Structured Output Layer with Auxiliary Targets for Context-Dependent Acoustic Modelling

Pawel Swietojanski, Peter Bell, Steve Renals

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom

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

Abstract

In previous work we have introduced a multi-task training technique for neural network acoustic modelling, in which context-dependent and context-independent targets are jointly learned. In this paper, we extend the approach by structuring the output layer such that the context-dependent outputs are dependent on the context-independent outputs, thus using the context-independent predictions at run-time. We have also investigated the applicability of this idea to unsupervised speaker adaptation as an approach to overcome the data sparsity issues that comes to the fore when estimating systems with a large number of context-dependent states, when data is limited. We have experimented with various amounts of training material (from 10 to 300 hours) and find the proposed techniques are particularly well suited to data-constrained conditions allowing to better utilise large context-dependent state-clustered trees. Experimental results are reported for large vocabulary speech recognition using the Switchboard and TED corpora.

Index Terms: multitask learning, structured output layer, adaptation, deep neural networks

1. Introduction

Modelling context-dependent (CD) phones using tied-state clustered trees, initially proposed by Young and Woodland [1], has been a cornerstone of acoustic modelling for more than two decades, providing a flexible data-driven framework for managing the trade-off between the amount of training material and the final size of the model. Combining this technique with deep neural network (DNN) / hidden Markov model (HMM) hybrid models [2, 3] was one of the major factors in the recent success of DNNs for acoustic modelling [4, 5, 6]. The use of CD states as targets for a single DNN allows for a greater level of parameter sharing, in comparison with Gaussian mixture models (GMMs), where distinct clusters are modelled by a different mixtures of Gaussian components, as well as with earlier approaches to modelling context-dependency in hybrid models, where an ensemble of distinct networks was trained to estimate a set of conditional probabilities necessary to derive a CD likelihood score [7, 8, 9].

Despite its widespread and successful use, the optimal clustering for GMM-based systems is often suboptimal for DNNs [10, 11]. Under data-constrained conditions some additional initialisation techniques [5, 12, 13, 14, 15] need to be applied to fully utilise large CD trees. We propose a structured output layer – an approach that allows the optimisation and prediction of CD and context-independent (CI) targets jointly, with an explicit dependence of CD targets on CI targets. This makes it possible to use CI predictions at test time as well as learning a more difficult task in combination with an easier one.

2. Structured Output Layer

We build our model based on a multi-task learning approach [16] and its applications to robust [17] and cross-lingual [18, 19, 20] acoustic modelling, where the hidden representation is shared and jointly optimised across tasks. In this paper we are concerned with multi-task training within a single language. The choice of an auxiliary task was inspired by the work of Zhang and Woodland [13] who found the use of CI targets for layer-wise discriminative pre-training followed by CD fine-tuning leads to the models that better generalize, and, due to low dimensionality of the CI task, are also faster to pre-train. The idea of layer-wise pre-training itself was proposed by Bengio et al. [21] and was further explored in acoustic modelling for speech recognition by Seide et al. [12]. However, in [12], contrary to [13], pre-training and fine-tuning relied on the same context-dependent task. More recently we extended the CI-based initialisation technique to multi-task training [22] where both context-independent and context-dependent targets are jointly trained. All these methods implicitly implement a form of *curriculum learning* [23] where a lower entropy task (with respect to the complexity of classification task or the number of the optimised weights used for intermediate predictions) is employed to iteratively place some relevant prior on the parameters: for example, by forcing the model to predict simpler (but related) concepts first, or using initially fewer parameters which are then expanded as the training progresses.

In this work we further extend [13, 22] by using the CI layer not only at the (pre-)training stage but also to compute CD outputs at run-time – the structured output layer (SOL). The SOL estimates the CI outputs m_t as an auxiliary task – (1) and (2). In the original multitask formulation, the CD outputs s_t would be estimated independent of the CI outputs at runtime – (3) and (4) – whereas using the SOL, the CD outputs are given by (5) and (6). If $\mathbf{a}_m$ represents the CI layer activations, and $\mathbf{a}_s$ and $\mathbf{a}_{sm}$ represent the CD layer activations with and without dependency on the the CI layer, then we have:

$$\mathbf{a}_m = (\mathbf{x}_t\mathbf{M} + \mathbf{m}) \tag{1}$$

$$P(m_t|\mathbf{o}_t) = \text{softmax}(\mathbf{a}_m) \tag{2}$$

$$\mathbf{a}_s = (\mathbf{x}_t\mathbf{S} + \mathbf{b}) \tag{3}$$

$$P(s_t|\mathbf{o}_t) = \text{softmax}(\mathbf{a}_s) \tag{4}$$

$$\begin{aligned}\mathbf{a}_{sm} &= (\mathbf{x}_t\mathbf{S} + \psi(\mathbf{a}_m)\mathbf{C} + \mathbf{b}) \\ &= (\mathbf{a}_s + \psi(\mathbf{a}_m)\mathbf{C})\end{aligned} \tag{5}$$

$$P(s_t|\mathbf{o}_t) = \text{softmax}(\mathbf{a}_{sm}), \tag{6}$$

where $\mathbf{o}_t$ is the acoustic input. The SOL layer, depicted in Fig 1, is then composed of parameters $\boldsymbol{\theta}_{SOL} = \{\mathbf{S}, \mathbf{M}, \mathbf{C}, \mathbf{b}, \mathbf{m}\}$, where $\mathbf{S} \in \mathbb{R}^{X \times S}$ and $\mathbf{b} \in \mathbb{R}^S$ represent hidden to CD weight matrix and bias, respectively. $\mathbf{M} \in \mathbb{R}^{X \times M}$ and $\mathbf{m} \in \mathbb{R}^M$ are

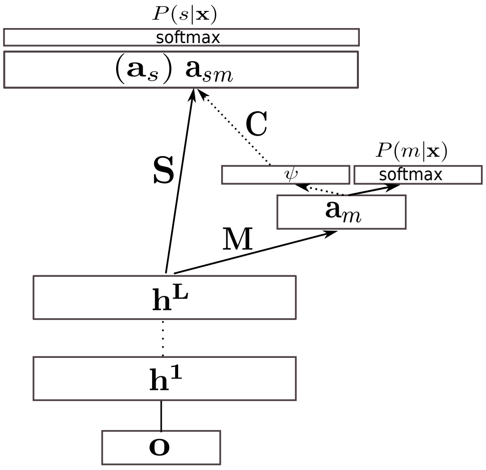

Figure 1: The model with the structured output layer (SOL). $P(s|\mathbf{x})$ can be computed with or without a dependendency on the mononphone layer to compute either $\mathbf{a}_{sm}$ (5) or $\mathbf{a}_s$ (3).

for hidden to CI targets while $\mathbf{C} \in \mathbb{R}^{M \times S}$ are the CI to CD connection weights, allowing us to use the easier monophone prediction task when deciding on (harder) context-dependent tied-state both in training and also at run-time. ψ is the non-linearity used for the activations of the CI layer in the SOL. The remaining part of the model follows the usual structure with L hidden layers $\{\mathbf{h}^1, \dots, \mathbf{h}^L\}$. In the remainder of this paper, we will be focused mostly on the SOL layer itself, rather than the model as a whole. As such, we introduce an auxiliary variable $\mathbf{x}_t \in \mathbb{R}^X$ which denotes the vector of top hidden layer activations at a time t, or when considering a mini-batch of examples, $\mathbf{x}_t$ becomes to be $\mathbf{x} \in \mathbb{R}^{B \times X}$, where B is the mini-batch size.

We learn the model by optimising a global cost, the weighted sum of each of the separate costs:

$$\mathcal{F} = (1-\alpha)\mathcal{F}_s + \alpha\mathcal{F}_m, \tag{7}$$

where both CD ($\mathcal{F}_s = -\sum_t \log P\left(s_t|\mathbf{o}_t;\boldsymbol{\theta}_s\right)$) and CI ($\mathcal{F}_m = -\sum_t \log P\left(m_t|\mathbf{o}_t;\boldsymbol{\theta}_m\right)$) components are expressed as a gradient descent on a negative log likelihood over T training examples. Note that we obtain both predictions in parallel in a single forward-pass which is different from our earlier work [22] and from the multi-task framework in general, where the tasks are usually treated as independent and processed sequentially. Effectively, the gradients used to update the parameters are expressed as the weighted average of both tasks with the kth parameter's gradient taking the following form:

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}^k} = -\sum_t^T \Big[(1-\alpha)\frac{\partial}{\partial \boldsymbol{\theta}_s^k}\log P\left(s_t|\mathbf{o}_t;\boldsymbol{\theta}_s\right) + \alpha\frac{\partial}{\partial \boldsymbol{\theta}_m^k}\log P\left(m_t|\mathbf{o}_t;\boldsymbol{\theta}_m\right)\Big] \tag{8}$$

Given that $\boldsymbol{\theta}$ includes all the model's parameters (including those in the hidden layers), the task-specific parameter subsets are defined as $\boldsymbol{\theta}_s = \boldsymbol{\theta}\backslash\{\mathbf{M}, \mathbf{m}\}$ and $\boldsymbol{\theta}_m = \boldsymbol{\theta}\backslash\{\mathbf{S}, \mathbf{C}, \mathbf{b}\}$ for $\mathcal{F}_s$ and $\mathcal{F}_m$, respectively. In practice, to perform updates, we simply set unrelated gradients (with respect the given cost) to zero when computing final partial derivatives in (8), for example, we set $\partial\mathcal{F}_m/\partial\mathbf{S} = 0$ and scale the corresponding learning rate for $\partial\mathcal{F}_s/\partial\mathbf{S}$ by $1/(1-\alpha)$. Likewise, for $\mathcal{F}_m$ we set $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m = 0$ and scale the CI learning rate by $1/\alpha$.

Depending on our assumptions, the back-propagation of CD errors may also influence the parameters on CI path, including the $\mathcal{F}_m$ classification layer. We consider four scenarios:

1. Gradients of $\mathcal{F}_s$ on the "monophone" path are truncated after $\mathbf{C}$ and the back-propagated errors from $\mathcal{F}_s$ cost to the lower layers is:

   $$\frac{\partial \mathcal{F}_s}{\partial \mathbf{x}_t} = \frac{\partial \log P\left(s_t|\mathbf{o}_t;\boldsymbol{\theta}_s\right)}{\partial \mathbf{a}_s}\frac{\partial \mathbf{a}_s}{\partial \mathbf{x}_t}$$

2. Gradients flow through $\mathbf{M}$ down to the lower layers, but $\mathbf{M}$ and $\mathbf{m}$ by $\mathcal{F}_s$ is considered constant, so $\{\partial\mathcal{F}_s/\partial\mathbf{M}, \partial\mathcal{F}_s/\partial\mathbf{m}\} = 0$ and the error signals are:

   $$\frac{\partial \mathcal{F}_s}{\partial \mathbf{x}_t} = \frac{\partial \log P\left(s_t|\mathbf{o}_t;\boldsymbol{\theta}_s\right)}{\partial \mathbf{a}_s}\frac{\partial \mathbf{a}_s}{\partial \mathbf{x}_t} + \frac{\partial \log P\left(s_t|\mathbf{o}_t;\boldsymbol{\theta}_s\right)}{\partial \mathbf{a}_{sm}}\frac{\partial \mathbf{a}_{sm}}{\partial \psi}\frac{\partial \psi}{\partial \mathbf{a}_m}\frac{\partial \mathbf{a}_m}{\partial \mathbf{x}_t}$$

3. $\mathcal{F}_S$ influences all dependent parameters, so the back-propagation is as in point 2 above, but partial derivatives $\partial\mathcal{F}_s/\partial\mathbf{M}$ and $\partial\mathcal{F}_s/\partial\mathbf{m}$ are non-zero and used to update $\mathbf{M}$ and $\mathbf{m}$ in eq. (8).

4. $\mathbf{C}$ in not learned jointly in MT learning but is added at post-processing stage and fine-tuned given the predictions for $P\left(s_t|\mathbf{o}_t;\boldsymbol{\theta}_s\right)$ and $P\left(m_t|\mathbf{o}_t;\boldsymbol{\theta}_m\right)$

The model with a SOL layer exhibits the advantages of classic single-language multi-task approaches [22, 24, 25] – its hidden representation is shared across the tasks, so the resulting features are less prone to over-fitting and, as a result, should yield a better generalisation.

