Differentiable pooling for unsupervised speaker adaptation

DIFFERENTIABLE POOLING FOR UNSUPERVISED SPEAKER ADAPTATION

Pawel Swietojanski and Steve Renals

Centre for Speech Technology Research, University of Edinburgh, Edinburgh EH8 9AB

{p.swietojanski, s.renals}@ed.ac.uk

ABSTRACT

This paper proposes a differentiable pooling mechanism to perform model-based neural network speaker adaptation. The proposed technique learns a speaker-dependent combination of activations within pools of hidden units, was shown to work well unsupervised, and does not require speaker-adaptive training. We have conducted a set of experiments on the TED talks data, as used in the IWSLT evaluations. Our results indicate that the approach can reduce word error rates (WERs) on standard IWSLT test sets by about 5–11% relative compared to speaker-independent systems and was found complementary to the recently proposed learning hidden units contribution (LHUC) approach, reducing WER by 6–13% relative. Both methods were also found to work well when adapting with small amounts of unsupervised data – 10 seconds is able to decrease the WER by 5% relative compared to the baseline speaker independent system.

***Index Terms*—** Differentiable pooling, Speaker Adaptation, Deep Neural Networks, TED, LHUC

1. INTRODUCTION

Acoustic modelling based on neural network estimates of probability model scores [1,2] have resulted in significant reductions in word error rate (WER), compared with discriminatively trained Gaussian mixture model (GMM) based systems [3]. Context-dependent deep neural network (DNN) acoustic models learn layered non-linear feature extraction from training data. Direct adaptation to the characteristics of a particular speaker, channel, or acoustic environment has been demonstrated to improve the accuracy of DNN acoustic models [4,5,6,7,8,9], and some GMM-based adaptation techniques have been used to effectively adapt the feature space of a DNN system, in particular constrained (feature-space) MLLR, referred to as fMLLR [10].

Adaptation techniques for neural network models operate in three main ways. Feature transform approaches aim to normalise the feature space according to the speaker. The dominant technique, for both DNN-based and GMM-based systems, is fMLLR, in which a linear transform of the feature space is estimated to maximise the likelihood of the adaptation data. This is based on the first pass recognition: in the case of DNN acoustic models a parallel GMM-based system is used to estimate the transform. fMLLR has been shown to reduce the WER significantly when used with DNN acoustic models [3, 11]. A direct linear transformation of the feature space, trained as an additional layer of a neural network acoustic model, has also been proposed [4, 5, 6], but experiments have shown fMLLR to be more effective.

A second approach uses auxiliary features, in which the acoustic feature vectors are augmented with additional speaker-specific features computed for each speaker at both training and test stages. i-vectors [12] have proven to be the most successful additional features, in terms of WER, and it has been shown that they are complementary to fMLLR feature space adaptation [13].

Finally, model-based adaptation involves adapting the weights of a DNN directly [9]. Adapting the weights of a DNN on a per speaker basis, can lead to extremely speaker-dependent parameter sets, and estimation issues come to the fore when estimating such parameter sets on small amounts of adaptation data [8]. A number of techniques have been proposed in which small subsets of the network parameters are adapted [14, 15, 16, 17], or the weight matrices are factorised [18] to reduce the size of speaker-dependent parameters.

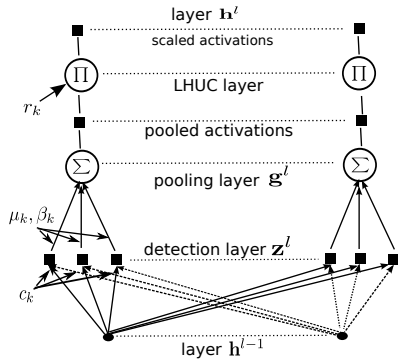
This paper is based on the recently introduced compact (unsupervised) DNN model-based adaptation technique via learning hidden unit contributions (LHUC) in a speaker-specific way [19, 20]. In LHUC, an amplitude parameter is introduced for each hidden unit, tied on a per-speaker basis, and estimated in a supervised [19] and/or unsupervised [20] way using first-pass alignments. This technique resulted in significant reductions in WER, when tested using the TED talks datasets from the IWSLT evaluation, and was complementary to fMLLR [20]. In this paper we extend the approach to parameterised, differentiable pooling functions applied to the hidden layers, which are allowed to learn localised feature combinations within pooling regions, in a speaker-dependent manner.

