Title	Wise Teachers Train Better DNN Acoustic Models
Authors	Ryan Price, Kenichi Iso, Koichi Shinoda
Citation	EURASIP Journal on Audio Speech and Music Processing, , 10, pp. 1-19
Pub. date	2016, 4
Creative Commons	See next page.

License

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Wise teachers train better DNN acoustic models

Ryan Price[1]*, Ken-ichi Iso[2] and Koichi Shinoda[1]

Abstract

Automatic speech recognition is becoming more ubiquitous as recognition performance improves, capable devices increase in number, and areas of new application open up. Neural network acoustic models that can utilize speaker-adaptive features, have deep and wide layers, or more computationally expensive architectures, for example, often obtain best recognition accuracy but may not be suitable for the given budget of computational and storage resources or latency required by the deployed system. We explore a straightforward training approach which takes advantage of highly accurate but expensive-to-evaluate neural network acoustic models by using their outputs to relabel training examples for easier-to-deploy models. Experiments on a large vocabulary continuous speech recognition task offer relative reductions in word error rate of up to 16.7 % over training with the hard aligned labels by effectively making use of large amounts of additional untranscribed data. Somewhat remarkably, the approach works well even when only two output classes are present. Experiments on a voice activity detection task give relative reductions in equal error rate of up to 11.5 % when using a convolutional neural network to relabel training examples for a feedforward neural network. An investigation into the hidden layer weight matrices finds that soft target-trained networks tend to produce weight matrices having fuller rank and slower decay in singular values than their hard target-trained counterparts, suggesting that more of the network's capacity is utilized for learning additional information giving better accuracy.

Keywords: Soft targets, Deep neural networks, Online speech recognition, Speaker-adaptive features, Model compression

1 Introduction

Over the last several years, neural network (NN) acoustic models have become an essential component in many state-of-the-art automatic speech recognition (ASR) systems, with the most accurate NN acoustic models being considerably complex in size and architecture. Deep neural networks (DNNs) have offered big gains on many ASR benchmarks [1]. Key to the success of DNN is attributed in large part to modeling tied context-dependent (CD) states in the output layer, large windows of acoustic input, and many wide layers of nonlinear units. In addition, techniques developed for Gaussian mixture model (GMM)-based large vocabulary continuous speech recognition (LVCSR), such as discriminative training, are now also incorporated in DNN training in the form of sequential training methods [2, 3], improving accuracy further.

In addition to sequential training, use of speaker-adaptive (SA) features originally designed for GMM-based systems is becoming widely adopted in DNN-HMM hybrid systems with good results reported on benchmark transcription tasks [3–6]. Feature-space maximum likelihood linear regression (fMLLR) [7] is a common choice for reducing interspeaker variability. Beyond the traditional GMM-based LVCSR techniques, appending identity vectors (*i*-vectors), which capture speaker and channel information in a low-dimensional representation, to each frame of DNN input has become a promising approach for making DNN more speaker invariant [8, 9]. Improving DNN noise robustness was explored in [10] by augmenting DNN input vectors with an estimate of the noise present in the signal. Naturally, these approaches require the speaker transforms or augmented features to be present at both training and test time.

*Correspondence: rypric00@gmail.com
[1] Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan
Full list of author information is available at the end of the article

Complexity of the NN being used in speech processing applications is increasing as well. Graphics processing units and large distributed systems have made training DNN with tens of millions of parameters and thousands of output units routine for many ASR tasks. Alternative, more complex architectures such as convolutional neural networks (CNN) offer promising gains over DNN, especially in joint architectures with DNN trained on SA features [11]. Furthermore, ensembles of various deep architectures are also being explored [12]. Larger models and some newer architectures offer improved accuracy but are also more difficult to deploy for many applications due to increased computational cost at recognition time.

Many of these improvements are, for one reason or another, difficult to fully take advantage of for online recognition tasks requiring low latency with computation and storage limitations. Unless applied in an incremental adaptation approach, SA features require multiple decoding passes or processing of an entire utterance or utterances to estimate statistics. In some cases, speaker identity may be unknown or unreliable at test time. Very large DNN, or alternative architectures such as CNN, may simply be too computationally or storage intensive for some desired online applications, massive batch processing tasks where real-time factor is very important, or embedded systems without undergoing extensive optimization.

With this in mind, we explore a strategy for training easier-to-deploy DNN acoustic models which begins by training the most accurate NN for the task without regards to the constraints for the deployed system. Such a system may be trained with additional frames of future context which reduce the word error rate (WER) but would increase the latency of an online system, for example. Various input features, model sizes, and architectures may be undesirable or infeasible for a given deployed system but can play an important role in training a NN that does meet the necessary constraints. By relabeling training examples with the outputs from an expensive model, an easier-to-deploy model can be trained to approximate the function learned by the expensive model. In other words, the prediction outputs by trained, expensive-to-evaluate networks are assigned as soft targets for an easy-to-deploy network to learn to predict, rather than training with the original 0/1 labels. The expensive networks are not needed at recognition time after training the easy-to-deploy network. Because the ground truth labels obtained from forced alignment of the reference transcription are not required, this lets us use large amounts of untranscribed data to increasingly better approximate the function learned by the expensive-to-evaluate NN.

The concept of training a NN using the outputs from another, more accurate NN, or ensemble of NNs has been studied using various approaches previously [13–16]. We refer to the approach taken in this paper as "soft target training" and feel it offers many opportunities for improving performance of NNs in situations where low computational complexity and low memory footprint are important, and untranscribed data can be obtained at low cost. The soft target training approach of this paper shares some similarities with previous works which are described in detail in Section 2.4.

It is interesting to note that the speech recognition community actually began work on training NNs with soft posterior probabilities instead of hard targets generated from forced alignment a long time ago [17–20]. While this paper, along with other recent ones, focuses on using soft targets from a more accurate "teacher" network to train an easier-to-deploy DNN, these older works were concerned with improving recognition accuracy of a single NN used during both training and deployment by generating training targets which more smoothly represented the transitions between HMM states. Training was an iterative process alternating between rounds of updating NN parameters using backpropagation and reestimating the soft targets used for training by performing a soft alignment of the data and reference transcription using the forward-backward algorithm given the current NN. Like several more recent approaches, a cross-entropy-based criterion was optimized when training the network with the soft targets generated by the forward-backward algorithm, though [20] optimized a mean squared error-based criterion instead.

We evaluate our approach on a LVCSR task using a 110-h subset of the Switchboard-1 corpus, treating the remainder of the 300 h of data as untranscribed in order to demonstrate the effectiveness of having additional untranscribed data. Training a full-sized DNN (30.4 million parameters) with log-mel filterbank inputs using the outputs from a DNN of the same size which has been sequence-trained with fMLLR inputs, yields 8.2 and 6.2 % relative reductions in WER compared to a sequence-trained DNN using hard aligned targets and log-mel filterbank inputs. When reducing the size of the DNN to be much smaller (3.4 million parameters), 4.6 and 3.3 % relative reductions in WER are obtained compared to a small, sequence-trained DNN with hard aligned targets and log-mel filterbank inputs. When the set of untranscribed data is augmented with data from the Fisher corpus, relative reductions in WER are increased to 8.2 and 11.2 % for the small DNN. Viewed in terms of parameter reduction, this is a 88.8 % reduction in parameters while actually achieving a 1.6 and 5.3 % relative reduction in WER compared to a much larger DNN sequence-trained with log-mel filterbank inputs and hard aligned targets. We also demonstrate the usefulness of the approach for classifying speech and non-speech frames in a voice activity detection (VAD) task for mobile voice search. Training a multi-layer perceptron (MLP) with soft targets from a CNN gives up to 11.5 % relative reduction in equal error rate compared to training with hard targets from Viterbi alignment.

The rest of this paper is organized as follows. In the next section, we describe the basic algorithm for training with soft targets. LVCSR acoustic modeling experiments for full-sized DNN with SA and speaker-independent (SI) features are studied in Section 3, followed by experiments where the network size is reduced in Section 4. We demonstrate the suitability of the approach in the context of VAD with soft targets from a CNN in Section 5. Following that, we discuss why training with soft targets from a highly accurate NN can be more effective than with hard aligned targets. The paper finishes with conclusions and future work in Section 7.

2 Soft target training

2.1 Training a NN to approximate the function learned by a more accurate NN

Soft target training utilizes outputs from a more accurate *teacher* model to train a *student* model which satisfies some criteria for deployment (e.g., runs in real time on a given device). Teacher and student models are NNs in this work but that need not be the case in general. The teacher may be a single network or ensemble of networks from which the outputs are averaged. For simplicity, we always refer to the teacher as a single NN when describing the approach. Motivations for this type of training have typically been (a) approximating a single NN with a smaller NN and (b) condensing an ensemble of models down to a single NN having approximately the same or greater number of parameters as a single NN in the ensemble but much less than the combination of all models in the ensemble. In addition to these cases, we allow for (c) a student to have a "cheaper-to-obtain" input representation than the teacher, which is another way of making a deployed model faster to evaluate.

The soft target training approach used in this paper proceeds as follows. The teacher network is first trained to minimize the cross-entropy cost function using labeled data set A with the provided 0/1 labels ("hard targets"). For training the student network, a possibly much larger data set B is used. Examples in B need not be labeled. The fully trained teacher network is then used to provide soft targets for training examples from data set B by forward propagating each input in order to obtain the outputs which will serve as labels for the corresponding input. When the output layer has a softmax activation function, these labels will be a vector of class conditional probabilities summing to 1. The probabilities over all classes predicted by the teacher are used, rather than the just the class having the maximum prediction, hence the term soft targets as opposed to hard targets. As we discuss in Section 6, these predictions from the teacher can actually be much more informative than the original 0/1 class labels because they impart the relationship between different classes and inputs that has been internalized by the teacher. Depending on the teacher, training examples from B may be further transformed or augmented in some way to obtain the appropriate input feature representation for the teacher trained with data set A, for example, by using unsupervised fMLLR or appending *i*-vectors. The training examples from B are then input to the student which is trained using the corresponding soft targets obtained from the teacher.

A pair of teacher and student networks is illustrated in Fig. 1 with softmax outputs $\mathbf{y}_{\mathcal{T}}$ and $\mathbf{y}_{\mathcal{S}}$ at frame n for teacher and student, respectively. Inputs are given by $\mathbf{x}_{\mathcal{T}}(n)$ and $\mathbf{x}_{\mathcal{S}}(n)$ for teacher and student, respectively, and may have different representations but are time synchronous. The pre-activation output for the student is given by $\mathbf{a}_{\mathcal{S}}$, and C is the cross-entropy cost function.