The other potential advantage comes from a modelling perspective: it is well known that the perceptron (or a logistic classification layer) can only solve linearly separable problems, with Exclusive Or (XOR) being an infamous example [26]. It is also clear that the transformed acoustic features in the top hidden layer retain highly non-linear characteristics (this can be seen by an error sensisitivty analysis). The well known solution for the "perceptron problem" is an extra intermediate layer connecting the inputs with the outputs [27], or in a even simpler scenario, an extra unit describing the relation between the inputs and sending the outcome to the output unit. The latter case is what an auxiliary layer can do in our model, projecting the activations onto CI space, based on which the CD layer can additionally partition the CD space using CI predictions.

The idea of auxiliary targets has been investigated as a "local" coordinate optimisation system [28], where a long chain of back-propagation through many layers is replaced by a shallow sequence of layer-oriented objectives.

3. Experiments

We work with the TED talks corpus [29] following the IWSLT (`www.iwslt.org`) evaluations and the Switchboard corpus of conversational telephone speech [30].

For TED talks we primarily work on 143 hours of data obtained with light supervision [31], following the recipe described in detail in [32]. For the purpose of this work, we additionally sub-sample random subsets of 10 and 30 hours of training material to simulate data constrained conditions. We do most experiments on the 30 hours split, reporting the most promising configurations on 10 hours and the full 143 hours. This work, compared to [32] benefits from better language models developed for 2014 IWSLT evaluation campaign [33], in

Table 1: WER(%) results on tst2010 set. Models trained on 30 hour data-split with $\alpha = 0.3$

	Model	tst2010	+4gm
S1	CD−NN (1k)	23.1 ± 0.1	19.7
S2	SOL−NN const.$\partial\mathcal{F}_s/\partial\{\mathbf{M},\mathbf{m}\}$	22.8	19.5
S3	SOL−NN $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m$	21.9 ± 0.2	18.7
S4	SOL−NN + PI Monophones	22.7	19.4
S5	SOL−NN + Retrained CD	22.5	19.2
S6	CD−NN (2k)	22.6	19.2
S7	SOL−NN (2k) $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m$	21.6	18.5

particular, we decode with pruned trigrams and rescore with 4-grams. The models are trained on unadapted PLP [34] features with first and second order time derivatives with an 11 (±5) frame context window. All models under all data conditions are trained with 12,000 CD targets. For the CI task we use 186 position-dependent phones, and in some control experiments we use 45 monophones. For data constrained scenarios (10 and 30 hours) our models have 6 hidden layers with 1,000 units each, we additionally perform low-rank factorisation of the output CD layer by inserting a linear-bottleneck [35, 36], i.e. our layer becomes $\mathbf{S} = \mathbf{S}_{in} \times \mathbf{S}_{out}$, where $\mathbf{S}_{in} \in \mathbb{R}^{X \times N}$ and $\mathbf{S}_{out} \in \mathbb{R}^{N \times S}$ with N=256.

For Switchboard we use the Kaldi GMM recipe [37, 38], using Switchboard-1 Release 2 (LDC97S62). Our acoustic models share the same architecture as the models used for Full-TED data, except for compatibility with the results reported by other researchers we train on unadapted MFCC features. The results are reported on Hub5 00 (LDC2002S09).

3.1. Structured output layer

In this section we look at different training scenarios for SOL−NN, comparing with a baseline DNN model, with 1,000 hidden units and the low-rank factorisation of the CD output layer. The baseline results are given in row (S1) of Table 1.

We explored the training scenarios outlined in Section 2. We found that both truncation of $\mathbf{C}$ (scenario 1) and optimising $\mathbf{C}$ as a post-processing step (scenario 4) resulted in very high frame error rates in comparison with the baseline. Row (S2) gives word error rates (WERs) for the case where the CD cost is not used to update $\mathbf{M}$ and $\mathbf{m}$; row (S3) shows the opposite scenario indicating that updating the CI-dependent parameters using the $\mathcal{F}_s$ cost yields the lowest WER, 21.9%, a 6% relative improvement over the baseline. Row (S4) is a model trained on 45 position-independent phones (compared to the other models utilising 186 position-dependent phones). Model (S5) is built from the hidden representation of (S3) with a new CD regression layer showing that both the SOL layer and multitask training are important.[1] Finally, rows (S6) and (S7) present WERs for larger models showing over 1% absolute (or 0.7% for rescored lattices) gain for SOL−NN structure.