2. POOLING

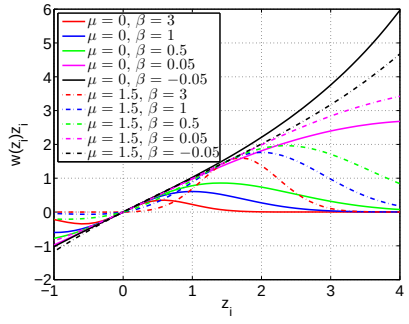
Pooling is an approach which combines a set of hidden unit outputs into a summary statistic, first used in computer vision to combine spatially local features [21]. Average pooling has been used in convolutional neural networks [22, 23] and max pooling has been used in the context of hierarchies of features for object recognition [24, 25, 26]. More generally, this approach to reducing the dimensionality of hidden layers by the interpolation or selection of some subsets of hidden unit activations has become well investigated beyond computer vision, and the $\max$ operator has been interpreted as a way to learn piecewise linear activation functions, referred to as the maxout model [27]. Maxout models have been widely investigated for both fully-connected [28,29,30] and convolutional [31,32] DNN acoustic models.

Max-pooling performs a one-from-K selection, and hence does not allow hidden unit outputs to be interpolated, or their combination to be learned within a pool. There have been a number of approaches to pooling with differentiable operators – *differentiable pooling* – a notion introduced by Zeiler and Fergus [33] in the context of constructing unsupervised feature extraction for support vector machines in computer vision tasks. In L_p-norm pooling [34] the sufficient statistic is the p-norm of the hidden unit activations in the pool; when $p = 1$ this corresponds to averaging, and when $p = \infty$, it corresponds to maxout. L_p-norm pooling was recently

This research was supported by EPSRC Programme Grant EP/I031022/1, *Natural Speech Technology*.



(a)



(b)

Fig. 1. A scheme of LHUC+DiffP layer where mth speaker dependent parameters for k-th pooling unit are $\theta_k^m = \{\mu_k, \beta_k, r_k\}$. b) Example activations $u_k(z_i)z_i$ as a function of z_i for various combinations of μ_k and β_k.

applied within the context of a convolutional neural network acoustic model [35], where it did not reduce WER, and as an activation function in a fully-connected DNN [36], where was reported to improve over maxout models.

The order (p) of L_p-norm poolers can also be learnt from data [37], separately for each of the pooling units. However, experimental validation of whether this improves over fixed order L_p units, or whether learning p in a speaker-dependent fashion will act as an effective approach to speaker adaptation is yet to be investigated.

3. DIFFERENTIABLE POOLING DNNS

We use a Gaussian kernel to estimate the pooling weights, giving a low-dimensional set of speaker parameters that might be estimated using adaptation data.

A feed-forward neural network acoustic model estimates the posterior distribution of tied state s_t given acoustic signal $\mathbf{x}_t$, $P(s_t|\mathbf{x}_t)$, using multiple layers of non-linear deterministic transformations. Each such layer is composed of a number of (hidden) units computed as a weighted sum of units from the previous layer (or the inputs) which is then followed by non-linear operation ϕ. This operation for the j-th unit in the l-th layer is written as:

$$z_j^l\left([h_1^{l-1}, \dots, h_{N_l}^{l-1}]\right) = c_j^l \cdot \phi\left(b_j^l + \sum_i^{N_l} w_{ij}^l h_i^{l-1}\right), \quad (1)$$

where h^{l-1} is the input to the lth hidden layer which is parameterised by weight matrix $\mathbf{W}^l \in \mathbb{R}^{N^l \times M^l}$ and bias vector $\mathbf{b} \in \mathbb{R}^{M^l}$. The non-linear activation function, ϕ, is a sigmoid in this paper. c_j is a hidden unit scaling factor, or amplitude, which usually is set to 1.0 as the weighting for speaker-independent models is performed by the weight matrices of the upper layers.