The next section begins with a description of the standard training approach for DNN acoustic models and continues the description of soft target training in greater detail for the DNN acoustic modeling case.

2.2 Soft target training for DNN acoustic models

DNN acoustic models take multiple frames of acoustic input and produce HMM state posterior probabilities for a desired input frame as their output. Input $\mathbf{x}$ is processed through many layers of nonlinear transformation before posterior probabilities $p(s|\mathbf{x})$ for CD states s are obtained from DNN output vector $\mathbf{y}$ by applying the

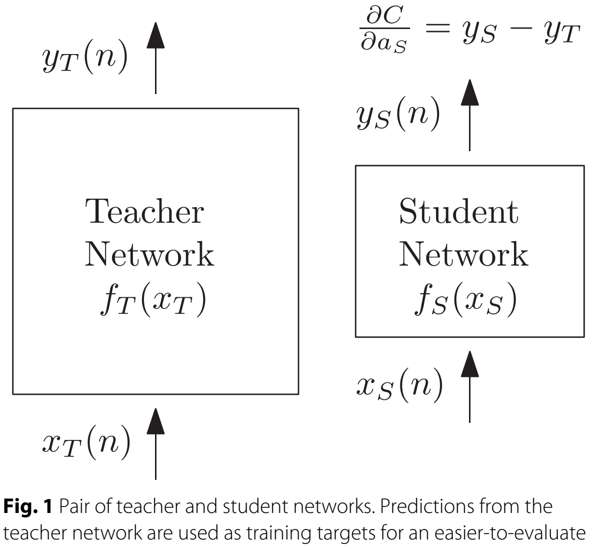
$y_T(n)$
$\frac{\partial C}{\partial a_S} = y_S - y_T$
$y_S(n)$
Teacher Network $f_T(x_T)$
Student Network $f_S(x_S)$
$x_S(n)$
$x_T(n)$

Fig. 1 Pair of teacher and student networks. Predictions from the teacher network are used as training targets for an easier-to-evaluate student network using a large amount of unlabeled data. Teacher and student networks may have different input representations, sizes, and architectures

softmax function element-wise to activations $\mathbf{a}$ at output layer L

$$\mathbf{y}^{(L)} = \frac{1}{Z}\exp(\mathbf{a}^{(L)}), \tag{1}$$

where $\mathbf{a}^{(L)} = (\mathbf{W}^{(L)})^T\mathbf{y}^{(L-1)} + \mathbf{b}^{(L)}$ with weight matrix $\mathbf{W}^{(L)}$, output of the previous layer $\mathbf{y}^{(L-1)}$, bias vector $\mathbf{b}^{(L)}$, and softmax normalization term Z. The CD state posterior probabilities estimated by the DNN with (1) are divided by the CD state prior probabilities estimated from the training data to obtain the CD state emission likelihoods used by the hybrid ASR system.

DNN acoustic model training typically begins with frame-wise minimization of cross-entropy criterion. The general form is given by

$$-\sum_{n}^{N}\sum_{i=1}^{S}\hat{P}(s_i|\mathbf{x}(n))\log P(s_i|\mathbf{x}(n)), \tag{2}$$

where $\hat{P}$ is the "true" posterior distribution, P is the posterior distribution output by the DNN being trained, S is the number of HMM states modeled in the output layer, and N is the number of frames in the training set. When CD state posterior probabilities for training are determined using a hard alignment of the reference transcription, (2) simplifies to the familiar form of negative log likelihood because $\hat{P}(s_t|\mathbf{x}(n)) = 1$ for target state s_t and is zero for all other states at frame n. Training is carried out using the standard backpropagation algorithm with (2) as the loss function and the gradient with respect to the pre-softmax activations in the output layer for a given frame is

$$\frac{\partial C}{\partial \mathbf{a}^{(L)}} = \mathbf{y}^{(L)} - \mathbf{t}, \tag{3}$$

where $\mathbf{t}$ is a one-hot target vector indicating the target state for the given frame. Frame-discriminative cross-entropy training is often followed by some form of sequence-discriminative training [2, 3].

Soft target training also follows from the general form of cross-entropy in (2), with $P_{\mathcal{T}}$ denoting the output distribution of the teacher and $P_{\mathcal{S}}$ the output distribution of the student DNN being trained

$$-\sum_{n}^{N}\sum_{i=1}^{S}P_{\mathcal{T}}(s_i|\mathbf{x}_{\mathcal{T}}(n))\log P_{\mathcal{S}}(s_i|\mathbf{x}_{\mathcal{S}}(n)), \tag{4}$$

and the outputs from the teacher network, $\mathbf{y}_{\mathcal{T}}$, replace the one-hot target vector, $\mathbf{t}$, in (3).

This makes soft target training very easy to implement requiring little modification to existing cross-entropy minimization code. Furthermore, hyperparameter values that work well for cross-entropy training with hard targets also work well for soft targets, in our experience. One other slight modification is that early stopping is performed by monitoring the cross-entropy cost with soft targets from the teacher on a development set, rather than the frame classification rate or cross-entropy cost with hard aligned targets.

2.3 Use of untranscribed data

In practice, having a sufficiently large amount of data seems key for obtaining an accurate student network when the function learned by the teacher is complex and the student is constrained. Since outputs from the teacher network are used to label training examples, the data used to train the student model does not need to be labeled. While techniques like generating synthetic data and randomly perturbing examples in the training set are practical in some circumstances, one of the most attractive ways to increase the size of the training set for soft target training is to simply add real, unlabeled data that matches well with the distribution of the training and test data. Fortunately, large amounts of untranscribed speech data often exist for many interesting online applications and can be used during the training of the student network in order to obtain a better approximation of the function learned by the teacher network. Furthermore, many feature space adaptation methods beneficial for DNN acoustic modeling such as fMLLR, *i*-vector, and noise estimates can be estimated in an unsupervised fashion without requiring a transcription.

It is worth noting that the use of unlabeled data in soft target training is different from that of semi-supervised learning. While specific semi-supervised approaches may vary, self-training methods are common and can be described as a process in which an initial DNN is trained on some labeled data and used to classify unlabeled data, out of which the examples resulting in the most confident predictions are selected and added to the pool of supervised training data. In the case of ASR, recognition hypotheses for the untranscribed data are obtained by decoding with an initial DNN trained with transcribed data and are used as ground-truth transcriptions for subsequent training of the initial DNN. See [21] and [22] for recent examples. Unlabeled data is used in soft target training to aid the student network in learning to better approximate the outputs of a fully trained teacher network. The unlabeled data used for soft target training is not selected using any confidence measure since the aim is simply to train the student to make the same predictions as the teacher. In contrast, semi-supervised learning aims to outperform training with only transcribed data. Semi-supervised learning could potentially play a role in the soft target training process as well. Untranscribed data could potentially be incorporated while the teacher network is being trained using the self-training method described above in order to obtain an even more accurate teacher. The untranscribed data would then be used again when training the student network with soft targets from the fully trained teacher network following the approach described in Section 2.1.

2.4 Previous work

The idea of using a fast, compact model to approximate the function learned by a slower, larger, more accurate model is not a recent one. Zeng and Martinez [13] proposed training a single MLP to approximate an ensemble of 10 small MLPs for several classification tasks drawn from the UCI data repository. In their work, the number of units in the hidden layer was determined for each MLP and task based on the error rate observed on validation data with the result that the hidden layer size of the approximating MLP was always larger (often double) than that of a MLP trained with the original labels. In contrast to earlier works, they used the averaged vector of class probability output by the ensemble to relabel training data sampled from the original training data set, rather than the single class having the maximum probability.

Menke et al. [23] trained a small MLP to accurately approximate a single larger MLP, as opposed to an ensemble. The number of hidden units was reduced from 200 to 100 when predicting phoneme classes for digit recognition on the TI digit corpus with only a slight increase in WER. While previous works had mostly relied on creating additional synthetic data in order to increase the size of the training set for the approximating network, Menke et al. [23] utilized additional unlabeled data from the same distribution. Later, Buciluă et al. [14] also studied training a single MLP to approximate an ensemble, calling the approach "model compression". Several methods of creating artificial data for training the approximating model were proposed and compared on different classification tasks. They found that a diverse ensemble of models could be accurately approximated by a MLP having drastically fewer parameters on the classification tasks studied.

Despite having been around for quite some time, this approach has not yet been widely taken advantage of in the ASR community, even with a boom in the use of large, increasingly complex, deep architectures and growing opportunity for online ASR applications. A few newly published papers have begun to explore the topic. In a recent work, Li et al. [15] trained a small DNN by minimizing the KL divergence between the output distribution of the small DNN and large DNN. As noted in their paper, this is equivalent to minimizing the general form of cross-entropy (4) with soft targets. Utilizing additional untranscribed data from the same mobile phone short message task was found to be effective at reducing the WER of the small DNN on an internal data set when the teacher DNN was trained with either cross-entropy or sequential training criterion. Training a small student DNN is discussed in more detail in Section 4.

Ba and Caruana [24] showed that a MLP with a single, extremely wide hidden layer could be trained to similar recognition accuracy as deep architectures on TIMIT phone recognition and CIFAR-10 object recognition tasks. Rather than the probabilities produced by the softmax, the pre-softmax activations (logits) were used as regression targets for training the approximating MLP, arguing that this enables the student to better learn the internal model of the teacher. In contrast to other related work, the aim of [24] was to explore the interesting question "Do deep nets really need to be deep?" rather than obtain an accurate, faster-to-deploy NN. In order to achieve similar accuracy to the deep architectures on TIMIT, the number of parameters in the single hidden layer NN was equal to, or drastically greater than the those of the deep architectures.

Hinton and Vinyals et al. [16] also refers to the vector of class conditional probabilities as soft targets and adds a "temperature" variable to the softmax function in the output layer in order to produce a softer output distribution over the classes, as an alternative to using the logits as targets. In addition to the smoothed probabilities of the ensemble, the approximating DNN is trained to also predict the original hard targets using a weighted average of two cross-entropy objective functions, one with soft targets from the ensemble and the other with hard targets. Thus, labeled training data is required. Also, two additional hyperparameters (temperature and relative weight of hard target objective function) must be tuned. However, they were able to obtain the same WER as an ensemble with a single DNN. A single, large DNN acoustic model for mobile voice search was trained with the original labels and soft targets from an ensemble of DNNs having different parameter initializations but the same size, input features, and training set, and WER was reduced from 10.9 to 10.7 % on a development set, matching the WER of the ensemble.