Table 2 presents WERs for different activation functions (ψ) connecting $\mathbf{M}$ and $\mathbf{C}$ using (S3) model from Table 1. The linear connection was found to work best, and in the following part of the paper we follow the structure and optimisation procedure used to train model (S3).

Table 2: WER(%) on tst2010 set for different $\mathbf{M}$ to $\mathbf{C}$ ψ activations. The base model is SOL−NN $\partial\mathcal{F}_s/\partial\boldsymbol{\theta}_m$, $\alpha = 0.3$

ψ-activation				
linear	softmax	sigmoid	relu	tanh
21.9	22.2	22.5	22.4	23.1

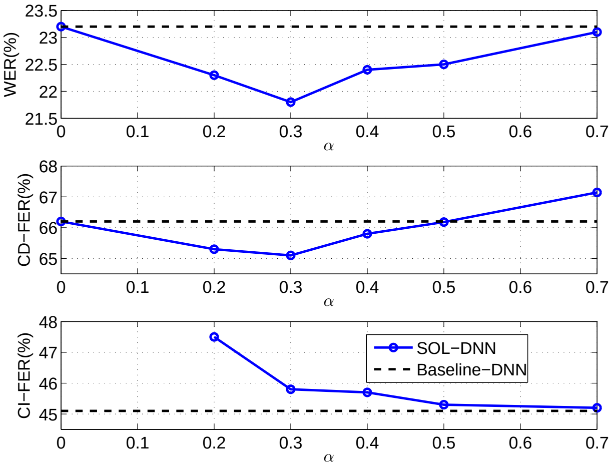

Figure 2: WER and FER as a function of α.

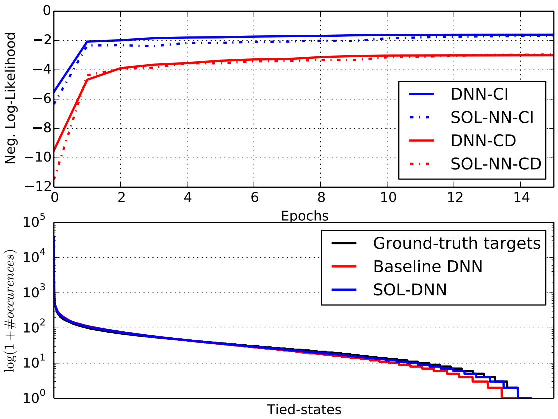

Figure 3: Top) Convergence plots for SOL−NN and baseline CD/CI models models on dev2010. Bottom) Distribution of CD states obtained from ground-truth labels and the estimates of CD−NN and SOL−NN models.

Figure 2 shows WER (on tst2010), as well as corresponding CD and CI frame error rates (FER) for different weighting constants α, the best WER results (and also FER for CD task) were obtained with α=0.3.

Finally, Fig 3 shows convergence plots of baseline and SOL−NN models (no significant differences) as well as the predicted distributions of CD states under both models compared with the expected one obtained using ground-truth alignments of dev2010 (all sorted by occurrence frequencies). The SOL−NN better deals with modelling a tail of a distribution, which could explain why there are small differences in the log likelihoods but significant reductions in word error rates.

[1] We did some sanity checks, and the corresponding models (S1) and (S3) with retrained top layers converged to their base model accuracies where all layers were jointly optimised.

Table 3: Detailed results on tst2010 and unsupervised adaptation with LHUC using auxiliary targets on 30 hours models.

	α				
System	0	0.3	0.5	0.7	1
Baseline	21.9				
+4gm	18.6				
Adaptation with 10 seconds per speaker					
+LHUC	21.3	21.2	21.1	21.25	21.6
++4gm	18.0	18.0	17.9	18.2	18.45
Adaptation with all speaker's data					
+LHUC	18.3	18.3	18.3	18.5	19.0
++4gm	16.0	15.7	15.8	15.7	16.1

3.2. Multi-task adaptation

In this section we investigate the feasibility of using the CI targets to perform unsupervised two-pass adaptation. Our motivation is that CI modelling is usually characterised by a lower frame error rate, and at the same time there is less sparsity in the distribution of CI targets, given the same amount of adaptation data, hence potentially obtaining better adaptation results compared to a CD-only objective. A similar approach, but using hierarchy of output layers and CI-only adaptation targets was proposed by Price et al. in [39].

