd$	$\surd$	$\surd$	24.7
	$\times$	$\surd$	$\surd$	25.2
	$\times$	$\surd$	$\times$	25.6
HDNN-$H_{256}L_{15}$	$\times$	$\times$	$\times$	28.4
	$\surd$	$\surd$	$\surd$	25.9
	$\times$	$\surd$	$\surd$	26.4
	$\times$	$\surd$	$\times$	26.6

TABLE IV
RESULTS OF SMBR TRAINING WITH AND WITHOUT REGULARIZATION.

		WER (eval)		
Model	sMBR Update	CE	$p = 0.2$	$p = 0$
HDNN-$H_{512}L_{10}$	$\{\theta_h, \theta_g, \theta_c\}$	27.2	24.9	25.0
HDNN-$H_{512}L_{10}$	θ_g	27.2	25.8	25.3
HDNN-$H_{256}L_{10}$	$\{\theta_h, \theta_g, \theta_c\}$	28.6	26.0	28.3
HDNN-$H_{256}L_{10}$	θ_g	28.6	27.0	26.8

D. Adaptation

The previous experiments show that the gate functions can largely control the behavior of a multi-layer neural network feature extractor with a relatively small number of model parameters. This observation inspired us to study speaker adaptation using the gate functions. Our first experiments explored unsupervised speaker adaptation, in which we decoded the evaluation set using the speaker-independent models, and then used the resulting pseudo-labels to fine-tune the gating parameters (θ_g) in the second pass. The evaluation set contained around 8.6 hours of audio, with 63 speakers, an average of around 8 minutes of speech per speaker, which corresponds to about 50 000 frames. This is a relatively small amount of adaptation data, given the size of θ_g (0.5 million parameters in the HDNN-$H_{512}L_{10}$ system). We set the learning rate to be 2×10^{-4} per sample, and we updated θ_g for 5 adaptation epochs.

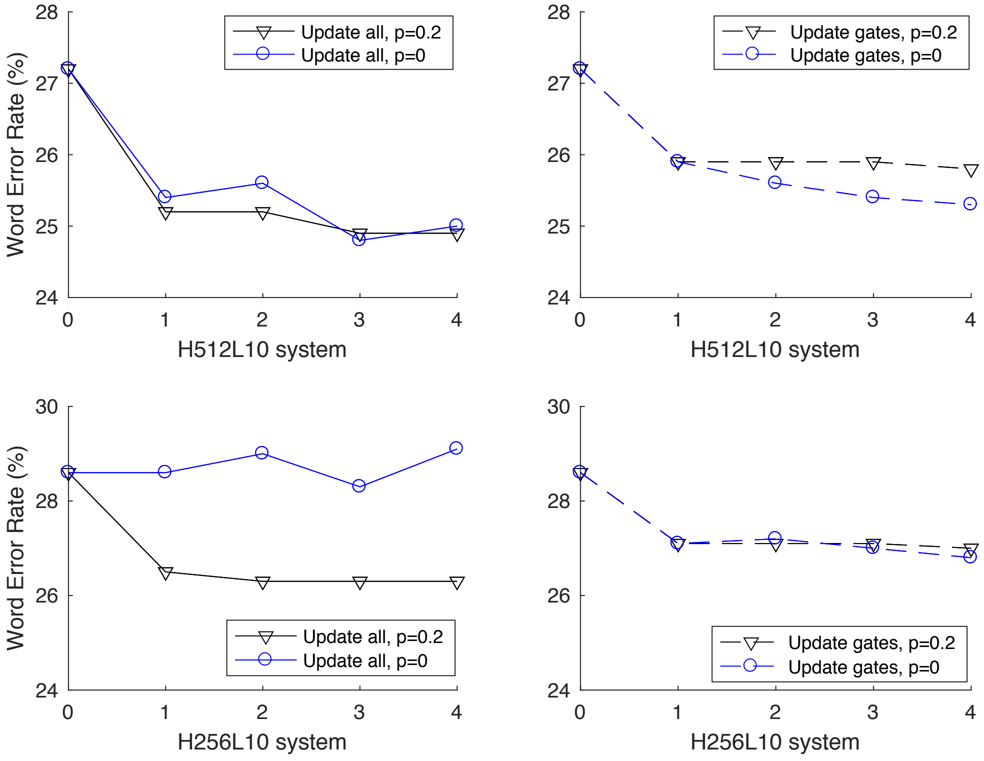


Fig. 2. Convergence curves of sMBR training with and without CE regularization (controlled by parameter p). Networks had 10 hidden layers and 256 or 512 hidden units per layer.