We have focused on a setting for soft target training in which large amounts of untranscribed data are available, as this is often the situation for many applications we are interested in. This work adds to the growing empirical evidence that training with soft targets from an accurate teacher is an effective method of training NNs and extends the soft target training concept in a new and useful way by pairing teacher and student networks which have different input representations. While straightforward to accomplish, we believe this has considerable practical importance because it opens up a number of other potential sources for teacher networks, beyond ensembles and simply larger networks, which may be approximated with easier-to-deploy student networks. Under this strategy, potential teacher networks include SAT-DNN with speaker-normalized features [25], networks trained with i-vectors or noise estimates appended, and networks trained with wider context windows than the students, for example. While previous works on soft target training for DNN acoustic modeling have mainly relied on internal data sets, we have experimented on a widely used benchmark LVCSR data set and test sets, not just for training a smaller size DNN but also showing how soft target training can be effective for training a student DNN having the same size as the teacher but with cheaper-to-obtain input features. In addition, we also experiment with transfer across architectures using a single CNN teacher and MLP student in a new application for soft target training having only two output classes, VAD for a mobile voice search system. We confirm that soft targets from the output distribution of the teacher are more informative for the student than hard targets based on the class having the maximum prediction of the teacher and provide some interesting insight into the weight matrices learned by soft target training which suggests they tend to be fuller rank as a result of the richer information in soft targets.

We recently learned of related work by Chan et al. [26]. In their newly published work, they looked at using outputs from a large recurrent neural network as labels to train a small DNN and achieved a large reduction in WER compared to training the small DNN with hard targets from a forced alignment on the 81 hour Wall Street Journal corpus of read speech. Their formulation of loss function for training with soft targets is similar to that of Li et al. [15] and the one in this paper, but they did not include any additional unlabeled training data during soft target training.

3 Improving speaker-independent DNNs via soft targets from speaker-adaptive DNNs

In this section, we present soft target training experiments to improve generalization ability of student DNNs on a LVCSR task. In these experiments, the student DNNs and teacher DNNs have the same size but the student DNNs are trained with SI inputs and teacher DNNs are trained with fMLLR features. The aim of these experiments is to train the best SI DNN for deployment and we compare two approaches to obtaining a single-pass, SI system with a fixed DNN acoustic model size. One is a student DNN trained with soft targets from an accurate, batch-mode-adapted teacher DNN and the other is a baseline DNN trained the conventional way with hard targets generated by Viterbi alignment of the reference transcription. In Section 4, the method is applied to training of a much smaller student DNN for parameter reduction.

3.1 General setup

3.1.1 Data

We conducted experiments on the Switchboard-1 corpus which is an English conversational telephone speech transcription task. The first 100k utterances from the training data are used to create a 110-h subset as done in [3, 5, 25]. The 110-h subset was created primarily to have the remainder of the 300-h Switchboard corpus left for use as untranscribed data when studying the effect of additional untranscribed data in soft target training experiments. Thus, we have a 110-h subset of transcribed data for use when training the teacher DNNs and baseline systems and 110- and 300-h data sets for untranscribed soft target training of student DNNs[1]. For all DNN training, the data sets were further split into training and development sets making up 90 and 10 %, respectively, with the development set used for hyperparameter search and early stopping. WER is reported on the SWB portion of the 2000 Hub5 set (Hub5'00-SWB) and the FSH portion of the Spring 2003 Rich Transcription set (RT03S-FSH). Hub5'00-SWB acts as a development set in that we used it to select a language model weight parameter for RT03S-FSH results.

3.1.2 Baseline GMM-HMM system

The GMM-HMM system was built with the standard Kaldi recipe for Switchboard [27] using the 110-h transcribed subset. The baseline is trained on 13-dimensional MFCC features. Cepstral mean normalization is applied per speaker and up to second-order derivatives are appended. Linear discriminant analysis (LDA) is applied to project 9 frames of spliced features down to 40 dimensions and then a single semi-tied covariance (STC) transform is estimated. Following [3], we will refer to these as LDA+STC features. Speaker-adaptive training (SAT) is performed with a single fMLLR transform estimated per speaker. The maximum likelihood trained SAT model has 4179 tied CD states which serve as output classes for the DNN acoustic model. Trigram language models trained on Switchboard-1 and Fisher transcripts are interpolated and then pruned, giving a language model with 323K trigrams and 552K bigrams.

3.1.3 Speaker-independent DNN baseline

A SI DNN baseline is trained using the 110-h transcribed subset with 40-dimensional log-mel filterbanks which have been concatenated to form an input window of 11 frames. Mean and variance normalization are performed using a rolling window. The DNN has 6 hidden layers with 2048 sigmoid units in each layer. Both the SI and SA DNN baselines were initialized with generative pretraining by stacking restricted Boltzmann machines (RBM). For details of the pretraining procedure, see [3].

Stochastic gradient descent (SGD) is used to minimize the objective function in all DNN experiments. SGD was performed with a minibatch size of 256 frames, and the learning rate was annealed according to the relative improvement in cost on the 10 % held-out development set. Training was stopped when the relative improvement in cost on the development set was insufficient.

Frame-discriminative cross-entropy minimization with hard alignments from the GMM-HMM SAT baseline gives WERs of 19.9 and 25.1 % on Hub5'00-SWB and RT03S-FSH, respectively, for the SI DNN baseline. This result compares favorably to [25] which also performs speaker-independent experiments with a 110-h subset of Switchboard-1 and Hub5'00-SWB, so we believe this is a strong baseline. Key differences with [25] are the use of a larger DNN and slightly larger language model; however, Miao et al. [25] used the full 110-h subset for training and had a separate development set, and mean and variance normalization were performed per speaker.

Following cross-entropy training, sequential training was performed using state-level minimum Bayes risk (sMBR) criterion. We follow the procedure in [3] with training starting from alignments and lattices generated using the cross-entropy-trained DNN. After the first training pass through the full 110-h data set, alignments and lattices are regenerated and two more epochs are performed at a constant learning rate. The sequentially trained SI DNN baseline gave WERs of 18.3 and 22.6 % on Hub5'00-SWB and RT03S-FSH, respectively.

3.2 Soft target training from a DNN teacher with fMLLR

DNNs trained with SA features can offer attractive performance improvements over SI features such as log-mel filterbanks. Although most SA feature approaches were designed for a GMM-HMM-based system, these features can give strong results for DNN-based systems as well [3–6, 28]. In this section, we present work on teacher networks trained with fMLLR features.

3.2.1 Background

Speaker-normalized features $\hat{\mathbf{x}}$ are produced from initial feature vectors $\mathbf{x}$ by an affine transformation $\hat{\mathbf{x}} = M_s\xi$, where M_s is the fMLLR matrix estimated during adaptation and ξ is an extended feature vector defined as $\xi = [\mathbf{x}^T 1]$. A single transform, M_s, is estimated for each speaker or speaker cluster s by maximizing the likelihood of observing the data from s, given the model. The transforms are estimated under a GMM-HMM acoustic model but the transformed features are routinely used with DNN acoustic models as well.

For offline applications where all the adaptation data is available prior to recognition, such as broadcast news transcription and telephone transcription, multiple pass architectures running in real-time are common [29, 30]. First an initial SI decoding pass is done to generate hypotheses used for estimating a fMLLR transform, followed by a speaker-dependent decoding pass. Statistics used for cepstral mean and variance normalization, as well as fMLLR transform estimation, are computed in batch over a whole conversation side. Therefore, this approach excludes many applications we are interested in where minimum latency is important and we cannot wait until all speech has been received before starting decoding.

Incremental adaptation is an alternative to the batch-mode approach used for offline transcription tasks. A method for incremental online fMLLR adaptation was proposed in [31], and Lei et al. explored incremental online fMLLR adaptation for DNNs in [32]. In their approach, the transform is initialized to identity and incrementally updated once there is enough adaptation data for the speaker. The updated transform is then applied to the following utterances in the session, eliminating the need for making multiple decoding passes before outputting a recognition result. However, initial utterances receive no adaptation and the benefit is limited compared to the batch-mode approach where the DNN is trained in the canonical feature space and all test utterances undergo a speaker-dependent decoding. On an English mobile voice search and short message dictation task, online fMLLR adaptation gave only a 1.7 % relative reduction in WER compared to the SI DNN in [32]. Furthermore, as noted in [33], if transform statistics associated with a certain speaker or device are to be retained for future use, a large, complex infrastructure is required for carrying out the associated operations of storing, retrieving, and updating.

Because the teacher DNN is not required for deployment, a teacher DNN trained on fMLLR features is a good candidate for soft target training of a student DNN whose input features do not require additional processing steps beyond SI feature extraction. In this way, some of the accuracy gains obtained by the teacher can be leveraged by the student without the need for multiple decoding passes or complex infrastructure at time of deployment. There is likely to be some gap in performance between the SA input teacher DNN and SI input student DNN. However, even when fMLLR is feasible for online recognition, the trade-off between ease of deployment and best possible accuracy may be acceptable in many situations.

3.2.2 fMLLR DNN baseline results

To serve as a teacher DNN, we trained a SA input DNN baseline on the 110-h transcribed subset with one fMLLR transform estimated per speaker using the SAT GMM-HMM baseline and forced alignments of the reference transcriptions. These transforms are estimated on top of the LDA+STC features described in Section 3.1.2. For input to the DNN, we create a window of 11 frames of the 40-dimensional fMLLR features. The features are zero mean and unit variance normalized using a global estimate from the 110-h training data. As with the SI DNN baseline, RBM pretraining was done to initialize the network.

The cross-entropy and sequential training procedures for the fMLLR input DNN are similar to the SI DNN baseline described in Section 3.1.3. The network trained with cross-entropy gave WERs of 16.9 and 20.1 % on Hub5'00-SWB and RT03S-FSH, respectively. By further training with sMBR criterion for 3 epochs, the WERs were reduced to 15.1 and 18.2 % on Hub5'00-SWB and RT03S-FSH, respectively.

3.2.3 Experimental results

In this section, we present the results of soft target training using the fMLLR DNN baseline described in the previous section as a teacher DNN to provide labels for a student DNN to learn from. The student DNN has the same size as the SI DNN baseline and teacher DNN. The input features to the student DNN are the same as the SI DNN, 11 frames of 40-dimensional log-mel filterbanks which have been mean and variance normalized using a rolling window. However, all student DNNs start from random initialization because RBM generative pretraining was not beneficial. We experimented with doing layer-wise discriminative pretraining [4], using soft targets from the teacher instead, but this also did not perform better than random initialization. Error on the development set was reduced more rapidly at the onset of finetuning with soft targets in the pretrained networks but was overtaken after a few epochs by the network which was not pretrained. The network without pretraining continued finetuning with soft targets slightly longer and converged to a solution with lower cross-entropy error on the development set.