We adapt our speaker-independent models with a technique which learns hidden unit contributions (LHUC) [40, 41] given unsupervised adaptation data. We report the adaptation results for two scenarios, using both a limited amount of 10 seconds of speech per speaker as well as full two pass adaptation. For the 10s scenario we repeated the experiments 5 times, for randomly selected utterances, and report the average WERs.

The results on tst2010 are reported in Table 3 showing that around 0.2-0.3% absolute gain was obtained on top of CD-only adaptation for both scenarios. Interestingly, when using all adaptation data available for a given speaker, interpolated adaptation does not bring a WER reduction with a pruned trigram LM, but re-scoring with a 4-gram brings up to 0.3% absolute gain on top of CD targets only. The cost objectives for adaptation with different α are plotted in Fig 4.

We observe similar gains when adapting models on other test sets. With $\alpha = 0.5$, WER on dev2010 is 0.3% abs. lower for the 10s adaptation scenario. Similarly to tst2010, interpolated ($\alpha = 0.5$) adaptation on dev2010 with the whole speaker's data reduced the WER by 0.2% abs. when compared to CD-only adaptation. On tst2011 adapting with 10s gave smaller reductions for both methods (0.2% abs.) regardless of α; this could be due to the fact tst2011 is better matched to training conditions and benefits less from adaptation.

3.3. Full TED and Switchboard

Finally, we report summary results on three scenarios for TED (10, 30 and 143 hours) in Table 4, as expected, we observe similar gains from the proposed method on even more constrained scenario (10 hours) and also, as expected, the adaptation brings larger gains there since the SI models see less speaker variability during training. For larger models and more training data the gains diminish, with a small improvement of 0.3% on tst2011, but not on dev2010 and tst2010 for Full TED. On Switchboard task (Table 5) the SOL-NN model reduced WERs on Switchboard (SWB) part at the same time increasing the metric on CallHome (CHE) test data, making the model falling back 0.2% WER in average behind the baseline.

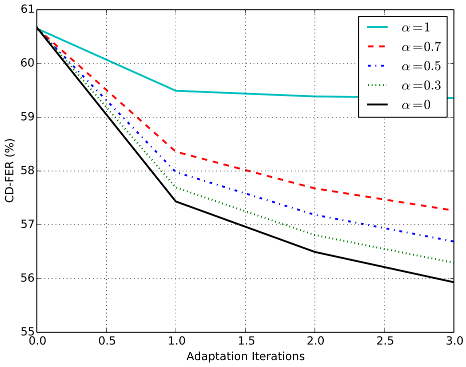

Figure 4: Context-dependent frame error rates as a function of adapting iterations for different values of α on tst2010.

Table 4: Summary results on the remaining TED test-sets and different amounts of training material and LHUC adaptation with pruned and (4gm) language models.

	WER (%)		
System	dev2010	tst2010	tst2011
10 hour			
CD-NN	27.0 (23.3)	28.1 (24.4)	22.7 (19.3)
SOL-NN	25.6 (21.9)	26.6 (22.8)	21.3 (18.3)
+LHUC (All)	22.8 (19.8)	22.4 (19.4)	19.0 (16.7)
30 hour			
CD-NN	22.9 (19.8)	23.2 (19.7)	19.2 (15.8)
SOL-NN	22.0 (19.1)	21.9 (18.7)	17.8 (14.9)
+LHUC (All)	19.3 (17.1)	18.3 (16.0)	15.5 (13.4)
143 hour			
CD-NN	18.3 (15.7)	17.9 (15.2)	14.6 (12.5)
SOL-NN	18.3 (15.7)	17.9 (15.1)	14.4 (12.2)
+LHUC (All)	16.8 (14.6)	14.9 (12.7)	12.8 (11.2)

Table 5: WER(%) on Switchboard Hub00

	Hub5'00		
Model	SWB	CHE	TOTAL
CD-NN	15.8	28.4	22.1
SOL-NN	15.6	28.9	22.3

4. Conclusions

We have proposed a structured output layer, an approach in which an auxiliary (context-independent) task is used as a regularizer during training but also as an auxiliary predictor in deriving context-dependent tied states for decoding. We have investigated various training strategies for this technique, and have shown that this approach is an effective way of addressing unsupervised adaptation with sparse data. For future work we are interested in evaluating with respect to other techniques proposed for low-resource speech recognition as well as extending to sequence discriminative training.

