JOINT OPTIMISATION OF TANDEM SYSTEMS USING GAUSSIAN MIXTURE DENSITY NEURAL NETWORK DISCRIMINATIVE SEQUENCE TRAINING

C. Zhang & P. C. Woodland

Cambridge University Engineering Dept., Trumpington St., Cambridge, CB2 1PZ U.K.
{cz277,pcw}@eng.cam.ac.uk

ABSTRACT

The use of deep neural networks (DNNs) for feature extraction and Gaussian mixture models (GMMs) for acoustic modelling is often termed a tandem system configuration and can be viewed as a Gaussian mixture density neural network (MDNN). Compared to the direct use of DNN output probabilities in the acoustic model, the tandem approach suffers from a major weakness in that the feature extraction stage and the final acoustic models are optimised separately. This paper proposes a joint optimisation approach to all the stages of the tandem acoustic model by using MDNN discriminative sequence training. A set of techniques is used to improve the training performance and stability. Experiments using the multi-genre broadcast (MGB) English data show that the proposed method produced a 6% relative lower word error rate (WER) than that of a traditional discriminatively trained tandem system. The resulting jointly optimised tandem systems are comparable in WER to hybrid DNN systems optimised using discriminative sequence training with the same number of parameters.

1. INTRODUCTION

DNNs are included in hidden Markov model (HMM) based automatic speech recognition (ASR) systems either using the "tandem" approach in which DNN produced features, such as the bottleneck (BN) features, are modelled using GMMs (i.e. BN-GMM-HMMs) [1–3], or by directly employing the DNN posterior probabilities in the HMM acoustic model in a "hybrid" configuration (i.e. DNN-HMMs) [4–7].

Although hybrid systems have recently drawn more attention, there are several reasons that make the tandem approach still of interest. First, the DNN and GMMs can be combined to form an MDNN [8], which is a general framework for modelling non-Gaussian conditional probability distributions. This is in contrast to the distributions generated by a conventional DNN acoustic model with a softmax output function that have equivalent terms to single Gaussians with a shared covariance matrix [8, 9]. Secondly, it is straightforward to improve the performance of tandem systems by applying techniques developed for GMMs to MDNN such as adaptation methods [10, 11]. Finally, tandem and hybrid systems are known to produce complementary errors, and hence significant performance improvements can be obtained by system combination [12–16].

Conventional tandem systems use GMMs independently estimated using features from a pre-trained DNN, and therefore, it is not guaranteed that the features are the most appropriate to be modelled by the selected GMM setup. To overcome this weakness, we propose a tandem system joint optimisation method that has the features and acoustic models trained together based on the minimum phone error (MPE) criterion [17, 18]. From a hybrid system view, the MDNN is initialised with a conventional tandem system, then refined by lattice based MPE sequence training. Although MPE is used in this paper as the joint optimisation criterion, other related discriminative sequence criteria [19–21] could also be used with the method. Standard GMM-HMM MPE training is first revisited, and the related parameter smoothing and variance floor methods are modified for use with stochastic gradient descent (SGD). Next, tandem system joint optimisation is investigated. A number of methods are used to improve the system performance which include linear to rectified linear unit (ReLU) [22, 23] function conversion, relative update value clipping, amplified GMM learning, and various different parameter update schemes. The final combination yields comparable performance to MPE trained DNN-HMMs. Further experiments show that the jointly optimised tandem system is useful in DNN-HMM construction and system combination. Previously, cross-entropy (CE) and maximum likelihood (ML) based tandem system joint training were studied based on MDNNs [24] and standard DNNs with a parameterised softmax output function [25, 26]. MPE training has also been applied to the task with the GMMs still optimised by the extended Baum-Welch (EBW) algorithm [27].

The rest of the paper is organised as follows. Section 2 reviews the tandem system configuration and build procedure. SGD based MPE training is discussed in Section 3, which is followed by the joint optimisation approach in Section 4. The experimental setup and results are presented in Section 5 and 6, followed by conclusions.

2. CONVENTIONAL TANDEM SYSTEMS

2.1. GMM-HMM Acoustic Models

GMMs are widely used to represent the state output distributions in HMM acoustic models for ASR. By ignoring HMM transition probabilities, the log-likelihood of an HMM state s is defined as

$$\ln p(\mathbf{z}(t)|s) = \ln \sum\nolimits_g \omega_{sg} \mathcal{N}(\mathbf{z}(t)|\mu_{sg}, \sigma_{sg}), \tag{1}$$

where $\mathbf{z}(t)$ is the input vector at time t; $\mathcal{N}(\mathbf{z}|\mu_{sg}, \sigma_{sg})$ is the g th Gaussian component with μ_{sg} and σ^2_{sg} the mean and variance vectors, and ω_{sg} is the corresponding weight. Note that by using a variance vector instead of a covariance matrix, it is assumed that all of the dimensions of $\mathbf{z}$, z_d, can be treated as independent variables.