Given (1), the pooling operation is defined as an weighted average over a set of hidden units, G, where the k-th pooling unit at lth layer, $g_k^l(\cdot)$, is expressed as:

$$g_k^l\left(\{z_j^l\}_{j\in G_k}\right) = \sum_{i\in G_k} u_k^l(z_i^l) z_i^l. \quad (2)$$

The pooling weights u_k^l are normalised to sum to one within a pooling region G_k (3) and each weight u is coupled with the corresponding value of z by a Gaussian kernel (4) (one per pooling unit) parameterised by $\theta_k^l = \{\mu_k^l, \beta_k^l\}$ – mean and precision, respectively:

$$u_k^l(z_i^l) = \frac{v_k^l(z_i^l)}{\sum_{i'\in G_k} v_k^l(z_{i'}^l)}, \quad (3)$$

$$v_k^l(z_i^l) = \exp\left(-\frac{\beta_k^l}{2}\left(z_i^l - \mu_k^l\right)^2\right). \quad (4)$$

This formulation allows generalised poolings to be learned – from average ($\beta \to 0$) to max ($\beta \to \infty$) – separately for each pooling unit g_k within a model. We can learn the parameters θ_k jointly with the other parameters using error back-propagation, and we can refine those interpolation coefficients to effectively adapt the model for unseen speakers (Sections 4 and 5).

Fig 1 b) shows examples of a single weighted activation $u(z_i)z_i$ (using (3) and (4)) as a function of the range of z_i given μ_k and β_k. One can argue that learning θ_k would have limited effect, as the range of hidden units z_j is constrained by a sigmoid non-linearity to $[0, 1]$. The ranges in which pooling units operate (c_k in (1)) are learned similarly to [38, 20], but c_k is tied per pooling region G_k rather than being optimised separately for each hidden unit (experimental validation is in Section 5.1).

A differentiable pooling layer is illustrated in Fig 1 a): the scaling parameter r_k is used only for some speaker dependent (SD) models, and as such can be ignored when considering a speaker independent (SI) model. In this case the model is of L hidden layers, parameterised by $\mathbf{\Theta}^{SI} = \left\{\{\mathbf{W}^l, \mathbf{b}^l, \mathbf{c}^l, \mu^l, \beta^l\}_{l=1}^L, \mathbf{U}, \mathbf{b}\right\}$, where $\mathbf{U}, \mathbf{b}$ denotes a connection matrix and biases of the output layer. In the following we derive the gradients to update the pooling parameters μ and β. Gradients for the remaining parameters are computed by standard back-propagation.

We optimise the per-frame negative log posterior probability cost function $\mathcal{F}(\mathbf{\Theta}^{SI}) = -\sum_t^T \log P(s_t|\mathbf{x}_t; \mathbf{\Theta}^{SI})$ over T training examples defined as a pair – ground-truth tied-state and associated acoustic vector: $\{s_t, \mathbf{x}_t\}$. If $\mathcal{E}^l$ denotes an error signal passed to the l-th pooling layer $\mathbf{g}^l$, then the gradients to learn parameters of the k-th pooling unit $\theta_k^l = \{\mu_k^l, \beta_k^l\}$ can be obtained using a multipath chain rule:

$$\frac{\partial \mathcal{E}_k^l}{\partial \theta_k^l(n)} = \frac{\partial \mathcal{E}_k^l}{\partial g_k^l} \sum_{i\in G_k} \frac{\partial g_k^l}{\partial u_k^l(z_i^l)} \sum_{i'\in G_k} \frac{\partial u_k^l(z_i^l)}{\partial v_k^l(z_{i'}^l)} \frac{\partial v_k^l(z_{i'}^l)}{\partial \theta_k^l(n)}, \quad (5)$$