Table V shows the adaptation results, from which we observe a small but consistent reduction in WER for different model configurations (both CE and sMBR trained) when using fMLLR speaker adapted features. The results indicate that updating all the model parameters yields smaller improvements. With speaker adaptation and sequence training, the HDNN system with 5 million model parameters (HDNN-$H_{512}L_{10}$) works slightly better than the DNN baseline with 30 million parameters (24.1% from row 5 of Table V vs. 24.6% from row 1 of Table I), while the HDNN model with 2 million parameters (HDNN-$H_{256}L_{10}$) has only a slightly higher WER compared to the baseline (25.0% from row 6 of Table V vs. 24.6% from row 1 of Table I). In Figure 3 we show the adaptation results for a different number of iterations. We observe that the best results can be achieved after 2 or 3 adaptation iterations; further updating the gate functions θ_g does not result in overfitting. For validation we performed experiments with 10 adaptation iterations, and again we did not observe overfitting. This observation is in line with the sequence training experiments, demonstrating that the gate functions are relatively resistant to overfitting.

In order to evaluate the impact of the accuracy of the labels to this adaptation method as well as the memorization capacity of the highway gate units, we performed a set of diagnostic experiments, in which we used the oracle labels for adaptation. We obtained the oracle labels from a forced alignment using the DNN model trained with the CE criterion and word level transcriptions. We used this fixed alignment for all the adaptation experiments in order to compare the different seed models. Figure 4 shows the adaptation results with oracle labels, suggesting that an increased reduction in WER may be achieved when the supervision labels are more accurate. In the future, we shall investigate the model for domain adaptation, where the amount of adaptation data is usually relatively larger, and the ground truth labels are available.

TABLE V
RESULTS OF UNSUPERVISED SPEAKER ADAPTATION. HERE, WE ONLY UPDATED θ_g USING THE CE CRITERION, WHILE THE SPEAKER-INDEPENDENT (SI) MODELS WERE TRAINED BY EITHER CE OR SMBR. SA DENOTES SPEAKER ADAPTED MODELS.

ID	Model	Seed	Update	WER (`eval`)	
				SI	SA
1	HDNN-$H_{512}L_{10}$	CE	θ_g	27.2	26.5
2	HDNN-$H_{256}L_{10}$			28.6	27.9
3	HDNN-$H_{512}L_{15}$			27.1	26.4
4	HDNN-$H_{256}L_{15}$			28.4	27.6
5	HDNN-$H_{512}L_{10}$	sMBR		24.9	**24.1**
6	HDNN-$H_{256}L_{10}$			26.0	**25.0**
7	HDNN-$H_{512}L_{15}$			24.7	24.0
8	HDNN-$H_{256}L_{15}$			25.9	24.9
9	HDNN-$H_{128}L_{10}$			29.4	28.7
10	HDNN-$H_{512}L_{10}$		$\{\theta_h, \theta_g, \theta_c\}$	24.9	24.5
11	HDNN-$H_{256}L_{10}$			26.0	25.4
12	HDNN-$H_{128}L_{10}$			29.4	28.8

E. Teacher-Student training

After sequence training and adaptation, the HDNN with 2 million model parameters has a similar accuracy to the

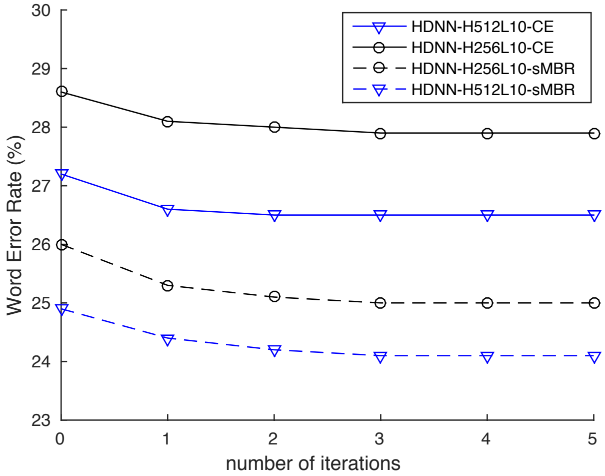


Fig. 3. Unsupervised adaptation results with different number of iterations. The speaker-independent models were trained by CE or sMBR, and we used the CE criterion for all adaptation experiments.