In this paper, GMMs are trained using SGD, an unconstrained optimisation method. The following parameter transformations are used to ensure $\sigma_{sgd} > 0$ as well as ω_{sg} is positive and sums to one,

$$\sigma_{sgd} = \exp(\tilde{\sigma}_{sgd}), \tag{2}$$

Chao Zhang has been supported by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology). Supporting data for this paper is available at the repository https://doi.org/10.17863/CAM.6990.

$$\omega_{sg} = \exp(\tilde{\omega}_{sg}) / \sum\nolimits_{g'} \exp(\tilde{\omega}_{sg'}), \tag{3}$$

where $\tilde{\sigma}_{sgd}$ and $\tilde{\omega}_{sg}$ are the actual parameters updated by SGD. Therefore, the partial derivatives of a criterion $\mathcal{F}$ are

$$\frac{\partial \mathcal{F}}{\partial \tilde{\omega}_{sg}} = \sum\nolimits_t P(s,g|\mathbf{z}(t)) \cdot (1-\omega_{sg}), \tag{4}$$

$$\frac{\partial \mathcal{F}}{\partial \mu_{sgd}} = \sum\nolimits_t P(s,g|\mathbf{z}(t)) \cdot \frac{z_d(t)-\mu_{sgd}}{\sigma^2_{sgd}}, \tag{5}$$

$$\frac{\partial \mathcal{F}}{\partial \tilde{\sigma}_{sgd}} = \sum\nolimits_t P(s,g|\mathbf{z}(t)) \cdot \frac{(z_d(t)-\mu_{sgd})^2-\sigma^2_{sgd}}{\sigma^2_{sgd}}, \tag{6}$$

$$\frac{\partial \mathcal{F}}{\partial z_d(t)} = -\sum\nolimits_g P(s,g|\mathbf{z}(t)) \cdot \frac{z_d(t)-\mu_{sgd}}{\sigma^2_{sgd}}, \tag{7}$$

where $p(s,g|\mathbf{z}(t))$ is

$$\frac{\partial \mathcal{F}}{\ln p(\mathbf{z}(t)|s)} \cdot \frac{\omega_{sg}\mathcal{N}(\mathbf{z}(t)|\mu_{sg},\Sigma_{sg})}{\sum_{g'}\omega_{sg'}\mathcal{N}(\mathbf{z}(t)|\mu_{sg'},\Sigma_{sg'})}. \tag{8}$$

It is worth noting that Eqns. (1) and (5)-(7) can be rearranged and computed using the highly optimised BLAS general matrix multiplication or GEMM functions [28], to fully utilise the power of GPUs.

2.2. DNN Acoustic Feature Classifier

A DNN is a multi-layer classifier that maps an input vector $\mathbf{x}^{\text{in}}(t)$ to an output vector $\mathbf{y}^{\text{out}}(t)$ that defines the class. $\mathbf{x}^{\text{in}}(t)$ is usually formed by stacking the acoustic feature vector $\mathbf{o}(t+c)$, where c is any integer in a *context shift set* $\mathbf{c}$ that represents a time shift [29]. In a DNN layer l, the input to the nodes is called the *activation*, denoted as $\mathbf{a}_l(t)$, where $\mathbf{a}_l(t) = \mathbf{W}_l\mathbf{x}_l(t) + \mathbf{b}_l$; $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weight matrix and bias vector. $\mathbf{a}_l(t)$ is then transformed by the *activation function* to acquire the output value $\mathbf{y}_l(t) = f_l(\mathbf{a}_l(t))$. Layer l is connected with its next layer by $\mathbf{y}_l(t) = \mathbf{x}_{l+1}(t)$. If l is a hidden layer, $f_l(\cdot)$ is often either sigmoid, $\mathbf{y}_l(t) = (1+\exp(-\mathbf{a}_l(t)))^{-1}$, or ReLU, $\mathbf{y}_l(t) = \max(0, \mathbf{a}_l(t))$; otherwise l is the output layer, and $f^{\text{out}}(\cdot)$ is the softmax funtion defined by Eqn. (3), which normalises the activations into posterior probabilities associated with the HMM states. To train a DNN with SGD based on a criterion $\mathcal{F}$, $\partial\mathcal{F}/\partial\mathbf{W}_l$ and $\partial\mathcal{F}/\partial\mathbf{b}_l$ are computed by propagating $\partial\mathcal{F}/\partial\mathbf{y}^{\text{out}}(t)$ from the output layer to l using error backpropagation (EBP) [8].