where $\partial \mathcal{E}_k^l / \partial g_k^l$ represents a joining point of the standard chain rule up to layer l, and the particular derivatives are given by:

$$\frac{\partial g_k^l}{\partial u_k^l(z_i^l)} = z_i^l, \quad (6)$$

$$\frac{\partial u_k^l(z_i^l)}{\partial v_k^l(z_i^l)} = \Big(\sum_{m\in G_k} v_k^l(z_m^l)\Big)^{-1}\left(1 - u_k^l(z_i^l)\right), \quad (7)$$

$$\frac{\partial u_k^l(z_i^l)}{\partial v_k^l(z_{i'}^l)} = \Big(\sum_{m \in G_k} v_k^l(z_m^l) \Big)^{-1} \Big(-u_k^l(z_{i'}^l) \Big), \quad (8)$$

$$\frac{\partial v_k^l(z_i^l)}{\partial \mu_k^l} = \beta_k^l (z_i^l - \mu_k^l) v_k^l(z_i^l), \quad (9)$$

$$\frac{\partial v_k^l(z_i^l)}{\partial \beta_k^l} = -\frac{1}{2} \Big(z_i^l - \mu_k^l \Big)^2 v_k^l(z_i^l). \quad (10)$$

To pass error signals through the differentiable pooling layer, one needs to take into account that $u_k^l(z_i^l)$, for our choice of kernel, depends on the activation z (4). Using identities from (6),(7) and (9) and noticing that $\partial v_k^l(z_i^l)/\partial z_i^l = -\partial v_k^l(z_i^l)/\partial \mu_k^l$ one can derive the chain rule for the unit z_i^l of the G_k region as:

$$\begin{aligned} \frac{\partial g_k^l}{\partial z_i^l} &= \frac{\partial g_k^l}{\partial u_k^l(z_i^l)} \frac{\partial u_k^l(z_i^l)}{\partial v_k^l(z_i^l)} \frac{\partial v_k^l(z_i^l)}{\partial z_i^l} + u_k^l(z_i^l) \\ &= -\frac{\partial g_k^l}{\partial u_k^l(z_i^l)} \frac{\partial u_k^l(z_i^l)}{\partial v_k^l(z_i^l)} \frac{\partial v_k^l(z_i^l)}{\partial \mu_k} + u_k^l(z_i^l) \end{aligned} \quad (11)$$

4. SPEAKER-DEPENDENT DNN

Within this paper we experiment with three types of speaker-dependent models, i) speaker-dependent differentiable pooling (DiffP), ii) LHUC-based adaptation that learns hidden units contributions on a per-speaker basis [20], and iii) speaker-adaptive training using feature-space speaker normalisation based on fMLLR [10].

All three methods require first-pass decoding to obtain adaptation targets to either estimate fMLLR transforms for unseen speakers or to perform DNN speaker-dependent parameter update. For DiffP and LHUC we carry out adaptation minimising $\mathcal{F}(\mathbf{\Theta}^m) = -\sum_t^{T^m} \log P(s_t|\mathbf{x}_t^m; \mathbf{\Theta}^m)$ where $\mathbf{\Theta}^m$ denotes parameters for the m-th speaker (one can also regularise using the KL divergence between SI and SD models [8]). For fMLLR the model parameters are shared across the speakers, $\mathbf{\Theta}_{fMLLR}^m = \mathbf{\Theta}^{SI}$, and the model learns a conditional distribution of targets given linearly-transformed features using speaker-adaptive training.