Soft target training experiments were carried out with the 110- and 300-h untranscribed data sets. In all experiments where fMLLR transforms are used as inputs to a teacher DNN for soft target training, the transforms are estimated in an unsupervised way and the transcriptions are not used even if we have assumed some portion of the transcriptions were available when performing supervised training of the teacher DNN (i.e., the 110-h subset). The unsupervised transforms are computed from lattices using 3 decoding passes with the SAT GMM-HMM baseline.

The cross-entropy-trained teacher DNN was used to label training examples, and results for the 110-h and 300-h untranscribed training sets are shown in Table 1, with results from the cross-entropy-trained SI DNN baseline for comparison. When trained with 110 hours of data, the student DNN reduces WER by 2.0 and 3.6 % relative (0.4 and 0.9 % absolute) on Hub5'00-SWB and RT03S-FSH, respectively, compared to the cross-entropy-trained SI DNN baseline. The improvement using the 110-h subset is modest but when the amount of untranscribed data for training the student DNN is increased to 300 h, soft target training is more effective and the WER is reduced by 7.5 and 9.6 % relative.

Sequential training is often used to obtain the best results so it is critical that soft target training also outperforms a sequentially trained network too. Results for soft target training with labels from the sequentially trained teacher DNN are shown in Table 2. With only 110 h of untranscribed data, the student DNN does not seem to offer much improvement over the sequentially trained SI DNN baseline, giving a 2.2 % relative reduction in WER on Hub5'00-SWB but no improvement on RT03S-FSH. However, with the 300-h untranscribed training set, relative reductions in WER of 8.2 and 6.2 % are obtained on Hub5'00-SWB and RT03S-FSH, respectively.

3.3 Student DNN domain adaptation with untranscribed data

We were also interested in seeing how even greater amounts of untranscribed data might impact the effectiveness of soft target training, especially when an additional data source may be from a slightly different but highly similar domain as the one the teacher was trained on. Approximately 840 h of data was randomly selected from the Fisher English conversational telephone speech corpus. We then added approximately 825 h of that to the Switchboard training data to create a 1100-h untranscribed training set. The remaining 15 h were added to the development set.

Experiments using outputs from the cross-entropy-trained teacher DNN, and sequentially trained teacher DNN, to label training examples for student DNNs were performed and results are shown in Table 3. When comparing to the SI DNN baseline trained with cross-entropy, the student DNN trained with outputs from the fMLLR teacher DNN trained with cross-entropy ("FMLLR-Xent outputs") achieves relative reductions in WER of 11.1 and 16.7 % on Hub5'00-SWB and RT03S-FSH, respectively. When comparing to the SI DNN baseline sequentially

Table 1 WER (%) for 6×2048 network with soft targets from cross-entropy-trained teacher with fMLLR inputs

Input features	Targets	Data	Hub5'00-SWB	RT03S-FSH
FMLLR	Hard alignment	110 h transcribed	16.9 %	20.1 %
FBANK	Hard alignment	110 h transcribed	19.9 %	25.1 %
FBANK	FMLLR-XEnt outputs	110 h untranscribed	19.5 %	24.2 %
FBANK	FMLLR-XEnt outputs	300 h untranscribed	*18.4 %*	*22.7 %*

Table 2 WER (%) for 6×2048 network with soft targets from sequence-trained teacher with fMLLR inputs

Input features	Targets	Data	Hub5'00-SWB	RT03S-FSH
FMLLR	Hard alignment	110 h transcribed	15.1 %	18.2 %
FBANK	Hard alignment	110 h transcribed	18.3 %	22.6 %
FBANK	FMLLR-sMBR outputs	110 h untranscribed	17.9 %	22.6 %
FBANK	FMLLR-sMBR outputs	300 h untranscribed	*16.8 %*	*21.2 %*

trained with sMBR, the student DNN trained with outputs from the fMLLR teacher DNN trained with sMBR ("FMLLR-sMBR outputs") achieves relative reductions in WER of 10.4 and 13.7 % on Hub5'00-SWB and RT03S-FSH, respectively.

Soft target training continued to be effective as the untranscribed data set size was increased by adding Fisher data, yielding WERs substantially lower than training with only the Switchboard data, even though the teacher DNN was never trained on data from Fisher. While WERs are also decreased on Hub5'00-SWB, adding the Fisher data to the untranscribed training set had a much greater impact on the WERs for RT03S-FSH. This suggests that soft target training may be of some use for domain adaptation with untranscribed data. We have already observed that adding untranscribed data from the same domain (i.e., Switchboard-1) as the transcribed training set is effective; but when a teacher DNN performs reasonably well in another target domain (i.e., RT03S-FSH), untranscribed data from that domain can be used to reduce the mismatch between training and test conditions for the student DNN, bringing performance of the student DNN even closer to that of the teacher DNN in the target domain.

It is also interesting to ask to what extent the gap in performance between teacher and student can be closed. In the case of the cross-entropy-trained teacher DNN, 73.3 and 84.0 % of the improvement in WER was transfered to the student DNN on Hub5'00-SWB and RT03S-FSH, respectively. In the case of the sMBR-trained teacher DNN, 59.4 and 70.5 % of the improvement in WER was transferred to the student DNN. The performance gap between student and teacher is considerably less on RT03S-FSH than on Hub5'00-SWB. We believe this is due to the large amount of untranscribed Fisher data added, making the student more effective in that domain.

4 Parameter reduction by soft target training

There is growing interest in ASR for embedded systems, not just in mobile devices and gaming consoles but also in wearable devices, automobiles, and smart appliances to name a few. Accordingly, an increasing demand for expanded functionality should be expected as well. DNNs have offered impressive gains in recognition accuracy but can be considerably more computationally and storage intensive than the previous paradigm of GMM-HMM, making them more challenging to deploy in embedded systems. This section focuses on soft target training as a way of obtaining compact yet accurate DNN.

4.1 Approaches for faster, smaller DNN

Much effort has gone into finding ways to improve DNN runtime. Vanhoucke et al. lay out several tools for creating a highly optimized CPU implementation using fixed-point arithmetic, SSE instructions, and lazy evaluation of the softmax layer [34]. Lei et al. [35] applied these optimizations along with frame skipping [36], which computes posteriors every *n*th frame and reuses them for *n* consecutive frames, to a small DNN for embedded recognition that ran in real-time with large accuracy improvements over a GMM acoustic model but still had a small footprint. However, accuracy was considerably less than that of a full-size DNN server-based system having an order of magnitude more parameters.

Trying to reduce the number of parameters in a large DNN after training, rather than directly training a small DNN, may result in less loss of accuracy relative to the original large DNN. Numerous approaches have been designed around this concept. DNN parameter sparseness was exploited in [37], allowing for large reductions in storage. However, non-zero parameters are randomly distributed in each layer, requiring the use of indices to keep track of them and the speedup of calculation depends heavily on the implementation and hardware used.

Table 3 WER (%) for 6×2048 network with soft targets and additional untranscribed from Fisher

Input features	Targets	Data	Hub5'00-SWB	RT03S-FSH
FBANK	Hard alignment	110 h transcribed (Xent)	19.9 %	25.1 %
FBANK	FMLLR-Xent outputs	1100 h untranscribed SWBD + FSH	*17.7 %*	*20.9 %*
FBANK	Hard alignment	110 h transcribed (sMBR)	18.3 %	22.6 %
FBANK	FMLLR-sMBR outputs	1100 h untranscribed SWBD + FSH	*16.4 %*	*19.5 %*

Alternatively, low-rank matrix factorization can be used to exploit the redundancy of weight matrices found in DNN. A linear bottleneck layer can be inserted in between the last hidden layer and outputs *prior* to training, as in [38], or singular value decomposition (SVD) can be applied to the hidden layer weight matrices *after* training the network to produce a low-rank approximation [39]. SVD-based restructuring of DNN may result in the loss of some accuracy but Xue et al. found that if the approximation is not too drastic, the loss may be recovered by re-finetuning of the restructured DNN [39]. However, it may not be possible to realize a large enough reduction in parameters to shrink a large DNN designed to run on a server down to the size required for embedded recognition on a device using SVD restructuring without an appreciable loss in accuracy.

4.2 Training a small DNN with soft targets from a large DNN

In Section 3.2, we examined soft target training using a student network having the same size as the teacher network. A large, highly accurate teacher could also be used to relabel examples for training a much smaller student network. For the small student network, we use a 5 hidden layer, 512 hidden unit network (5×512) with log-mel filterbank inputs. The 6 hidden layer, 2048 hidden unit network (6×2048) sequentially trained with fMLLR input features is used again as the teacher network in this section. Optimizations like those found in [34] could also be applied to a network obtained through this approach for further speed-up. Note that unlike weight matrix decomposition and node pruning [40] approaches, soft target training is considerably more flexible because the input feature type and NN architecture are not restricted to be the same as the teacher. Furthermore, an ensemble of networks could be used to relabel inputs for the small network.

As noted earlier, Li et al. have also recently studied soft target training for a small-sized DNN [15]. A teacher DNN having 5 hidden layers with 2048 hidden units was used to label outputs for a student DNN having 5 hidden layers with 512 hidden units, giving a reduction in parameters of approximately 85 % compared to the large DNN and relative reductions in WER of 5.08 and 2.91 % compared to a small DNN having the same size but trained with hard targets using cross-entropy and sequential training, respectively. All networks were trained with the same speaker-independent features, log-mel filterbank inputs with up to second-order derivatives. The task was a mobile phone short message dictation task evaluated on an internal data set.

In contrast, we use a large teacher DNN sequentially trained with SA features (fMLLR) to train a small DNN with SI features (log-mel filterbank). We demonstrate that a small DNN trained in this way cannot only obtain lower WER than a small DNN sequentially trained with hard targets but can actually *surpass* the performance of a large sequentially trained DNN with SI features having an order of magnitude more parameters when a substantial amount of untranscribed data is used for training the small student DNN[2]. A small baseline DNN having 5 hidden layers with 512 hidden units was sequentially trained with log-mel filterbank inputs and targets derived from a hard alignment. Both the teacher and baseline networks were trained with 110 h of transcribed data. The same teacher DNN from Section 3.2 that was sequentially trained with fMLLR inputs was used to provide soft targets for the small student which sees log-mel filterbank inputs. Table 4 shows the WERs for training the small DNN student with 110 and 300 h of untranscribed data. When 110 h of untranscribed data is used for soft target training, the student network performs slightly worse than the sequence-trained small baseline DNN trained with hard targets. However, with the additional untranscribed training data, the student DNN gives 4.6 and 3.3 % relative reductions in WER on HUB5’00-SWB and RT03S-FSH compared to the sequence-trained baseline DNN of the same size.