2.3. Tandem System Construction

A common setup for DNN feature extraction is to use a reduced dimension hidden layer, i.e., a BN layer [2, 3], whose output vector, $\mathbf{y}^{\text{bn}}(t)$, is very compact and suitable to be directly used as GMM input features. The BN DNN training procedure is similar to normal DNN acoustic model construction with a CE objective function. Once the model is trained, the BN layer is changed to a linear activation function $\mathbf{y}^{\text{bn}}(t) = \mathbf{a}^{\text{bn}}(t)$, and the layers beyond the BN layer are removed. The use of the linear activation function keeps the discrimination ability in $\mathbf{y}^{\text{bn}}(t)$ [12], which, in this paper, is directly used as GMM input vector $\mathbf{z}(t)$ without any modification.

To construct a high performance tandem system, monophone BN-GMM-HMMs are first built using ML training, which are later expanded to initial triphone BN-GMM-HMMs. The final ML triphone system is trained using a *two-model re-estimation* method [30], with the alignments for decision tree clustering [31] produced by the well-trained initial triphone system. The system can be further refined by discriminative GMM-HMM training. It should be noted that the BN-GMM-HMMs can be viewed as MDNN-HMMs from a hybrid approach point of view.

3. MPE TRAINING FOR GMMS WITH SGD

3.1. Lattice based MPE Training

MPE is a criterion that directly optimises the expected error rate at the phone level [17, 19, 32]. MPE is defined as

$$\mathcal{F}^{\text{MPE}} = \frac{\sum_h p(\mathbf{O}|h)^\kappa P(h)\,\text{PhoneAccuracy}(r,h)}{\sum_h p(\mathbf{O}|h)^\kappa P(h)}, \tag{9}$$

where $\mathbf{O}$ is the input observation sequence, κ is the inverse language model scaling factor; r and h are the reference and hypothesis labels; PhoneAccuracy(r,h) measures the raw phone accuracy of h. To reduce the computation cost in calculating the statistics over all possible hypotheses, lattices are used as a compact representation of the hypothesis space. Further MPE details can be found in [17, 18].

3.2. Parameter Smoothing and $L2$ Regularisation

It was observed that directly applying MPE training to GMM-HMMs using the EBW algorithm caused severe over-fitting issues, which can be solved by the use of *I-smoothing* [17]. I-smoothing applies a data dependent interpolation between a discriminative criterion and the ML criterion. It takes the data availability of each Gaussian component into account with a component dependent interpolation coefficient $\tau^{\text{ML}}(s,g) = \tau^{\text{ML}}/P^{\text{ML}}(s,g|\mathbf{o}(t))$, where $P^{\text{ML}}(s,g|\mathbf{o}(t))$ is $P(s,g|\mathbf{o}(t))$ calculated with the ML criterion $\mathcal{F}^{\text{ML}} = \ln p(\mathbf{O}|s)$ at t. $\tau^{\text{ML}}(s,g)$ is viewed as a constant when differentiated. Meanwhile, if the maximum mutual information (MMI) criterion [33] is used to replace ML in smoothing, it is further referred to as a dynamic *MMI Prior*. The MMI objective function is defined as

$$\mathcal{F}^{\text{MMI}} = \ln \frac{p(\mathbf{O}|r)^\kappa P(r)}{\sum_h p(\mathbf{O}|h)^\kappa P(h)}. \tag{10}$$

In order to simulate I-smoothing and an MMI prior in the SGD framework, we use the *H-criterion* [34] to intepolate $\mathcal{F}^{\text{MPE}}$ with $\mathcal{F}^{\text{MMI}}$, with a weighting coefficient of τ^{MMI}. $\mathcal{F}^{\text{MMI}}$ is presmoothed by I-smoothing with $\tau^{\text{ML}}(s,g)$, which adds a constant τ^{ML} to $P^{\text{MMI}}(s,g|\mathbf{o}(t))$ [17]. Furthermore, the use of $L2$ *regularisation* is also investigated, which adds a term $\lambda\cdot\theta^2/2$ to the objective function, and hence is also termed $L2$ regularisation, where λ is the coefficient and θ is the parameter to penalise. The full objective function is

$$\mathcal{F}^{\text{MPE}} + \tau^{\text{MMI}}\,(\mathcal{F}^{\text{MMI}} + \tau^{\text{ML}}(s,g)\,\mathcal{F}^{\text{ML}}) + \frac{\lambda}{2}\theta^2. \tag{11}$$