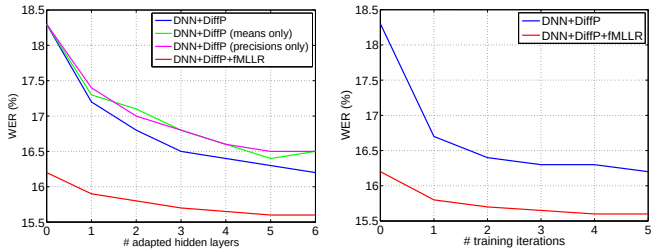
While adapting DiffP models, we start with pooling parameters learned in a speaker-independent manner and update them separately for each of talker, $\mathbf{\Theta}_{DiffP}^m = \{\{\mu_k^l, \beta_k^l\}_{k=1}^{R_l}\}_{l=1}^L$, where $R_l = \sum_l (M^l - G^l)/F^l + 1$. If F denotes pooling shift and G the pooling size, the DiffP model has at most $2 \cdot R_l$ speaker-dependent parameters. For the models presented in this work, if not stated otherwise, $G = F = 3$.

LHUC-based adaptation inserts additional scaling parameters $\mathbf{r}_m$ at the output of the l-th layer, $\mathbf{\Theta}_{LHUC}^m = \{\mathbf{r}_m^l\}_{l=1}^L$. When combined with DiffP models, this scaling occurs after pooling, in which case the dimensionality of LHUC SD parameters is $\sum_l R_l$; for the DNN+LHUC-only models it stays the same as the number of hidden units in $\mathbf{z}$ layer, $\sum_l M^l$. In the latter case, $\mathbf{r}_m^l$ is equivalent to $\mathbf{c}^l$ in (1). $\mathbf{r}_m^l$ is re-parameterised during optimisation so it stays in the range [0,2] – this helps to provide the expected weighted input to the upper layer and constraints the modelling capacity which is beneficial when adapting with noisy targets [20].

Fig 1 a) depicts a complete layer composed of both DiffP and LHUC SD parts. Such layers can be stacked on each other to form deeper structures.

5. EXPERIMENTS

We carried out experiments using a corpus of publicity available TED talks (`http://www.ted.com`) following the IWSLT ASR evaluation protocol [39] (`http://iwslt.org`). We use baseline systems described in [20,7]. The training data consisted of 143 hours of speech (813 talks). We present results on three predefined IWSLT test sets: `dev2010`, `tst2010`, and `tst2011` containing 8, 11, and 8 ten-minute talks respectively.



(a) (b)

Fig. 2. WER(%) on `tst2010` as a function of a) #layers with SD pooling regions and b) number of adaptation iterations

The baseline acoustic model was a DNN with 6 hidden layers and 2048 units per layer together with 12000 tied state outputs. The input features had a dimension of 351: PLP-12 (including C0), with first and second derivatives, with ± 4 frames of context. DiffP models have a detection layer of size 3525 which is then reduced to 1175 hidden units by pooling ($F = 3$, $G = 3$). Hyper-parameters for training DiffP models are the same as for the DNN and are described in detail in [20]. All models are adapted with a large learning rate of 0.8, which is the same for each SD parameter. Experiments were carried out using the Kaldi speech recognition toolkit [40] and DNNs were trained using the PyLearn2 library [41].

We use `tst2010` to perform more detailed analyses. A collective summary of results on `dev2010`, `tst2010`, and `tst2011` are reported at the end this section. All the adaptation experiments, unless explicitly stated otherwise, were performed unsupervised.

5.1. Speaker-independent DiffP models

The SI DiffP models operate on non-overlapping groups of size $G = 3$. This hyper-parameter was optimised on the `dev2010` set giving WERs of 19.2%, 19.0%, and 19.4% for $G = 2$, $G = 3$, and $G = 5$, respectively.

As have already been mentioned, the amplitude parameters $\mathbf{c}^l$ in detection layer $\mathbf{z}^l$ (eq. (1)) were optimised jointly with μ and β. WERs using SI DiffP models without learning amplitudes were about 0.5% absolute worse than those reported for DiffP (SI) in Table 1. After the SI model is trained, $\mathbf{c}^l$ stays speaker-independent. For the LHUC experiments using DiffP, a second amplitude parameter is inserted after the pooling layer (Section 4).

The final results for SI DiffP models with the above modifications are reported in the second row of Table 1 – where they show a consistent absolute reduction in WER of about 0.3%, compared to the baseline DNN models.