4.3 Beyond lossless parameter reduction

Soft target training was effective for training the small DNN student when the amount of untranscribed training data was increased from 110 to 300 h. Similar to Section 3.3, we tried adding additional untranscribed training data from the Fisher corpus when training the small DNN student. Table 5 shows the results for the small DNN student trained with the large untranscribed set and a large DNN baseline sequentially trained with log-mel filterbank inputs and the 110-h transcribed subset for comparison. This allows for a comparison between soft

Table 4 WER (%) for 5×512 network with soft targets from 6×2048 sequence-trained teacher with fMLLR inputs

Input features	Targets	Data	Hub5’00-SWB	RT03S-FSH
FMLLR	Hard alignment	110 h transcribed	15.1 %	18.2 %
FBANK	Hard alignment	110 h transcribed	19.6 %	24.1 %
FBANK	FMLLR-sMBR outputs	110 h untranscribed	20.0 %	24.7 %
FBANK	FMLLR-sMBR outputs	300 h untranscribed	*18.7 %*	*23.3 %*

Table 5 Parameter reduction by soft target training

Network	# of Params	Data	Hub5'00-SWB	RT03S-FSH
6×2048 (hard align., sMBR)	30.4 mil.	110 h transcribed (SWBD)	18.3 %	22.6 %
5×512 (soft target-trained)	3.4 mil.	1100 h untrans. (SWBD + FSH)	***18.0%***	***21.4%***

target training and parameter reduction methods applied to a trained network, such as SVD restructuring. Note that even if we assume no degradation in accuracy when applying SVD or other methods to reduce the full size log-mel filterbank model in Table 5 to have the same number of parameters as the small DNN[3], the small DNN student trained with soft targets using 1100 h of untranscribed data still significantly outperforms it. Viewed in terms of parameter reduction, this is a 88.8 % reduction in parameters while achieving 1.6 and 5.3 % relative reductions in WER. Thus, we believe that soft target training of a small DNN student can be a very effective method for parameter reduction, even when the reduction in parameters is drastic, as long as the teacher network is very accurate and the amount of untranscribed data is large. It allows for very efficient use of parameters in a small model.

5 Voice activity detection experiments

5.1 Problem description

When the training targets are tied context-dependent HMM states, one can readily imagine what kind of information about the structure of the data is passed on from teacher to student via the teacher's predictions. These predictions may reveal similarities between states within a triphone HMM, between triphones sharing a common center phone, and relative relationships all the way up to broad classes, such as vowels and consonants, conceivably. In this section, we turn from an application with thousands of output classes resulting from decision tree clustering to the other extreme, to see if soft target training is useful in a binary classification task that is also germane to ASR.

VAD is an integral front-end component in many speech processing systems and is used for identifying speech and non-speech frames in an audio signal. While there may be several ways to approach VAD [41], we have chosen to formulate the problem as a binary classification task using neural networks to differentiate between speech and non-speech at the frame level, similar to [42]. Several recent examples of neural network-based VAD can be found in relation to the DARPA RATS program with noisy communication channels [43, 44].

5.2 System description

The teacher network in these experiments is a CNN. CNNs provide shift, scale, and distortion invariance to some extent through use of local receptive fields, shared weights, and spatial or temporal sub-sampling [45]. When applied to speech processing, CNN can compensate for distortions in the frequency domain and show improvements over DNN [11]. However, computing the activations of a convolutional filter (unit) is much more expensive than a traditional hidden unit in an DNN. We use a convolutional architecture similar to the one used in [28] which is as follows. There are 2 convolutional layers with the first one having 256 filters with 9×9 receptive fields, followed by max pooling in frequency with a pooling size of 1×3. The second layer has 256 filters with 3×4 receptive fields. Outputs from this layer are input to 2 fully connected feedforward layers having 1024 hidden units with a sigmoid non-linearity. The CNN was trained using the PDNN library [46].

The architecture for the student model is a MLP having 2 hidden layers with 1024 sigmoid hidden units each. Input features are 40-dimensional log-mel filterbanks (spanning 0–8 kHz) with delta and delta-delta features appended. The 5 preceding and 5 following frames are included giving a 1320-dimensional input vector. The features are zero mean and unit variance normalized using a global estimate from the training data. All networks are trained using these features, in contrast to the acoustic modeling experiments in Section 3 where teacher and student networks were trained on different input features.

5.3 Experiments and results

This work studies VAD as a front-end component of a mobile voice search system. As such, we are primarily interested in using VAD for identifying non-speech periods which can be dropped from the input of the speech recognizer, as well as deciding when the utterance has ended. We evaluate the approach using an internal Yahoo! Japan mobile voice search and voice dialog data set. The CNN teacher and MLP baseline were trained using approximately 320 h of data with hard targets. We trained MLP student networks with approximately 320, 640, and 960 h of unlabeled data using soft targets from the CNN. A development set of 4000 utterances was used for system development and early stopping of training based on relative reduction in the cost. A minibatch size of 256 and momentum of 0.9 were used. Ground truth labels for our VAD experiments were obtained using human-annotated start and end times of speech and word transcriptions. Non-speech frames at the beginning and end of all utterances are determined using human-annotated start and end times of speech. Short pauses in speech occurring within the duration of speech are detected by aligning the portion of an utterance annotated as speech, along with a small margin of silence at the start and end boundaries, using the word transcription and GMM-HMM baseline. CD state labels for each frame in the alignment are then converted to speech and non-speech frame labels by mapping them to speech and silence classes. Long periods of silence are not expected to occur within the duration of speech as the average length of the utterances is very short (around 3 s).

Evaluation is done using four held-out test sets referred to as "Mob-1," "Mob-2," "Mob-3," and "Mob-4." Mob-1 and Mob-4 consist of mobile voice search data. Mob-2 and Mob-3 consist of voice dialog data. Each test set contains approximately 10,000 utterances. Note that Mob-1 consists of data collected from noisier conditions than the other three test sets. It has a considerably lower average signal-to-noise ratio, making it more challenging. Equal error rate (EER) serves as the performance metric. To calculate EER, we must first determine the false rejection rate (FRR) and false acceptance rate (FAR), which are the percentage of speech frames which were misclassified as being non-speech, and the percentage of non-speech frames which were misclassified as speech, respectively. EER is the operating point at which these two types of errors occur equally.

Results on the four test sets are shown in Table 6. The CNN considerably outperforms the MLP baseline trained on hard targets, making it a good candidate for a teacher network. With the same amount of training data, the MLP student network with soft targets from the CNN is able to surpass the MLP baseline on all four test sets. This is interesting since there are only two target classes to learn compared to the acoustic modeling case where we expect the student's learning to be enriched by the intricate relationships between output classes found in the teacher's outputs. Nonetheless, learning to generalize like the CNN is quite beneficial for the MLP student. When the amount of unlabeled training data is increased from 320 to 640 h, EER of the MLP student network is even further decreased. The trend continues when unlabeled training data is increased to 960 h, giving relative reductions of 7.2–11.5 % compared to the MLP baseline. With this amount of unlabeled data, the MLP student is able to recover between 88.9 and 97.0 % of the gain achieved by the CNN despite not having any convolutional layers itself.

6 Analyzing soft target training

6.1 Interpretation of soft targets

The wisdom of the teacher is in its outputs. The outputs of the teacher reflect what it has learned about the relationship between specific inputs and similar output classes over the course of training. As noted in [16], the relative strengths of the teacher's outputs imply the extent to which a given input appears similar to examples belonging to other classes and this is important information for learning how to generalize well. Consider a DNN acoustic model with outputs corresponding to tied CD states. The set of states that will be modeled are determined by decision tree clustering which produces a large number of tied CD states, many of which differ in only a small but still significant way, such as left or right context, or by the state's position within a HMM. Given that the states are structured in this way, it is understandable that many inputs are likely to appear to the teacher to be somewhat similar to examples from several different classes of states.

Examining bar plots of the posterior probability output by the teacher network from Section 3.2.2 illustrates this idea clearly. To create the plots shown in Fig. 2, we have selected two inputs to forward propagate through the fully trained teacher network. We refer to the two inputs as "frame A" and "frame B" for simplicity but each one is a single, 11-frame window of unsupervised fMLLR features. The center frames of both inputs are from state "1238," which would be the training label for both inputs if hard targets were generated from a forced alignment of a reference transcription. Frames A and B were spoken by different speakers, "speaker 1" and "speaker 2," respectively. These inputs were taken from the Switchboard 300-h untranscribed data set used for training student DNNs in our experiments.

Figure 2 shows the posterior probability output by the teacher DNN for frames A and B. Due to the large number of states being modeled, state IDs for only the top five states with the highest posterior probabilities are shown next to their corresponding bars. As expected, many of

Table 6 EER (%) for 2×1024 network with soft targets from CNN

Network	Targets	Data	Mob-1	Mob-2	Mob-3	Mob-4
CNN	Hard alignment	320 h labeled	4.10 %	2.51 %	2.46 %	2.77 %
MLP baseline	Hard alignment	320 h labeled	4.46 %	2.75 %	2.79 %	3.09 %
MLP student	CNN outputs	320 h unlabeled	4.26 %	2.61 %	2.60 %	2.91 %
MLP student	CNN outputs	640 h unlabeled	4.19 %	2.55 %	2.52 %	2.85 %
MLP student	CNN outputs	960 h unlabeled	*4.14 %*	*2.52 %*	*2.47 %*	*2.79 %*