3.3. Percentile based Variance Floor

In MPE training with EBW, the use of a variance floor is beneficial to stabilise training after each parameter update. In particular the use of percentile based variance floor [18] is useful. This floors variances smaller than $\sigma^2_d(p\%)$, where $\sigma^2_d(p\%)$ is the value ranked at $p\%$ among all variances of d. When applying the method in the SGD framework, it is applied after every 10 updates, to avoid causing $\sigma^2_d(p\%)$ to increase. Furthermore, to save the cost from computing the exact $\sigma_d(p\%)$ by sorting algorithms, we use $\bar{\mu}(d) + \Phi^{-1}(\frac{p}{100}) \cdot \bar{\sigma}(d)$ as an approximate value, where $\bar{\mu}(d)$ and $\bar{\sigma}(d)$ are the mean and standard deviation of all variance values of d, and $\Phi(\cdot)$ is the *standard normal cumulative distribution*.

4. TANDEM SYSTEM JOINT OPTIMISATION

Tandem system joint optimisation is performed using MDNN-HMM discriminative sequence training. This section addresses various issues with this approach.

4.1. Use of ReLU to Replace Linear Activation Functions

In practice, we found that the use of linear activation functions in the BN layer can cause a stability issue in training. This happens when the average of $\partial\mathcal{F}/\partial\mathbf{y}^{\text{bn}}(t)$ over a mini-batch moves from positive to negative, and the parameters can become stuck at a very poor solution. We solve this issue by replacing the linear function with a ReLU function. In order to avoid the information loss caused by rectification, $\mathbf{y}^{\text{bn}}(t)$ is transformed to $\tilde{\mathbf{y}}^{\text{bn}}(t) = \mathbf{y}^{\text{bn}}(t) - \mu^{\text{bn}} + 6\sigma^{\text{bn}}$ by modifying $\mathbf{b}^{\text{bn}}$, where μ^{bn} and σ^{bn} are the mean and standard deviation vectors of $\mathbf{y}^{\text{bn}}$ estimated over the training set. This guarantees 99.99966% of $\mathbf{y}^{\text{bn}}(t)$ samples are rectified without information loss, assuming $\mathbf{y}^{\text{bn}}(t)$ follows a multivariate Gaussian distribution. Meanwhile, $\mathcal{N}(\mathbf{y}^{\text{bn}}(t)|\mu_{sg}, \sigma_{sg})$ can be transformed to $\mathcal{N}(\tilde{\mathbf{y}}^{\text{bn}}(t)|\mu_{sg} - \mu^{\text{bn}} + 6\sigma^{\text{bn}}, \sigma_{sg})$ without retraining.

4.2. Relative Update Value Clipping

In SGD training, a fairly large learning rate, which is necessary for fast convergence, can sometimes cause severe performance degradation. This can be due to large inaccurate gradients being generated due to various reasons such as poor acoustic conditions and erroneous reference labels *etc*. A widely used solution to prevent the parameters changing too much in a single update is to use update value clipping [35]. However, as the standard method requires specific clipping thresholds, it is tedious to use this here as the MDNN has both GMM and DNN parameters which are rather different in range.

Here we propose a method to find a relative threshold for clipping a particular collection of parameters, Θ. Let $u_\theta[n]$ be the proposed change of θ at the nth update, the mean and standard deviation of $|u_\theta[n]|$ for $\theta \in \Theta$ are $\mu_\Theta[n]$ and $\sigma_\Theta[n]$, and then $|u_\theta[n]|$ is clipped according to a threshold of $\mu_\Theta[n] + m\,\sigma_\Theta[n]$, where m is the relative threshold. Θ can be $\{\omega_{sg}\}$, $\{\mu_{sgd}\}$, and $\{\sigma_{sgd}\}$ for all s and g, or $\mathbf{W}_l$ or $\mathbf{b}_l$ for a particular layer l.

4.3. Amplified GMM Learning

The MDNN output layer has a rather different functional form than other DNN layers, and we found that the learning rate suitable for GMM parameters is significantly larger than a normal DNN learning rate. Thus, in MDNN-HMM sequence training, different learning rates, η and $\alpha \cdot \eta$ are used for the BN DNN and GMMs separately where α is the amplification factor. To regularise training properly, the $L2$ regularisation coefficient λ for GMMs is also scaled by α.