5.2. Speaker-dependent DiffP models

To investigate how many and which layers benefit most from making pooling regions speaker-dependent, we progressively adapted layers from the bottom (closest to the input) to the top (Fig 2 a)). The three bottom-most layers had the biggest impact on accuracy; however, further adaptation of the higher pooling layers did bring additional gains. The plot also shows the effect of adapting means or precisions

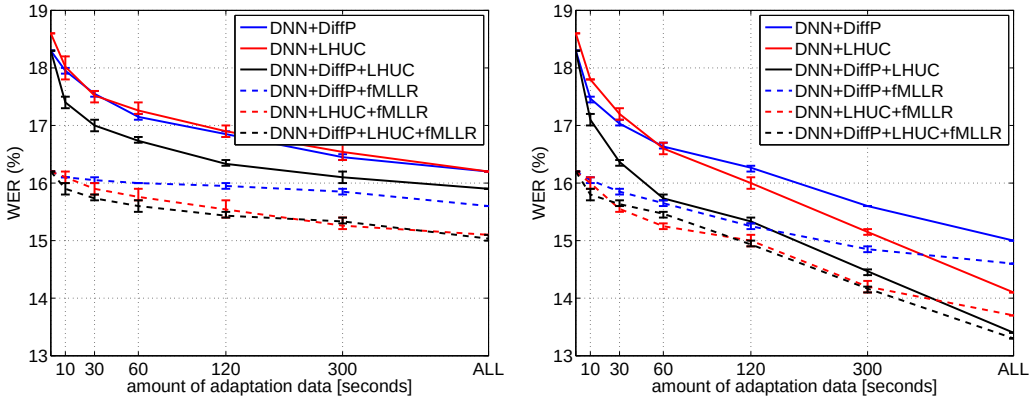


Fig. 3. WER(%) for different amounts of adaptation data a) unsupervised and b) oracle experiments on `tst2010`

Table 1. WER(%) results on IWSLT12 TED evaluation sets. Relative improvements are given in parentheses w.r.t DiffP (SI) model.

Model	`dev2010`	`tst2010`	`tst2011`
DNN [20]	19.3	18.6	15.2
DiffP (SI)	19.0	18.3	14.9
DiffP (SI) + fMLLR	18.1 (-4.7)	16.2 (-11.4)	13.5 (-9.4)
DiffP	18.1 (-4.7)	16.2 (-11.4)	13.9 (-6.8)
DiffP+LHUC	17.9 (-5.7)	15.9 (-13.1)	13.7 (-8.1)
DiffP+fMLLR	17.7 (-6.8)	15.6 (-14.7)	13 (-12.8)
DiffP+LHUC+fMLLR	17.5 (-7.9)	15.0 (-18)	12.8 (-14)

only, showing that both have a similar effect on WER – adapting both gives the lowest WER.

Fig 2 b) plots WER against the number of adaptation iterations, using the same alignments obtained from the first pass decoding lattices. Similar to LHUC [20], most of the decrease in WER was obtained after one iteration of adaptation; The other observation from both plots of Fig 2 is the limited complementarity of DiffP adaptation and fMLLR-based speaker normalisation. The observed WER reductions over the fMLLR-only system is 0.4%–0.6% absolute (2.2%–3.7% relative). This contrasts with LHUC+fMLLR [20] which resulted in 1.0–1.2% absolute WER reduction compared with fMLLR-only.

In the following experiments we carry out adaptation on all layers and adapt for three iterations. Following the same system configuration as in our previous experiments [20], we investigated how the amount of adaptation data affects WER by randomly selecting adaptation utterances to give totals of 10s, 30s, 60s, 120s, and 300s of speaker-specific adaptation data per talker. To improve reliability, we repeated these scenarios 5 times, and we report the average WERs in Fig 3 (left). We observe that 10s of unsupervised adaptation data decreases the WER of the large speaker independent model by 3% relative for either DiffP– or LHUC–only adaptation. This is further improved when adapting with more data to 5.4% relative with 30s and 7% relative with 60s. A full two-pass decoding yields a WER of 16.2% (11.4% relative improvement w.r.t DiffP (SI) model) – which is comparable to the result obtained with speaker adaptive fMLLR training. Combining DiffP+LHUC for the SI model further decreases the WER by 5%, 7%, and 9% relative for 10s, 30s, and 60s respectively – around 2% relative gain on top of those methods used independently. Combining all three adaptation techniques – DiffP+LHUC+fMLLR – decreases the final WER on `tst2010` by 18% relative. However, apart from when there is very limited adaptation data (10s–60s) this combination does not improve over LHUC+fMLLR. fMLLR transforms within this work were estimated once on all data available for the given speaker.