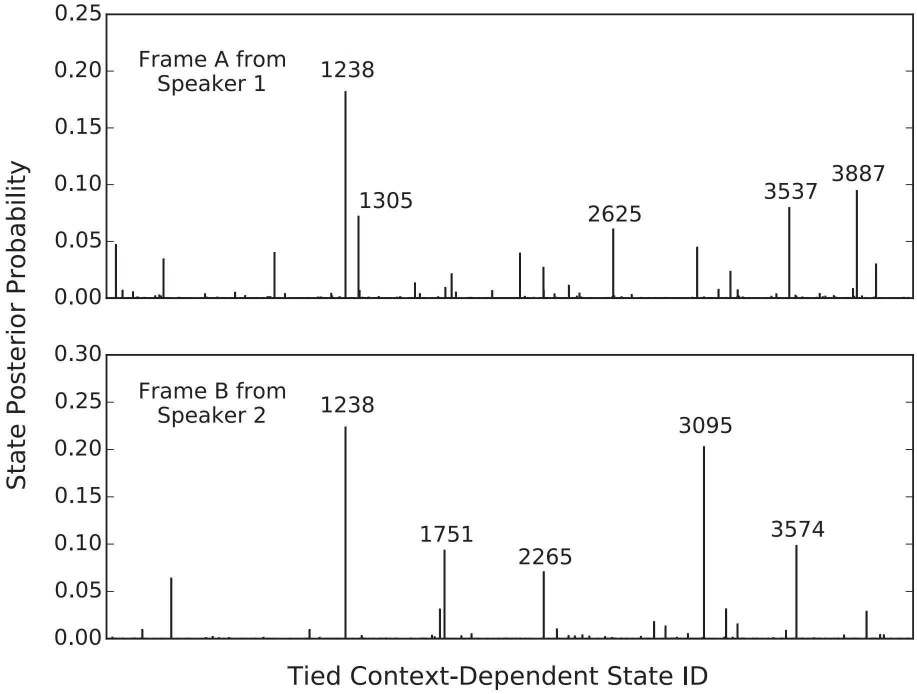

Fig. 2 Teacher DNN outputs for inputs from the same class. Posterior probability outputs by a teacher DNN for two inputs belonging to the same tied CD state are shown with state IDs for the five states having the highest posterior probability shown next to their corresponding bars. Even though the center frame belongs to the state, the output distributions are markedly different for the two inputs from different speakers, revealing different notions of similarity to other states learned by the teacher. This information can be conveyed to the student through soft targets but not hard targets

these correspond to states having the same center phone and HMM state position as state "1238," which has center phone "n" and HMM state position 0. While state "1238" was correctly given the highest posterior probability in both cases, we can see that the output distributions look quite different. Different sets of states had relatively high non-zero probabilities and this is important information for instructing the student how to generalize well. This information can be conveyed to the student through soft targets but not hard targets.

Figure 2 also demonstrates that outputs from the teacher can sometimes have high entropy. We measured the average entropy of this teacher's outputs on the full Switchboard data set to get a better idea of the teacher's predictions as a whole. The average entropy was 1.49 which suggests that outputs from this network often place a probability of well below 0.9 on any particular class for each input.

When inputs are particularly difficult for the teacher to classify, the teacher's outputs will have a higher entropy which signals to the student network not to be too sure about predicting a single class for the given input. This acts as a kind of regularization by making the target function the student should learn smoother. Training with hard targets in cases where similar inputs are mapped to dissimilar targets using 0/1 labels requires learning a highly nonlinear function [47]. We believe that the regularization effect of soft targets can be particularly important for avoiding overfitting in such cases.

We examine this idea further in the context of speech/non-speech classification for VAD by drawing an analogy to the thought experiment in [47]. Caruana et al. [47] provided a clear example of how functions with 0/1 targets can result in regions where similar inputs are mapped to dissimilar targets, even when the training set is sampled from a simple probability distribution. We have adapted it for our discussion here. While this is an obvious simplification, we believe it is useful for illustrating how soft target training can be beneficial even in a binary classification task, such as VAD. Suppose a frame of acoustic data x is labeled as containing speech with probability p and non-speech with $1 - p$, determined by the function shown in Fig. 3 and a training set with speech or non-speech labels is sampled from this distribution.

Where the probability assigned to x is low, there will be many non-speech frames and where the probability is high, there will be many speech frames. We may suppose these values of x correspond to frames which occur within long stretches of non-speech or speech, respectively, and can usually be classified fairly easily by the teacher network. On the other hand, values of x lying in the relatively flat region of Fig. 3 are labeled as speech

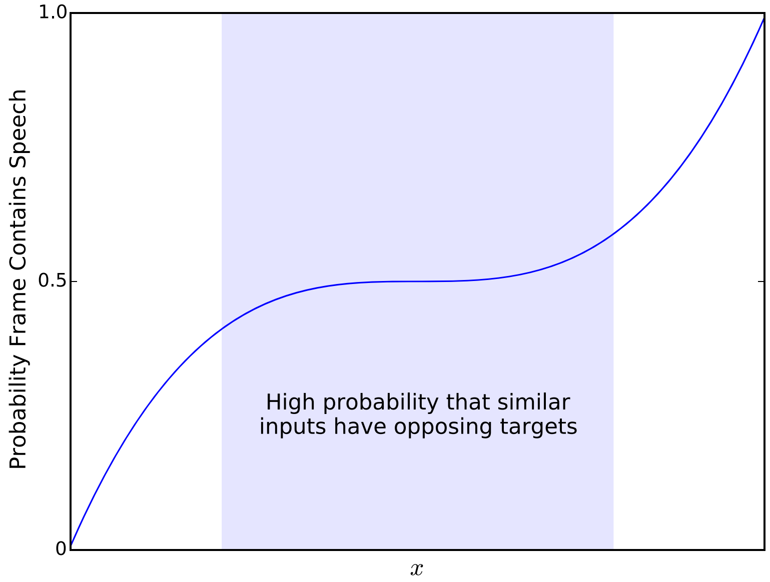

Fig. 3 Probability function for assigning speech/non-speech frames. Probability function that assigns probability $p = f(x)$ that frame contains speech for each value of acoustic data *x*. When a training set is sampled from this distribution, similar values of *x* are likely to be mapped to different 0/1 (non-speech/speech) targets in the shaded region. The resulting function will be highly nonlinear in this region and difficult to learn using hard targets

or non-speech with nearly equal probability, resulting in a situation where similar inputs are frequently mapped to dissimilar targets. We may suppose these frames lie near the boundary of transitions between speech and non-speech and can be difficult to classify correctly. Under the traditional training scheme, these inputs are labeled with hard targets and the network must learn a function that is highly nonlinear in this region. However, with soft target training, a teacher network can assign higher entropy outputs to these frames, which may be closer to the actual probability distribution. This would result in a less nonlinear function for the student compared to 0/1 labels, making the function smoother and possibly reducing the overfitting which can accompany the learning of highly nonlinear functions.

6.2 Soft vs. hard targets from teacher

In semi-supervised approaches, the transcribed data is augmented with additional untranscribed data and a supervised training signal is generated by labeling the untranscribed data with 0/1 labels based on the class having the maximum prediction output by the current network. In our experiments, we have also found adding additional untranscribed data to the training set very beneficial but with the distinction that labels are assigned using the soft targets from a teacher network. We have argued that important information is transferred in the soft targets which improves generalization of the student network, resulting in the lower WERs and EERs observed in Sections 3, 4, and 5. If the teacher was used to assign 0/1 labels instead of soft targets, such information would not be present. We verified this assertion by comparing the two alternatives for labeling untranscribed data using a teacher network.

The fMLLR input DNN sequence trained with the 110-h transcribed subset of Switchboard served as a teacher network and two student networks were trained with log-mel filterbank inputs and 300 h of untranscribed Switchboard data using outputs from the teacher network. One student DNN was trained with soft targets and the other was trained with hard targets generated by assigning a posterior probability of 1.0 to the class having the strongest prediction by the teacher. All networks had 6 hidden layers and 2048 hidden units.

The results for the soft target-trained student, and the student trained with outputs from the teacher which have been converted to hard targets, are shown in Table 7. We can see that the student DNN trained with soft targets from the teacher obtained WERs that were substantially lower than the student DNN trained by converting the

Table 7 WER (%) for students trained with soft vs. hard targets from teacher

Targets	Hub5'00-SWB	RT03S-FSH
Hard	17.9 %	22.4 %
Soft	*16.8 %*	*21.2 %*

teacher's outputs to hard targets. Relative reductions in WER of 6.1 and 5.4 % on HUB5'00-SWB and RT03S-FSH resulted from using the predictions directly output by the teacher instead of converting the teacher's outputs to 0/1 labels. This improvement can only be attributed to the use of soft targets. This should dispel a potential point of doubt by confirming that the output distribution provided by the teacher in our experiments is sufficiently soft and not so peaked as to be indistinguishable from hard targets based on the class of the teacher's maximum prediction. Therefore, the gains observed in our experiments cannot solely be the result of using additional untranscribed data and passing the inputs through a another network and making the same hard predictions. These soft targets carry some relevant information that is not conveyed with only a maximum prediction from the teacher.

6.3 Rank and information content of student weight matrices

Clearly, there is a fair amount of redundancy in the parameterization of very large neural networks. Denil et al. showed that given a few weight values it is possible to predict many of the remaining values [48]. When many of the weights are highly correlated, much of the capacity of large NNs is not really being used effectively. Under such conditions, SVD parameter reduction approaches [39] are very successful for large DNNs. If we accept that soft target training provides additional information that is more useful than hard targets, then it follows that learning this extra structure is accompanied by a fuller utilization of DNN capacity, meaning less redundant parameters. To measure this, we look at the rank of the weight matrices of networks trained with soft targets. The rank of a matrix is often interpreted as indicating its "information content" in applied settings such as image compression. Intuitively, a matrix with lower rank has lower information content compared to a matrix having the same order but higher rank, in the sense that it is more easily compressed with fewer elements needed for representation.

The ranks of the hidden layer weight matrices for the 6 hidden layer, 2048 hidden unit (6×2048) and 5 hidden layer, 512 hidden unit (5×512) networks are shown in Table 8. Ranks of input and output layers are not shown. The networks were trained with the 110-h subset using either soft targets from the fMLLR teacher DNN or the original hard targets. We can see that being trained to generalize like the teacher network using soft targets produces hidden layer weight matrices having fuller rank than training with hard targets. Even for the 6×2048 network, which is a large network for the 110-h subset, the soft target-trained weight matrices are nearly full rank, indicating much greater information content coming from the soft targets. In this sense, soft target training has better leveraged the available capacity of the large network.

Table 8 Rank of hidden layer weight matrices for networks trained with soft vs. hard targets using 110 h of data

Hidden layer	6×2048		5×512	
	Soft	Hard	Soft	Hard
1	2043	943	511	511
2	2040	731	511	510
3	2040	562	512	499
4	2039	483	512	498
5	2042	487	–	–

While the information content is much greater, it is not necessarily all relevant and it is difficult to assess to what extent it is necessary for good recognition performance. We are not able to observe this effect as readily in the weight matrices of the smaller networks since they have very limited capacity and are already nearly full rank with just hard targets.

In addition to the rank of the hidden layer weight matrices, we can also examine the distribution of singular values. The relative information content (RIC) [49] of the singular value decomposition of matrix W is defined as:

$$\mathrm{RIC}(k) = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{N} \sigma_i}, \tag{5}$$

where k is the number of singular values σ retained out of N total singular values. Having an RIC value near one indicates that nearly 100 % of the information contained in the original W is retained in a low-rank approximation with the first k singular values.