DNN baseline with 30 million model parameters. However, the model HDNN-$H_{128}L_{10}$ which has fewer than 0.8 million model parameters has a substantially higher WER compared to the DNN baseline (28.7% from row 9 of Table V vs. 24.6% from row 1 of Table I). We investigated if the accuracy of the small HDNN model can be further improved using teacher-student training. We first compare the teacher-student loss function (11) and the hybrid loss function (14). We used a CE trained DNN-$H_{2048}L_6$ as the teacher model, and used the HDNN-$H_{128}L_{10}$ as the student model. Figure 5 shows the convergence curves when training the model with the different loss functions, while Table VI shows the WERs. We observe that teacher-student training without the ground truth labels can achieve a significantly lower frame error rate on the cross validation set (Figure 5) which corresponds to a moderate WER reduction (Table VI: 31.3% vs. 32.0% on the `eval` set). However, using the hybrid loss function (14) does not result in further improvement, and when $q > 0$ during training convergence is slower (Figure 5). We interpret this result as indicating that the probabilities of uncorrected classes may play a lesser role, which supports the argument that they encode useful information for training the student model [39]. This hypothesis encouraged us to investigate the use of a high temperature to flatten the posterior probability distribution of the labels from the teacher model. The results are shown in Table VI; contrary to our expectation, using high temperatures results in higher WERs. In the following experiments, we fixed $q = 0$ and $T = 1$.

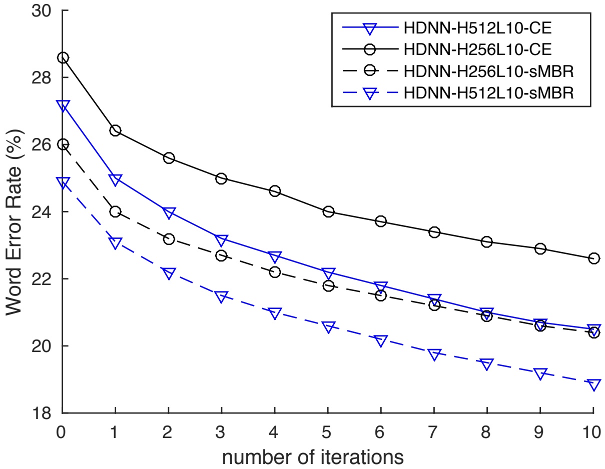


Fig. 4. Supervised adaptation results with oracle labels.

TABLE VI
RESULTS OF TEACHER-STUDENT TRAINING WITH DIFFERENT LOSS FUNCTIONS AND TEMPERATURES. q DENOTES THE INTERPOLATION PARAMETER IN EQ. (14), AND T IS THE TEMPERATURE. THE TEACHER MODELS WERE TRAINED USING THE CE CRITERION.

			WER	
Model	q	T	`eval`	`dev`
DNN-$H_{128}L_{10}$	–	–	34.1	31.5
HDNN-$H_{128}L_{10}$ baseline	–	–	32.0	29.9
HDNN-$H_{128}L_{10}$	0	1	31.3	29.3
HDNN-$H_{128}L_{10}$	0.2	1	31.4	29.5
HDNN-$H_{128}L_{10}$	0.5	1	31.3	29.4
HDNN-$H_{128}L_{10}$	1.0	1	31.3	29.4
HDNN-$H_{128}L_{10}$	0	2	32.3	29.9
HDNN-$H_{128}L_{10}$	0	3	33.0	30.6