4.4. Parameter Update Schemes

This paper investigates three different parameter update schemes for tandem system joint optimisation:

1. Update GMMs and hidden layers in an interleaved manner, which may also be useful as a regulariser; or
2. Update all parameters concurrently without restriction; or
3. Update all MDNN parameters concurrently, then update the GMMs only to make them fit the BN features better.

The various update schemes are compared in Section 6.2.

5. EXPERIMENTAL SETUP

The proposed techniques were evaluated by training systems on data from the ASRU 2015 Multi-Genre Broadcast (MGB) challenge [36]. The audio consists of seven weeks of BBC television programmes covering a wide range of genres, e.g., news, comedy, drama, sports, quiz shows, documentaries *etc*. 200 hours of data randomly selected from 2,180 shows is used as the full training set for which the difference between the sub-titles and the lightly supervised output had a phone matched error rate $< 20\%$. A 50 hour subset was evenly sampled from the 200 hour set. A trigram word level language model with a 160k word vocabulary was used in all experiments. The test set, **dev.sub**, contains 5.5 hours of audio data from 12 shows and is the official subset of the full MGB transcription development set [16, 36, 37]. The reference segmentation was used with automatic speaker clustering resulting in 8,713 utterances and 285 speaker clusters. Further details of the data preparation *etc.* were presented in [16, 37].

All experiments were conducted with HTK 3.5 [29, 38]. A 40d log-Mel filter bank (FBK) analysis was used and expanded to an 80d vector with its Δ coefficients. The inputs to all DNNs were produced with $\mathbf{c} = [-4, +4]$ [29] and normalised at the utterance level for mean and at the show-segment level for variance [16]. Triphone GMM-HMM systems with 4k/6k non-silence decision tree clustered tied-states were used for the 50h/200h training sets. The GMMs have 16 Gaussian components per state, except for the 3 silence states, which have 32 Gaussian components per state. The DNNs were built with a hidden layer structure of 720×1000^5 for acoustic modelling and $720 \times 1000^4 \times 39 \times 1000$ for feature extraction, and the BN layer size was 39. Their output layer sizes are 4k/6k, according to the number of GMM-HMM tied-states. CE DNN training was performed using our previous setup [37]. DNN layer MPE training used a fixed learning rate of 1.0×10^{-4} and a relative update value clipping threshold of $m = 3$, whereas a threshold of $m = 9$ for the GMMs in joint optimisation.

6. EXPERIMENTAL RESULTS

6.1. GMM MPE Training

EBW and SGD based GMM-only training were performed on the 50h training set, and compared in Fig. 1. Both EBW and SGD MPE training started from a baseline ML BN-GMM-HMM system with a WER of 38.4%. EBW MPE training with an MMI prior, I-smoothing, and a percentile based variance floor, can reduce the WER to 36.1% after 4 iterations. Unlike EBW, SGD based MPE GMM training with a learning rate of 5.0×10^{-3} and no regularisation can also reduce WER, though the results fluctuate over 8 epochs.

Next, the smoothing method (described in Section 3.2) and $L2$ regularisation were added. τ^{MMI} and τ^{ML} per frame were set to 3.0×10^{-5} and 2.0×10^{-6}, and λ was 4.0×10^{-4}. It can be seen that both smoothing and $L2$ regularisation help stabilise and improve the performance. When the percentile based variance floor is finally applied, SGD based MPE training consistently reduced the WER with every epoch and gave a WER of 35.8% after the 4th epoch. It can be seen from Fig. 1 that the final SGD based MPE GMM training works at least as well as the EBW based method.

6.2. Joint Sequence Training

From the experiments in Section 6.1, the learning rates suitable for the GMM layer are 50 times larger than for the hidden layers. Here,