Representative plots of RIC are shown in Fig. 4 for the fourth and third hidden layer weight matrices of the 6×2048 and 5×512 networks, respectively. Plots for the other square hidden layer matrices in these networks look similar. The soft target-trained networks were trained with 300-h untranscribed data using outputs from the fMLLR teacher DNN. The hard target-trained networks were trained with 110-h transcribed data. To confirm that the difference in RIC curves of the 6×2048 networks is not merely a result of 110 h of transcribed data being insufficient for that network size, we also included a 6×2048 SI input DNN trained with 300-h transcribed data. As with the rank, the RIC curves look similar for the small network, but the curves for the 6×2048 show very different distributions of singular values for the soft and hard target-trained networks. Unlike the hard target-trained networks in which the RIC rapidly increases and approaches 1.0 with only 500 singular values, the increase in RIC is much more gradual for the soft target-trained network meaning many more singular values will be important for representing the weight matrix of this layer. Indeed, with only 500 singular vectors, the RIC of the hard

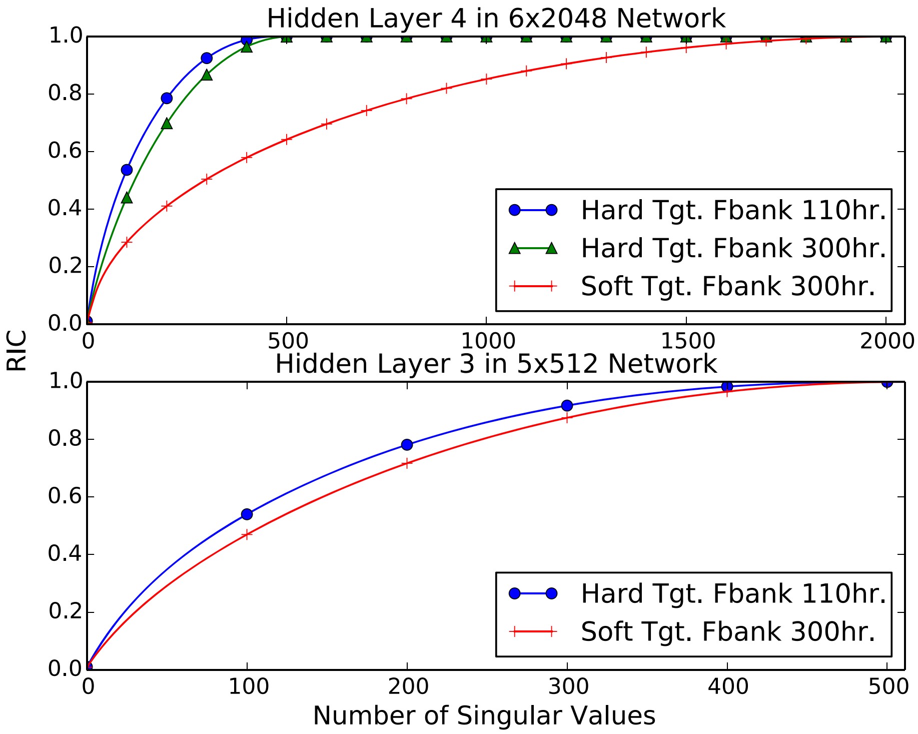

Fig. 4 Distribution of singular values of hidden layer weights. RIC shown for 6×2048 and 5×512 networks trained with hard and soft targets. Soft target-trained networks have a slower decay in singular values, requiring more singular values to be retained for a given value of RIC compared to hard target-trained networks

target-trained network is already nearly 1.0, compared to approximately 0.62 for the soft target-trained network. We attribute this to richer information learned by the soft target-trained network required in order to generalize like the teacher network. As with rank, differences in RIC are harder to observe in the 5×512 networks due to their limited capacity. Nonetheless, useful information is acquired by the small network through soft target training. We have confirmed this empirically in Section 4 by observing lower WERs with the soft target-trained 5×512 network as more untranscribed data is added. However, it is not apparent based only on rank and RIC.

7 Conclusions

We have explored a strategy for training easier-to-deploy DNN acoustic models using outputs from more accurate NNs that may be too expensive-to-evaluate for deployment. The prediction outputs by the fully trained, teacher network are assigned as soft targets for the less complex, student network to learn to predict, rather than the original labels. Having a highly accurate teacher network and large amount of additional untranscribed data are important for obtaining the best student network with this approach. Thus, it is best suited for settings in which a large amount of additional untranscribed data is available. We have extended this soft target training framework in a new and useful way by pairing teacher and student networks that have different input representations. Soft target training offers many opportunities for leveraging some of the gains obtained through state-of-the-art NN approaches while balancing the potential constraints of computational complexity and low memory footprint required for a specific deployed system. This is increasingly important as demand for online ASR grows in resource-constrained systems.

On a LVCSR task using the Switchboard-1 corpus, we demonstrated how accuracy of a speaker-independent DNN can be improved using soft targets from a DNN of the same size but trained with fMLLR inputs. Relative reductions in WER of 8.2 and 6.2 % were obtained over a sequence-trained DNN using hard aligned targets and log-mel filterbank inputs. Furthermore, when the untranscribed data set for soft target training was augmented with data from the Fisher corpus, WER was further decreased, giving relative reductions of 10.4 and 13.7 %, even though the teacher DNN was never trained on data from Fisher.

In Section 4, we studied parameter reduction using soft target training. A small student DNN having 3.4 million parameters trained with soft targets from a larger, sequence-trained DNN obtained 4.6 and 3.3 % relative reductions in WER compared to a small DNN of the same size, sequence-trained with hard aligned targets and log-mel filterbank inputs. Having an adequate amount of untranscribed data for soft target training is important. Accuracy of the small student DNN was greatly improved when the set of untranscribed data was increased further with data from the Fisher corpus, giving relative reductions in WER of 8.2 and 11.2 %. Viewed in terms of parameter reduction, this is a 88.8 % reduction in parameters while actually achieving a 1.6 and 5.3 % relative reduction in WER compared to the 30.4 million parameter DNN sequence-trained with log-mel filterbank inputs and hard aligned targets.

We also studied a new application for soft target training, VAD for a mobile voice search system. We found that transfer across architectures was very successful when training a MLP for classifying speech and non-speech frames with soft targets from a CNN. Relative reductions in EER of 7.2–11.5 % were obtained compared to training with hard targets from Viterbi alignment.

While between 88.9 and 97.0 % of the gain in performance obtained by a CNN could be transferred to an MLP student in our VAD experiments, the gap in performance between teacher and student was considerably larger in some of the SA feature experiments from Sections 3 and 4. Results have shown that the gap continues to be reduced as more untranscribed data is added for soft target training, but it would be interesting to see to what extent the gap between teachers with SA inputs and students with SI inputs can be eliminated with even more untranscribed data or better soft target training objective functions besides cross-entropy.

In Section 3.2.3, Table 2, no improvement was observed on RT03S-FSH when the 110-h training set was used to train the student with the outputs from a sequentially trained teacher relative to the baseline DNN sequentially trained with hard targets. Moreover, in Section 4.2, Table 4, when the 110-h training set was used to train a small student with the outputs from a large sequentially trained teacher, a degradation in accuracy relative to the small baseline DNN sequentially trained with hard targets was observed. While we were able to obtain substantial reductions in WER in both these cases when additional untranscribed data was used, it appears that it is more difficult to train a student using the cross-entropy loss function with soft targets from a teacher which has been sequentially trained. This is evidenced by comparing the gap in performance between student and teacher when the teacher is either trained with cross-entropy or sequentially trained. From Table 3, and given the WER of the teacher networks, we can calculate that approximately 73 and 84 % of the reduction in WER seen when comparing the cross-entropy-trained teacher to the hard target-trained baseline DNN on Hub5'00-SWB and RT03S-FSH, respectively, was transferred to the student. However, when the student was trained with soft targets from the sequentially trained teacher, only around 59 and 70 % of the reduction in WER seen when comparing the sequentially trained teacher to the hard target-trained baseline DNN on Hub5'00-SWB and RT03S-FSH, respectively, was transferred to the student. Nonetheless, we believe that using cross-entropy to train a student with soft targets from a sequentially trained teacher is a reasonable approach to take when additional untranscribed data is available based on the results we have observed. More effective loss functions for soft target training with outputs from a sequentially trained teacher may be worth investigating in the future.

Along the same line of thought, we may consider applying sequential training to the student after training with soft targets from the teacher using cross-entropy has converged. We performed a few iterations of sequential training using the sMBR criterion with a small learning rate on student DNNs following soft target training with cross-entropy. This was done in a supervised setting using hard targets with the 110-h transcribed training set. Adding sequential training with hard targets from transcribed data after soft target training with outputs from the sequentially trained teacher was not very useful for the small 5×512 DNN. Adding sequential training yielded reductions in WER ranging from 0.1 to 0.3 % absolute for the small student DNN trained with 110 and 300 h of untranscribed data but degraded WER on the small student DNN which had been trained with the large untranscribed data set augmented with data from the Fisher corpus. Results were somewhat better for the larger, 6×2048 student DNN which saw reduction in WER ranging from 0.2 to 0.4 % absolute but still experienced a degradation in WER on RT03S-FSH for the student DNN which had been trained with the large untranscribed data set augmented with data from Fisher.

Finally, the ensembles of networks used in other recent soft target training studies have been fairly homogeneous in terms of architecture and training data [16, 24]. While not in the context of soft target training, combinations of various architectures for ASR are being studied. In [12], an ensemble of several kinds of networks was created with the outputs combined in a final hidden layer and applied to phone recognition. DNN and CNN were jointly trained in [50] for a LVCSR task. With architectures like CNN and recurrent neural networks looking promising for LVCSR and speech processing tasks, creating a teacher ensemble with diverse combinations of different architectures and training data could be an interesting area to explore. However, increased training time associated with training all the component networks, as well as generating their outputs for student training, could be an obstacle.

Endnotes

[1] In Sections 3.3 and 4.3, we augment the untranscribed set further with data from the Fisher corpus to create a much larger untranscribed data set.

[2] We emphasize that were are not talking about the student DNN surpassing the teacher DNN which was trained on SA features.

[3] We suspect that this is a very large reduction of parameters to try to achieve using SVD restructuring.

Competing interests

The authors declare that they have no competing interests.

Author details
[1]Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan. [2]Yahoo Japan Corporation, Tokyo, Japan.