We then improved the teacher model by sMBR sequence training, and used this model to supervise the training of the student model. We found that the sMBR-based teacher model can significantly improve the performance of the student model (similar to the results reported in [8]). In fact, the error rate is lower than that achieved by the student model trained independently with sMBR (28.8% from row 2 of Table VII vs. 29.4% from row 9 of Table I on the `eval` set). Note that, since the sequence training criterion does not maximize the frame accuracy, training the model with this criterion often reduces the frame accuracy (see Figure 6 of [49]). Interestingly, we observed the same pattern in the case of teacher-student training of HDNNs. Figure 6 shows the convergence curves of using CE and sMBR based teacher models, where we see that the student model achieves much higher frame error rate on the cross validation set when supervised by sMBR-based teacher model, although the loss function (11) is at the frame level.

We then investigated whether the accuracy of the student model can be further improved by the sequence level criterion. Here, we set the smoothing parameter $p = 0.2$ in (17) and the default learning rate to be 10^{-5} following our previous work [15]. Table VII shows sequence training results for student models supervised by both CE and sMBR-based teacher models. Surprisingly, the student model supervised by the CE-based DNN model can be significantly improved by sequence training – the WER obtained by this approach is lower compared to the model trained independently with sMBR (28.4% from row 1 of Table VII vs. 29.4% from row 9 of Table I on the `eval` set). However, this configuration did not improve the student model supervised by an sMBR-based teacher model. After inspection, we found that this was due to overfitting. We then increased the value of p to enable stronger regularization and reduced the learning rate. Lower WERs

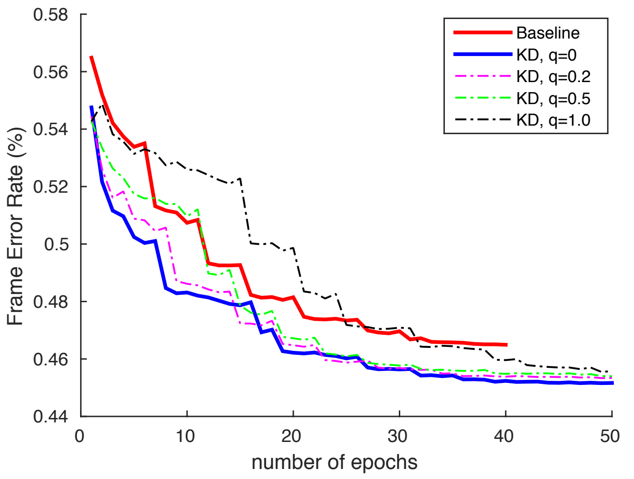


Fig. 5. Convergence curves of teacher-student training. The frame error rates were obtained from the cross validation set. The convergence slows as q increases. KD denotes teacher-student training.

TABLE VII
RESULTS OF SEQUENCE TRAINING ON THE eval SET. LR DENOTES THE LEARNING RATE. THE STUDENT MODEL IS HDNN-$H_{128}L_{10}$.

ID	Teacher model	LR	p	$\mathcal{L}^{(KL)} \to \widehat{\mathcal{L}(\theta)}$
1	DNN-$H_{2048}L_6$-CE	1×10^{-5}	0.2	31.3 → 28.4
2	DNN-$H_{2048}L_6$-sMBR	1×10^{-5}	0.2	28.8 → 28.9
3	DNN-$H_{2048}L_6$-sMBR	1×10^{-5}	0.5	28.8 → 28.0
4	DNN-$H_{2048}L_6$-sMBR	5×10^{-6}	0.2	28.8 → 28.6
5	DNN-$H_{2048}L_6$-sMBR	5×10^{-6}	0.5	28.8 → 28.0

were obtained as the table shows; however, the improvement is less significant as the sequence level information has already been integrated into the teacher model.

F. Teacher-Student training with adaptation

We then performed similar adaptation experiments to section IV-D for HDNNs trained by the teacher-student approach. We applied the second-pass adaptation approach for the standalone HDNN model, i.e., we decoded the evaluation utterances to obtain the hard labels first, and then used these labels to adapt the model using the CE loss (10). However, when using the teacher-student loss (11) only one-pass decoding is required because the pseudo-labels for adaptation are provided by the teacher, which does not need a word level transcription. This is a particular advantage of the teacher-student training technique. However, for resource-constrained application scenarios, the student model should be adapted offline, because otherwise the teacher model needs to be accessed to generate the labels. This requires another set of unlabelled speaker-dependent data for adaptation, which is usually not expensive to collect.