Acknowledgement: This research was supported by EPSRC Programme Grant EP/I031022/1, Natural Speech Technology (NST). The NST research data collection may be accessed at http://datashare.is.ed.ac.uk/handle/10283/786. Experiments were carried using Kaldi Speech Recognition Toolkit [38] while the acoustic modelling part was developed in Pylearn2 [42].

5. References

[1] S. J. Young and P. C. Woodland, "State clustering in hidden Markov model-based continuous speech recognition," *Computer Speech & Language*, vol. 8, no. 4, pp. 369–383, 1994.

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

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

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

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

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

[7] N. Morgan and H. Bourlard, "Factoring networks by a statistical method," *Neural Computation*, vol. 4, no. 6, pp. 835–838, 1992.

[8] H. Bourlard, N. Morgan, C. Wooters, and S. Renals, "CDNN: a context dependent neural network for continuous speech recognition," in *Proc. ICASSP*, vol. 2, 1992, pp. 349–352.

[9] H. Franco, M. Cohen, N. Morgan, D. Rumelhart, and V. Abrash, "Context-dependent connectionist probability estimation in a hybrid HMM-neural net speech recognition system," *Computer Speech and Language*, vol. 8, pp. 211–222, 1994.

[10] G. Wang and K. C. Sim, "Regression-based context-dependent modeling of deep neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 11, pp. 1660–1669, Nov 2014.

[11] M. Bacchiani and D. Rybach, "Context dependent state tying for speech recognition using deep neural network acoustic models," in *Proc. ICASSP*, 2014, pp. 230–234.

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

[13] C. Zhang and P. Woodland, "Context independent discriminative pre-training," unpublished work.

[14] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proc. IEEE SLT*, December 2012, pp. 246–251.

[15] Y. Miao and F. Metze, "Improving low-resource cd-dnn-hmm using dropout and multilingual dnn training." in *Proc. Interspeech*. ISCA, 2013, pp. 2237–2241.

[16] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997.

[17] S. Parveen and P. Green, "Multitask learning in connectionist robust ASR using recurrent neural networks," in *Proc. Interspeech*, 2003.

[18] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*, 2013.

[19] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *In Proc. ICASSP*, 2013.

[20] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *In Proc. ICASSP*, 2013.

[21] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19*, 2007, pp. 153–160.

[22] P. Bell and S. Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task training," in *Proc. ICASSP*, 2015.

[23] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. ICML*, 2009.

[24] D. Chen, B. Mak, C.-C. Leung, and S. Sivadas, "Joint acoustic modelling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *Proc. ICASSP*, 2014.

[25] M. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proc. ICASSP*, 2013.

[26] M. Minsky and S. Papert, *Perceptrons*. MIT Press, 1969.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error-propagation," in *Parallel Distributed Processing*. MIT Press, 1986, vol. 1, pp. 318–362.

[28] M. Á. Carreira-Perpiñán and W. Wang, "Distributed optimization of deeply nested systems," *CoRR*, vol. abs/1212.5921, 2012. [Online]. Available: http://arxiv.org/abs/1212.5921

[29] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proc EAMT*, 2012, pp. 261–268.

[30] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*. IEEE, 1992, pp. 517–520.

[31] E. Hasler, P. Bell, A. Ghoshal, B. Haddow, P. Koehn, F. McInnes, S. Renals, and P. Swietojanski, "The UEDIN systems for the IWSLT 2012 evaluation," in *Proc. IWSLT*, 2012.

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

[33] P. Bell, P. Swietojanski, J. Driesen, M. Sinclair, F. McInnes, and S. Renals, "The UEDIN system for the IWSLT 2014 evaluation," in *Proc. IWSLT*, 2014.

[34] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.

[35] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets." in *Proc. ICASSP*, 2013, pp. 6655–6659.

[36] F. Grézl, M. Karafiát, S. Kontar, and J. Černokcý, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc. ICASSP*, 2007.

[37] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, Lyon, France, August 2013.

[38] 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.

[39] R. Price, K. Iso, and K. Shinoda, "Speaker adaptation of deep neural networks using a hierarchy of output layers," in *Proc. IEEE SLT*, 2014.

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

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

[42] I. 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:1308.4214, 2013.