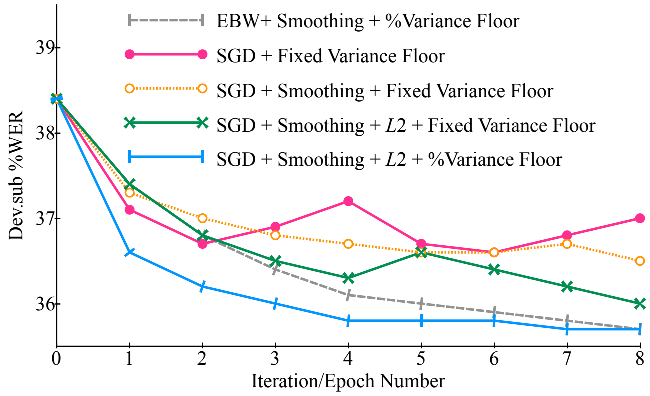

Fig. 1. *% WER on dev.sub for GMM-HMM MPE training (50h set).*

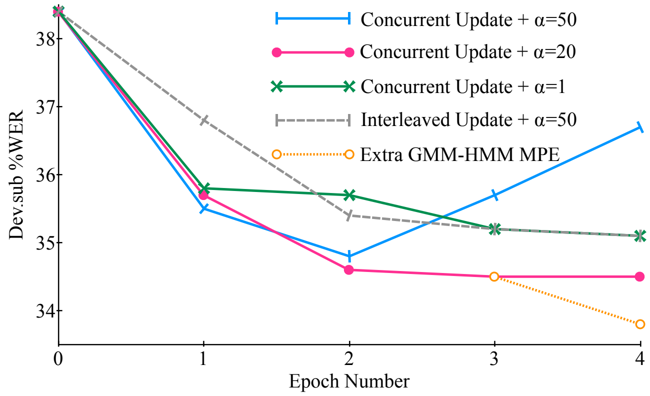

Fig. 2. *% WER on dev.sub for joint tandem MPE training (50h set).*

different parameter update schemes and GMM learning amplification factors were compared with $\eta = 1.0 \times 10^{-4}$ and $\lambda = 4.0 \times 10^{-5}$. For concurrent updates, it can be seen that $\alpha = 1$ and 20 gave a consistent WER reduction across epochs, and a WER of 34.5% was obtained when $\alpha = 20$. If α is further increased to 50, a WER of 34.6% was found at the 2nd epoch, but severe over-fitting occurred thereafter. In the interleaved update scheme, the GMM layer was updated first, and a WER of 35.1% after updating GMMs and hidden layers each for two epochs was achieved. If three epochs of the concurrent update and one epoch of SGD based GMM-HMM MPE training were applied, both with $\alpha = 20$, the best 50h SI tandem system WER of 33.8% was obtained. Therefore, this combined scheme is adopted in the following experiments.

6.3. Further Experiments

Table 1 contains the results on the 50h training set. $\mathrm{H}_0^{50\mathrm{h}}$ is the baseline CE DNN-HMM system, which has a WER of 3.9% relative lower than the ML trained BN-GMM-HMM system. After MPE training, the WER is further reduced by 7.3% relative and $\mathrm{H}_1^{50\mathrm{h}}$ is produced. $\mathrm{T}_2^{50\mathrm{h}}$ outperformed $\mathrm{H}_1^{50\mathrm{h}}$ since tandem system joint optimisation gives a larger improvement than DNN-HMM MPE training. Note that $\mathrm{H}_1^{50\mathrm{h}}$ and $\mathrm{T}_2^{50\mathrm{h}}$ also have similar numbers of parameters (8.7M and 8.8M respectively). By using the alignments from $\mathrm{T}_2^{50\mathrm{h}}$ for DNN-HMM training, a 0.6% absolute WER reduction was obtained, which was slightly better than using alignments produced by the MPE DNN-HMM system $\mathrm{H}_1^{50\mathrm{h}}$. If the training targets were derived from the tied-states of $\mathrm{T}_2^{50\mathrm{h}}$, another 0.4% absolute WER reduction was acquired, as $\mathrm{T}_2^{50\mathrm{h}}$ decision trees were constructed on BN features that are more suitable for clustering DNN output targets [9].

ID	System	WER%
$\mathrm{T}_0^{50\mathrm{h}}$	ML BN-GMM-HMMs	38.4
$\mathrm{T}_1^{50\mathrm{h}}$	MPE BN-GMM-HMMs	36.1
$\mathrm{T}_2^{50\mathrm{h}}$	MPE MDNN-HMMs	33.8
$\mathrm{H}_0^{50\mathrm{h}}$	CE DNN-HMMs	36.9
$\mathrm{H}_1^{50\mathrm{h}}$	MPE DNN-HMMs	34.2
$\mathrm{H}_2^{50\mathrm{h}}$	MPE DNN-HMMs+$\mathrm{H}_1^{50\mathrm{h}}$ align.	33.7
$\mathrm{H}_3^{50\mathrm{h}}$	MPE DNN-HMMs+$\mathrm{T}_2^{50\mathrm{h}}$ align.	33.6
$\mathrm{H}_4^{50\mathrm{h}}$	MPE DNN-HMMs+$\mathrm{T}_2^{50\mathrm{h}}$ align. & tree	33.2