Since the standard AMI corpus does not have an additional set of speaker-dependent data, we only show online adaptation results. We used the teacher-student trained model from row 1 of Table VII as the speaker-independent (SI) model because its pipeline is much simpler. The baseline system used the same network as the SI model, but it was trained independently. During adaptation, we updated the SI model using 5 iterations with a fixed learning rate of 2×10^{-4} per sample following our previous setup [15]. We also compared the CE loss (10) and the teacher-student loss (11) for adaptation (Table VIII). When using the CE loss function for both SI models, slightly better results wer obtained when updating the gates only, while updating all the model parameters gave smaller improvements, possibly due to overfitting. Interestingly, this is not the case for the teacher-student loss, where updating all the model parameters yielded lower WER. These results are also in line with the argument in [39] that the soft targets can work as a regularizer and can prevent the student model from overfitting.

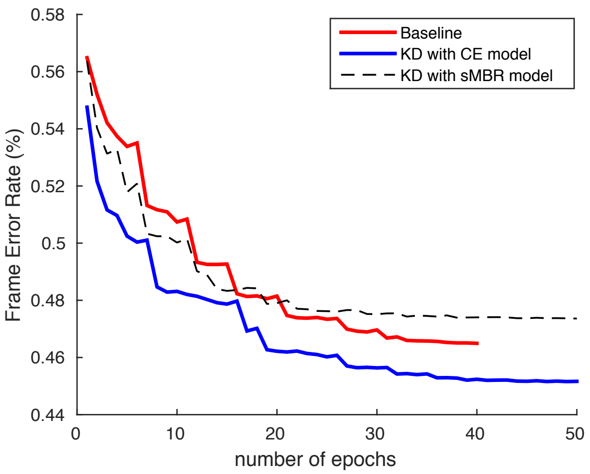


Fig. 6. Convergence curves of teacher-student training with CE or sMBR-based teacher model.

TABLE VIII
RESULTS OF UNSUPERVISED SPEAKER ADAPTATION. THE HARD LABELS ARE GROUND TRUTH LABELS, AND THE SOFT LABELS ARE PROVIDED BY THE TEACHER MODEL. HDNN-$H_{128}L_{10}$-KL DENOTES THE STUDENT MODEL.

			eval	
Model	Label	Update	SI	SA
HDNN-$H_{128}L_{10}$	Hard	$\{\theta_h, \theta_g, \theta_c\}$	29.4	28.8
HDNN-$H_{128}L_{10}$	Hard	θ_g	29.4	28.7
HDNN-$H_{128}L_{10}$-KL	Soft	$\{\theta_h, \theta_g, \theta_c\}$	28.4	27.5
HDNN-$H_{128}L_{10}$-KL	Soft	θ_g	28.4	27.8
HDNN-$H_{128}L_{10}$-KL	Hard	$\{\theta_h, \theta_g, \theta_c\}$	28.4	27.7
HDNN-$H_{128}L_{10}$-KL	Hard	θ_g	28.4	27.1

G. Summary

We summarize our key results in Table IX. Overall, the HDNN acoustic model can slightly outperform the sequence trained baseline using around 5 million model parameters after adapting the gate functions; using fewer than 2 million model parameters it performed slightly worse. If fewer than 0.8 million parameters are used, then the gap is much larger compared to the DNN baseline. With adaptation and teacher-student training, we can close the gap by around 50%, with difference in WER falling from roughly 5% absolute to 2.5% absolute.

V. CONCLUSIONS

Highway deep neural networks are structured, depth-gated feedforward neural networks. In this paper, we studied se-

TABLE IX
SUMMARY OF OUR RESULTS.