Fig 3 (right) presents a supervised adaptation (oracle) experiment in which the adaptation targets were obtained by aligning the audio data with reference transcripts. This experiment was performed in order to demonstrate the modelling capacity of the different model-based adaptation techniques. We do not refine what the model knows about speech, nor the way it classifies it (the feature receptors and output layer are fixed during adaptation and remain speaker independent), but show that the recomposition and interpolation of these basis functions by learning their roles in approximating (accurate) adaptation examples is able to decrease the WER by 26% relative for DiffP+LHUC scenario and 26.3% relative when combined with fMLLR. This supports our previous hypothesis [20] that effective adaptation methods can be designed in the space of speaker-independent components in which the final SD model is derived by appropriate selecting a relatively small number of weighting coefficients (compared to the number of model parameters). Fig 3 also shows that DiffP and LHUC are highly complementary to each other in terms of modelling capacity, and once accurate adaptation targets are available, test-only adaptation can learn the contribution of speaker adaptive training.

Results for the three predefined IWSLT12 test sets and various combinations of adaptation techniques are summarised in Table 1.

6. CONCLUSIONS

We have proposed a novel adaptation technique that performs speaker adaptation by learning speaker-specific pooling regions – DiffP. Our results across three IWSLT test sets indicate that the approach consistently reduces word error rates by 5–11% relative compared to unadapted systems, with a further reduction of 6–15% relative when combined with previously proposed LHUC adaptation. DiffP adaptation was found to work well with very limited adaptation data and does not require speaker adaptive training. Future work will compare and combine the approach with adaptation using i-vector auxiliary features. Code and recipes enabling reproduction of this research:
`http://www.cstr.ed.ac.uk/reproducibleResearch/`.

7. REFERENCES

[1] H Bourlard and N Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[2] S Renals, N Morgan, H Bourlard, M Cohen, and H Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Trans Speech and Audio Processing*, vol. 2, pp. 161–174, 1994.

[3] G Hinton, L Deng, D Yu, GE Dahl, A Mohamed, N Jaitly, A Senior, V Vanhoucke, P Nguyen, TN Sainath, and B Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[4] J Neto, L Almeida, M Hochberg, C Martins, L Nunes, S Renals, and T Robinson, "Speaker adaptation for hybrid HMM–ANN continuous speech recognition system," in *Proc Eurospeech*, 1995, pp. 2171–2174.

[5] V Abrash, H Franco, A Sankar, and M Cohen, "Connectionist speaker normalization and adaptation," in *Proc Eurospeech*, 1995, pp. 2183–2186.

[6] F Seide, X Chen, and D Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc IEEE ASRU*, 2011.

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

[8] D Yu, K Yao, H Su, G Li, and F Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition.," in *Proc IEEE ICASSP*, 2013, pp. 7893–7897.

[9] H Liao, "Speaker adaptation of context dependent deep neural networks.," in *In Proc. ICASSP*. 2013, pp. 7947–7951, IEEE.

[10] MJF Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, April 1998.

[11] P Bell, P Swietojanski, and S Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc IEEE ICASSP*, 2013.

[12] N Dehak, PJ Kenny, R Dehak, P Dumouchel, and P Ouellet, "Front end factor analysis for speaker verification," *IEEE Trans Audio, Speech and Language Processing*, vol. 19, pp. 788–798, 2010.