Table 1. *%WER on dev.sub for various 50h systems.*

The proposed approach was then validated on the larger 200h training set. All MDNN-HMM MPE training parameters are the same as for the 50h systems, except for the learning rate of 2.5×10^{-5}. Based on the results in Table 2, the jointly trained MPE MDNN-HMMs, $\mathrm{T}_1^{200\mathrm{h}}$, is comparable to the MPE DNN-HMMs, $\mathrm{H}_1^{200\mathrm{h}}$, both in performance and size, which is consistent with the 50h system results. Finally, the use of traditional GMM-HMM techniques, such as maximum likelihood linear regression (MLLR) [10] and joint decoding [14, 15], was studied for MPE MDNN-HMMs. With test-time unsupervised MLLR adaptation based on the hypotheses produced by $\mathrm{T}_1^{200\mathrm{h}}$, the SD system $\mathrm{T}_2^{200\mathrm{h}}$ outperformed the SI system $\mathrm{T}_1^{200\mathrm{h}}$ by a 4.0% relative WER reduction. Joint decoding was used to combine $\mathrm{H}_2^{200\mathrm{h}}$ with either $\mathrm{T}_1^{200\mathrm{h}}$ or $\mathrm{T}_2^{200\mathrm{h}}$, and the resulting systems $\mathrm{J}_1^{200\mathrm{h}}$ and $\mathrm{J}_2^{200\mathrm{h}}$ outperformed their constituent systems which showed the complementarity between DNN-HMMs and MDNN-HMMs.

ID	System	WER%
$\mathrm{T}_0^{200\mathrm{h}}$	ML BN-GMM-HMMs	33.7
$\mathrm{T}_1^{200\mathrm{h}}$	MPE MDNN-HMMs	29.8
$\mathrm{T}_2^{200\mathrm{h}}$	MPE MDNN-HMMs+MLLR	28.6
$\mathrm{H}_0^{200\mathrm{h}}$	CE DNN-HMMs	31.9
$\mathrm{H}_1^{200\mathrm{h}}$	MPE DNN-HMMs	29.6
$\mathrm{H}_2^{200\mathrm{h}}$	MPE DNN-HMMs+$\mathrm{T}_1^{200\mathrm{h}}$ align. & tree	29.0
$\mathrm{J}_1^{200\mathrm{h}}$	$\mathrm{T}_1^{200\mathrm{h}} \otimes \mathrm{H}_2^{200\mathrm{h}}$ joint decoding	28.3
$\mathrm{J}_2^{200\mathrm{h}}$	$\mathrm{T}_2^{200\mathrm{h}} \otimes \mathrm{H}_2^{200\mathrm{h}}$ joint decoding	27.4

Table 2. *%WER on dev.sub for various 200h systems.*

7. CONCLUSIONS

In this paper, conventional EBW based GMM-HMM MPE training is extended to the SGD framework and applied to MDNN discriminative sequence training for the joint optimisation of tandem systems that model features produced by a DNN. A set of methods are modified or proposed to improve the training performance, which results in an average of an 11.8% relative reduction in WER over traditional ML tandem systems. The refined tandem system is comparable to MPE trained hybrid system both in performance and number of parameters, and is furthermore useful for hybrid system construction and system combination. The jointly trained tandem system can also benefit from existing GMM based approaches, such as MLLR, which can further reduce the system WER.

8. REFERENCES

[1] H. Hermansky, D.P.W. Ellis, & S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems", *Proc. ICASSP*, Istanbul, 2000.

[2] F. Grézl, M. Karafiát, S. Kontár, & J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings", *Proc. ICASSP*, Honolulu, 2007.

[3] Z. Tüske, M. Sundermeyer, R. Schlüter, & H. Ney, "Context-dependent MLP for LVCSR: Tandem, hybrid or both?", *Proc. Interspeech*, Portland, 2012.

[4] H.A. Bourlard & N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[5] F. Seide, G. Li, & D. Yu, "Conversational speech transcription using context-dependent deep neural networks", *Proc. Interspeech*, Florence, 2011.

[6] G.E. Dahl, D. Yu, L. Deng, & A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 30–42, 2012.

[7] G.E. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, & B. Kingsbury, "Deep neural networks for acoustic modelling in speech recognition", *IEEE Signal Processing Magazine*, pp. 2–17, 2012.

[8] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, NY, USA, 2006.

[9] C. Zhang & P.C. Woodland, "Standalone training of context-dependent deep neural network acoustic models", *Proc. ICASSP*, Florence, 2014.

[10] C.J. Leggetter & P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[11] J.-L. Gauvain & C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[12] K.M. Knill, M.J.F. Gales, S.P. Rath, P.C. Woodland, C. Zhang, & S.-X. Zhang, "Investigation of multilingual deep neural networks for spoken term detection", *Proc. ASRU*, Olomouc, 2013.