Model	Size	WER
DNN-$H_{2048}L_6$ CE baseline	$30M$	26.8
+sMBR training	$30M$	**24.6**
HDNN-$H_{512}L_{10}$ CE baseline	$5.1M$	27.2
+sMBR training	$5.1M$	24.9
+ adaptation	$5.1M$	**24.0**
HDNN-$H_{256}L_{10}$ CE baseline	$1.8M$	28.6
+sMBR training	$1.8M$	26.0
+ adaptation	$1.8M$	**25.0**
HDNN-$H_{128}L_{10}$ CE baseline	$0.74M$	32.0
+sMBR training	$0.74M$	29.4
+ teacher-student training	$0.74M$	28.4
+ adaptation	$0.74M$	**27.1**

quence training and adaptation of these networks for acoustic modeling. In particular, we investigated the roles of the parameters in the hidden layers, gate functions and classification layer in the case of sequence training. We show that the gate functions, which only account for a small fraction of the whole parameter set, are able to control the information flow and adjust the behavior of the neural network feature extractors. We demonstrate this in both sequence training and adaptation experiments, in which considerable improvements were achieved by only updating the gate functions. Using these techniques, we obtained comparable or slightly lower WERs with much smaller acoustic models compared to a strong baseline set by a conventional DNN acoustic model with sequence training. Since the number of model parameters is still relatively large compared to the amount of data typically used for speaker adaptation, this adaptation technique may be more applicable to domain adaptation, where the expected amount of adaptation data is larger.

Furthermore, we also investigated teacher-student training for small-footprint acoustic models using HDNNs. We observed that the accuracy of the student acoustic model could be improved under the supervision of a high accuracy teacher model, even without additional unsupervised data. In particular, the student model supervised by an sMBR-based teacher model achieved lower WER compared to the model trained independently using the sMBR-based sequence training approach. Unsupervised speaker adaptation further improved the recognition accuracy by around 5% relative for a model with fewer then 0.8 million model parameters. However, we did not obtain improvements either using a hybrid loss function which interpolates the CE and teacher-student loss functions, or using a higher temperature to smooth the pseudo-labels. In the future, we shall evaluate this model in low resource conditions where the amount of training data is much smaller.

VI. ACKNOWLEDGEMENT

We thank the NVIDIA Corporation for the donation of a Titan X GPU, and the anonymous reviewers for insightful comments and suggestions that helped to improve the quality of this paper.

REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and Brain 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] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks.," in *Interspeech*, 2011, pp. 437–440.

[3] George Saon, Tom Sercu, Steven Rennie, and Hong-Kwang J. Kuo, "The IBM 2016 English Conversational Telephone Speech Recognition System," in *Proc. INTERSPEECH*, 2016.

[4] Herve A Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247, Springer, 1994.

[5] Steve Renals, Nelson Morgan, Hervé Bourlard, Michael Cohen, and Horacio Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, 1994.

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

[7] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. ICASSP*. IEEE, 2013, pp. 6655–6659.

[8] Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifan Gong, "Learning small-size DNN with output-distribution-based criteria," in *Proc. INTERSPEECH*, 2014.

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

[10] Romero Adriana, Ballas Nicolas, Kahou Samira Ebrahimi, Chassang Antoine, Gatta Carlo, and Bengio Yoshua, "Fitnets: Hints for thin deep nets," in *Proc. ICLR*, 2015.

[11] Quoc Le, Tamás Sarlós, and Alex Smola, "Fastfood-approximating kernel expansions in loglinear time," in *Proc. ICML*, 2013.

[12] Vikas Sindhwani, Tara N Sainath, and Sanjiv Kumar, "Structured transforms for small-footprint deep learning," in *Proc. NIPS*, 2015.

[13] Marcin Moczulski, Misha Denil, Jeremy Appleyard, and Nando de Freitas, "ACDC: A Structured Efficient Linear Layer," in *Proc. ICLR*, 2016.

[14] Liang Lu and Steve Renals, "Small-footprint deep neural networks with highway connections for speech recognition," in *Proc. INTERSPEECH*, 2016.

[15] Liang Lu, "Sequence training and adaptation of highway deep neural networks," in *Proc. SLT*, 2016.

[16] Liang Lu, Michelle Guo, and Steve Renals, "Knowledge distillation for small-footprint highway networks," in *Proc. ICASSP*, 2017.

[17] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber, "Training very deep networks," in *Proc. NIPS*, 2015.

[18] Hasim Sak, Andrew W Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," in *Proc. INTERSPEECH*, 2014.