[13] G Saon, H Soltau, D Nahamoo, and M Picheny, "Speaker adaptation of neural network acoustic models using i-vectors.," in *Proc IEEE ASRU*, 2013, pp. 55–59.

[14] K Yao, D Yu, F Seide, H Su, L Deng, and Y Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition.," in *Proc IEEE SLT*, 2012.

[15] SM Siniscalchi, J Li, and CH Lee, "Hermitian polynomial for speaker adaptation of connectionist speech recognition systems," *IEEE Trans Audio, Speech, and Language Processing*, vol. 21, pp. 2152–2161, 2013.

[16] JS Bridle and S Cox, "Recnorm: Simultaneous normalisation and classification applied to speech recognition," in *Advances in Neural Information Processing Systems 3*, 1990, pp. 234–240.

[17] O Abdel-Hamid and H Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc IEEE ICASSP*, 2013, pp. 4277–4280.

[18] J Xue, J Li, D Yu, M Seltzer, and Y Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proc IEEE ICASSP*, 2014.

[19] O Abdel-Hamid and H Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition.," in *Proc. Interspeech*. pp. 1248–1252, ISCA.

[20] P Swietojanski and S Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE SLT*, 2014.

[21] K Fukushima and S Miyake, "Neocognitron: A new algoriothm for pattern recognition tolerant of deformations," *Pattern Recognition*, vol. 15, pp. 455–469, 1982.

[22] Y LeCun, B Boser, JS Denker, D Henderson, RE Howard, W Hubbard, and LD Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.

[23] Y LeCun, L Bottou, Y Bengio, and P Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[24] M Riesenhuber and T Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[25] MA Ranzato, FJ Huang, Y-L Boureau, and Y LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *IEEE CVPR*, 2007.

[26] Y-L Boureau, J Ponce, and Y LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc ICML*, 2010.

[27] IJ Goodfellow, D Warde-Farley, M Mirza, A Courville, and Y Bengio, "Maxout networks," *arXiv:1302.4389*, 2013.

[28] Y. Miao, F. Metze, and S. Rawat, "Deep maxout networks for low-resource speech recognition," in *Proc. IEEE ASRU*, 2013.

[29] M. Cai, Y. Shi, and J. Liu, "Deep maxout neural networks for speech recognition," in *Proc. ASRU*, Dec 2013, pp. 291–296.

[30] P Swietojanski, J Li, and J-T Huang, "Investigation of maxout networks for speech recognition," in *Proc IEEE ICASSP*, 2014.

[31] S Renals and P Swietojanski, "Neural networks for distant speech recognition," in *Proc HSCMA*, 2014.

[32] L Toth, "Convolutional deep maxout networks for phone recognition," in *Proc Interspeech*, 2014.

[33] M D Zeiler and R Fergus, "Differentiable pooling for hierarchical feature learning," *CoRR*, vol. abs/1207.0151, 2012.

[34] P Sermanet, S Chintala, and Y LeCun, "Convolutional neural networks applied to house numbers digit classification," *CoRR*, vol. abs/1204.3968, 2012.

[35] T N Sainath, B Kingsbury, A Mohamed, G E Dahl, G Saon, H Soltau, T Beran, A Y Aravkin, and B Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *In Proc. IEEE ASRU*, 2013, pp. 315–320.

[36] X Zhang, J Trmal, D Povey, and S Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, 2014.

[37] Ç Gülçehre, K Cho, R Pascanu, and Y Bengio, "Learned-norm pooling for deep neural networks," *CoRR*, vol. abs/1311.1780, 2013.

[38] E Trentin, "Networks with trainable amplitude of activation functions," *Neural Networs*, vol. 14, pp. 471–493, 2001.

[39] M Federico, M Cettolo, L Bentivogli, M Paul, and S Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc IWSLT*, 2012.

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

[41] IJ Goodfellow, D Warde-Farley, P Lamblin, V Dumoulin, M Mirza, R Pascanu, J Bergstra, F Bastien, and Y Bengio, "Pylearn2: a machine learning research library," *arXiv preprint arXiv:1308.4214*, 2013.