[13] P. Swietojanski, A. Ghoshal, & S. Renals, "Revisiting hybrid and GMM-HMM system combination techniques", *Proc. ICASSP*, Vancouver, 2013.

[14] X. Liu, F. Flego, L. Wang, C. Zhang, M.J.F. Gales, & P.C. Woodland, "The Cambridge University 2014 BOLT conversational telephone Mandarin Chinese LVCSR system for speech translation", *Proc. Interspeech*, Dresden, 2015.

[15] H. Wang, A. Ragni, M.J.F. Gales, K.M. Knill, P.C. Woodland, & C. Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages", *Proc. Interspeech*, Dresden, 2015.

[16] P.C. Woodland, X. Liu, Y. Qian, C. Zhang, M.J.F. Gales, P. Karanasou, P. Lanchantin, & L. Wang, "Cambridge University transcription systems for the multi-genre broadcast challenge", *Proc. ASRU*, Scottsdale, 2015.

[17] D. Povey & P.C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training", *Proc. ICASSP*, Orlando, 2002.

[18] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, University of Cambridge, Cambridge, UK, 2003.

[19] M. Gibson & T. Hain, "Hypothesis spaces for minimum Bayes risk training in large vocabulary speech recognition", *Proc. ICASSP*, Pittsburgh, 2006.

[20] D. Povey & B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training", *Proc. ICASSP*, Honolulu, 2007.

[21] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, & K. Visweswariah, "Boosted MMI for model and feature-space discriminative training", *Proc. ICASSP*, Las Vegas, 2008.

[22] V. Nair & G.E. Hinton, "Rectified linear units improve restricted Boltzmann machines", *Proc. ICML*, Haifa, 2010.

[23] X. Glorot, A. Bordes, & Y. Bengio, "Deep sparse rectifier networks", *Proc. AISTATS*, Lauderdale, 2011.

[24] E. Variani, E. McDermott, & G. Heigold, "A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture", *Proc. ICASSP*, Brisbane, 2015.

[25] Z. Tüske, M. Sundermeyer, R. Schlüter, & H. Ney, "Integrating Gaussian mixtures into deep neural networks: Softmax layer with hidden variables", *Proc. ICASSP*, Brisbane, 2015.

[26] Z. Tüske, P. Golik, R. Schlüter, & H. Ney, "Speaker adaptive joint training of Gaussian mixture models and bottleneck features", *Proc. ICASSP*, Scottsdale, 2015.

[27] M. Paulik, "Lattice-based training of bottleneck feature extraction neural networks", *Proc. Interspeech*, Lyon, 2013.

[28] J.J. Dongarra, J. du Croz, & I. Duff, "A set of level 3 basic linear algebra subprograms", *IEEE Transactions on Mathematical Software*, vol. 16, pp. 1–17, 1990.

[29] C. Zhang & P.C. Woodland, "A general artificial neural network extension for HTK", *Proc. Interspeech*, Dresden, 2015.

[30] M. Gales, *Model-based Techniques for Robust Speech Recognition*, Ph.D. thesis, University of Cambridge, Cambridge, UK, 1995.

[31] S.J. Young, J.J. Odell, & P.C. Woodland, "Tree-based state tying for high accuracy acoustic modelling", *Proc. HLT*, Plainsboro, 1994.

[32] J. Kaiser, B. Horvat, & Z. Kačič, "A novel loss function for the overall risk criterion based discriminative training of HMM models", *Proc. Interspeech*, Beijing, 2000.

[33] P. Brown, *The Acoustic-Modeling Problem in Automatic Speech Recognition*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1987.

[34] P.S. Gopalakrishnan, D. Kanevsky, A. Nádas, & D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems", *IEEE Transactions on Information Theory*, vol. 37, pp. 107–113, 1991.

[35] T. Mikolov, *Statistical Language Models based on Neural Networks*, Ph.D. thesis, Brno University of Technology, Brno, Czech Republic, 2012.

[36] P. Bell, M.J.F. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Wester, & P.C. Woodland, "The MGB challenge: Evaluating multi-genre broadcast media transcription", *Proc. ASRU*, Scottsdale, 2015.

[37] C. Zhang & P.C. Woodland, "DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions", *Proc. ICASSP*, Shanghai, 2016.

[38] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, A. Ragni, V. Valtchev, P. Woodland, & C. Zhang, *The HTK Book (for HTK version 3.5)*, Cambridge University Engineering Department, 2015.