[19] Ossama Abdel-Hamid, Abdel-Rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 10, pp. 1533–1545, 2014.

[20] William Chan, Nan Rosemary Ke, and Ian Lane, "Transferring knowledge from a RNN to a DNN," in *Proc. INTERSPEECH*, 2015.

[21] Jeremy HM Wong and Mark JF Gales, "Sequence student-teacher training of deep neural networks," in *Proc. INTERSPEECH*. 2016, International Speech Communication Association.

[22] Yevgen Chebotar and Austin Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," in *Proc. INTERSPEECH*, 2016, pp. 3439–3443.

[23] Dumitru Erhan, Pierre-Antoine Manzagol, Yoshua Bengio, Samy Bengio, and Pascal 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.

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

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

[26] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al., "Greedy layer-wise training of deep networks," in *Proc. NIPS*, 2007, vol. 19, p. 153.

[27] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu, "Deeply-supervised nets," *arXiv preprint arXiv:1409.5185*, 2014.

[28] S Ioffe and C Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[29] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan 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.

[31] Pawel Swietojanski and Steve Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. SLT*. IEEE, 2014, pp. 171–176.

[32] P Swietojanski, J Li, and S Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[33] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yao, Sanjeev Khudanpur, and James Glass, "Highway Long Short-Term Memory RNNs for Distant Speech Recognition," *Proc. ICASSP*, 2015.

[35] Julian Georg. Zilly, Rupesh Kumar Srivastava, Koutnik Jan, and Jürgen Schmidhuber, "Recurrent highway networks," *arXiv preprint arXiv:1607.03474*, 2016.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[37] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.

[38] Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263–2276, 2016.

[39] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learning and Representation Learning Workshop*, 2015.

[40] Brian Kingsbury, Tara N Sainath, and Hagen Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. INTERSPEECH*, 2012.

[41] K Veselý, A Ghoshal, L Burget, and D Povey, "Sequence-discriminative training of deep neural networks," in *Proc. INTERSPEECH*, 2013.

[42] Hang Su, Gang Li, Dong Yu, and Frank Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. ICASSP*. IEEE, 2013, pp. 6664–6668.

[43] Matthew Gibson and Thomas Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition.," in *Proc. INTERSPEECH*. Citeseer, 2006.

[44] Brian Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. ICASSP*. IEEE, 2009, pp. 3761–3764.

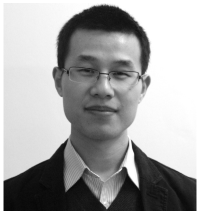
[45] Steve Renals, Thomas Hain, and Hervé Bourlard, "Recognition and understanding of meetings the AMI and AMIDA projects," in *Proc. ASRU*. IEEE, 2007, pp. 238–247.

[46] S Renals and P Swietojanski, "Distant speech recognition experiments using the AMI Corpus," in *New Era for Robust Speech Recognition – Exploting Deep Learning*, S Watanabe, M Delcroix, F Metze, and JR Hershey, Eds. Springer, 2016.

[47] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, et al., "An introduction to computational networks and the computational network toolkit," Tech. Rep., Tech. Rep. MSR, Microsoft Research, 2014.

[48] 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.

[49] Georg Heigold, Erik McDermott, Vincent Vanhoucke, Andrew Senior, and Michiel Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *Proc. ICASSP*. IEEE, 2014, pp. 5587–5591.



Liang Lu a Research Assistant Professor at the Toyota Technological Institute at Chicago. He received his Ph.D. degree from the University of Edinburgh in 2013, where he then worked as a Postdoctoral Research Associate until 2016 before moving to Chicago. He has a broad research interest in the field of speech and language processing. He received the best paper award for his work on the low-resource pronunciation modeling at the 2013 IEEE ASRU workshop.

Steve Renals (M'91 — SM'11 — F'14) received the B.Sc. degree from the University of Sheffield, Sheffield, U.K., and the M.Sc. and Ph.D. degrees from the University of Edinburgh. He is Professor of Speech Technology at the University of Edinburgh, having previously had positions at ICSI Berkeley, the University of Cambridge, and the University of Sheffield. He has research interests in speech and language processing. He is a fellow of ISCA.