Received: 30 April 2015 Accepted: 31 March 2016
Published online: 12 April 2016

References

1. G Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, B Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Proc. Mag. **29**(6), 82–97 (2012)
2. B Kingsbury, in *Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling*. Acoustics, Speech and Signal Processing (ICASSP), 2009 IEEE International Conference On (IEEE. Taipei, Taiwan, 2009), pp. 3761–3764
3. K Veselỳ, A Ghoshal, L Burget, D Povey, in *INTERSPEECH*. Sequence-discriminative training of deep neural networks (ISCA. Lyon, France, 2013), pp. 2345–2349
4. F Seide, G Li, X Chen, D Yu, in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop On*. Feature engineering in context-dependent deep neural networks for conversational speech transcription (IEEE. Waikoloa, HI, USA, 2011), pp. 24–29
5. SP Rath, D Povey, K Veselỳ, J Cernockỳ, in *INTERSPEECH*. Improved feature processing for deep neural networks (ISCA. Lyon, France, 2013), pp. 109–113
6. TN Sainath, B Kingsbury, B Ramabhadran, P Fousek, P Novak, A Mohamed, in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop On*. Making deep belief networks effective for large vocabulary continuous speech recognition (IEEE. Waikoloa, HI, USA, 2011), pp. 30–35
7. MJ Gales, Maximum likelihood linear transformations for HMM-based speech recognition. Comput. Speech Lang. **12**(2), 75–98 (1998)
8. G Saon, H Soltau, D Nahamoo, M Picheny, in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop On*. Speaker adaptation of neural network acoustic models using *i*-vectors (IEEE. Olomouc, Czech Republic, 2013), pp. 55–59
9. A Senior, I Lopez-Moreno, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Improving DNN speaker independence with *i*-vector inputs (IEEE. Florence, Italy, 2014), pp. 225–229
10. ML Seltzer, D Yu, Y Wang, in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On*. An investigation of deep neural networks for noise robust speech recognition (IEEE. Vancouver, Canada, 2013), pp. 7398–7402
11. TN Sainath, B Kingsbury, G Saon, H Soltau, A Mohamed, GE Dahl, B Ramabhadran, Deep convolutional neural networks for large-scale speech tasks. Neural Netw (2014). in press
12. L Deng, JC Platt, in *INTERSPEECH*. Ensemble deep learning for speech recognition (ISCA, Singapore, 2014), pp. 1915–1919
13. X Zeng, TR Martinez, Using a neural network to approximate an ensemble of classifiers. Neural. Process. Lett. **12**(3), 225–237 (2000)
14. C Buciluǎ, R Caruana, A Niculescu-Mizil, in *Knowledge Discovery and Data Mining, The 12th ACM SIGKDD International Conference On*. Model compression (ACM. Philadelphia, PA, USA, 2006), pp. 535–541
15. J Li, R Zhao, JT Huang, Y Gong, in *INTERSPEECH*. Learning small-size DNN with output-distribution-based criteria (ISCA, Singapore, 2014), pp. 1910–1914
16. GE Hinton, O Vinyals, J Dean, in *NIPS Deep Learning and Representation Learning Workshop*. Distilling the knowledge in a neural network (NIPS. Montreal, Canada, 2014)
17. H Bourlard, Y Konig, N Morgan, in *EUROSPEECH*. Remap: recursive estimation and maximization of a posteriori probabilities in connectionist speech recognition (ISCA, Madrid, Spain, 1995)
18. Y Konig, H Bourlard, N Morgan, in *Acoustics, Speech and Signal Processing (ICASSP), 1996 IEEE International Conference On*. Remap-experiments with speech recognition (IEEE. Atlanta, GA, USA, 1996), pp. 3350–3353
19. A Senior, T Robinson, in *Advances in Neural Information Processing Systems (NIPS)*. Forward-backward retraining of recurrent neural networks (NIPS, Denver, CO, USA, 1996), pp. 743–749
20. Y Yan, M Fanty, R Cole, in *Acoustics, Speech and Signal Processing (ICASSP), 1997 IEEE International Conference On*. Speech recognition using neural networks with forward-backward probability generated targets (IEEE. Munich, Germany, 1997), pp. 3241–3244
21. K Veselỳ, M Hannemann, L Burget, in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop On*. Semi-supervised training of deep neural networks (IEEE. Olomouc, Czech Republic, 2013), pp. 267–272
22. F Grézl, M Karafiát, in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop On*. Semi-supervised bootstrapping approach for neural network feature extractor training (IEEE. Olomouc, Czech Republic, 2013), pp. 470–475
23. J Menke, A Peterson, M Rimer, T Martinez, in *Neural Networks (IJCNN), 2002 International Joint Conference On*. Network simplification through oracle learning (IEEE. Honolulu, HI, USA, 2002), pp. 2482–2486
24. J Ba, R Caruana, in *Advances in Neural Information Processing Systems (NIPS)*. Do deep nets really need to be deep? (NIPS. Montreal, Canada, 2014), pp. 2654–2662
25. Y Miao, L Jiang, H Zhang, F Metze, in *Spoken Language Technology (SLT), 2014 IEEE Workshop On*. Improvements to speaker adaptive training of deep neural networks (IEEE, South Lake Tahoe, NV, USA, 2014), pp. 165–170
26. W Chan, NR Ke, I Lane, in *INTERSPEECH*. Transferring knowledge from a RNN to a DNN (ISCA. Dresden, Germany, 2015), pp. 3264–3268
27. D Povey, A Ghoshal, G Boulianne, L Burget, O Glembek, N Goel, M Hannemann, P Motlicek, Y Qian, P Schwarz, J Silovsky, G Stemmer, K Vesely, in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop On*. The Kaldi speech recognition toolkit (IEEE, Waikoloa, HI, USA, 2011), pp. 1–4
28. H Soltau, H-K Kuo, L Mangu, G Saon, T Beran, in *INTERSPEECH*. Neural network acoustic models for the DARPA RATS program (ISCA, Lyon, France, 2013), pp. 3092–3096
29. MJF Gales, DY Kim, PC Woodland, HY Chan, D Mrva, R Sinha, SE Tranter, Progress in the CU-HTK broadcast news transcription system. IEEE Tran. Audio Speech Lang. Process. **14**(5), 1513–1525 (2006)
30. G Saon, G Zweig, B Kingsbury, L Mangu, U Chaudhari, in *EUROSPEECH*. An architecture for rapid decoding of large vocabulary conversational speech (ISCA. Geneva, Switzerland, 2003), pp. 1977–1980
31. Y Li, H Erdogan, Y Gao, E Marcheret, in *INTERSPEECH*. Incremental on-line feature space mllr adaptation for telephony speech recognition (ICSA, Denver, CO, USA, 2002), pp. 1417–1420
32. X Lei, H Lin, G Heigold, in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On*. Deep neural networks with auxiliary gaussian mixture models for real-time speech recognition (IEEE, Vancouver, Canada, 2013), pp. 7634–7638
33. M Bacchiani, in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On*. Rapid adaptation for mobile speech applications (IEEE, Vancouver, Canada, 2013), pp. 7903–7907
34. V Vanhoucke, A Senior, MZ Mao, in *NIPS Deep Learning and Unsupervised Feature Learning Workshop*. Improving the speed of neural networks on cpus (NIPS, Granada, Spain, 2011)
35. X Lei, A Senior, A Gruenstein, J Sorensen, in *INTERSPEECH*. Accurate and compact large vocabulary speech recognition on mobile devices (ISCA. Lyon, France, 2013), pp. 662–665
36. V Vanhoucke, M Devin, G Heigold, in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On*. Multiframe deep neural networks for acoustic modeling (IEEE, Vancouver, Canada, 2013), pp. 7582–7585

37. D Yu, F Seide, G Li, L Deng, in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference On*. Exploiting sparseness in deep neural networks for large vocabulary speech recognition (IEEE, Kyoto, Japan, 2012), pp. 4409–4412
38. TN Sainath, B Kingsbury, V Sindhwani, E Arisoy, B Ramabhadran, in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference On*. Low-rank matrix factorization for deep neural network training with high-dimensional output targets (IEEE, Vancouver, Canada, 2013), pp. 6655–6659
39. J Xue, J Li, Y Gong, in *INTERSPEECH*. Restructuring of deep neural network acoustic models with singular value decomposition (ISCA, Lyon, France, 2013), pp. 2365–2369
40. T He, Y Fan, Y Qian, T Tan, K Yu, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Reshaping deep neural network for fast decoding by node-pruning (IEEE, Florence, Italy, 2014), pp. 245–249
41. J Ramirez, JM Górriz, JC Segura, in *Robust Speech Recognition and Understanding*, ed. by M Grimm, K Kroschel. Voice activity detection. Fundamentals and speech recognition system robustness (I-Tech Education and Publishing, Vienna, 2007), pp. 1–22
42. J Dines, J Vepa, T Hain, in *INTERSPEECH*. The segmentation of multi-channel meeting recordings for automatic speech recognition (ICSLP, Pittsburgh, PA, USA, 2006)
43. G Saon, S Thomas, H Soltau, S Ganapathy, B Kingsbury, in *INTERSPEECH*. The IBM speech activity detection system for the DARPA RATS program (ISCA, Lyon, France, 2013), pp. 3497–3501
44. S Thomas, S Ganapathy, G Saon, H Soltau, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions (IEEE, Florence, Italy, 2014), pp. 2519–2523
45. Y LeCun, L Bottou, Y Bengio, P Haffner, Gradient-based learning applied to document recognition. Proc. IEEE. **86**(11), 2278–2324 (1998)
46. Y Miao, Kaldi+PDNN: building DNN-based ASR systems with Kaldi and PDNN. arXiv preprint arXiv:1401.6984 (2014)
47. R Caruana, S Baluja, T Mitchell, in *Advances in Neural Information Processing Systems (NIPS)*. Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation (NIPS, Denver, CO, USA, 1996), pp. 959–965
48. M Denil, B Shakibi, L Dinh, MA Ranzato, N de Freitas, in *Advances in Neural Information Processing Systems (NIPS)*. Predicting parameters in deep learning (NIPS, Lake Tahoe, NV, USA, 2013), pp. 2148–2156
49. H Nobach, C Tropea, L Cordier, JP Bonnet, J Delville, J Lewalle, M Farge, K Schneider, R Adrian, in *Springer Handbook of Experimental Fluid Mechanics*, ed. by C Tropea, AL Yarin, and JF Foss. Review of some fundamentals of data processing, vol. 1 (Springer, Heidelberg, 2007), pp. 1337–1398
50. H Soltau, G Saon, TN Sainath, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Joint training of convolutional and non-convolutional neural networks, (2014), pp. 5572–5576