INTERSPEECH 2009 BRIGHTON

Efficient Generation and Use of MLP Features for Arabic Speech Recognition

J. Park, F. Diehl, M.J.F. Gales, M. Tomalin, & P.C. Woodland

Department of Engineering, University of Cambridge,
Trumpington St., Cambridge, CB2 1PZ, U.K.
{jhp33, fd257, mjfg, mt126, pcw}@eng.cam.ac.uk

Abstract

Front-end features computed using Multi-Layer Perceptrons (MLPs) have recently attracted much interest, but are a challenge to scale to large networks and very large training data sets. This paper discusses methods to reduce the training time for the generation of MLP features and their use in an ASR system using a variety of techniques: parallel training of a set of MLPs on different data sub-sets; methods for computing features from by a combination of these networks; and rapid discriminative training of HMMs using MLP-based features. The impact on MLP frame-based accuracy using different training strategies is discussed along with the effect on word rates from incorporating the MLP features in various configurations into an Arabic broadcast audio transcription system.

Index Terms: Arabic Speech Recognition, Multi-Layer Perceptron, Acoustic Modelling

10.21437/Interspeech.2009-84

1. Introduction

In recent years, the use of front-end features derived from Multi-Layer Perceptrons (MLP) trained to estimate phone posterior probabilities have received a lot of attention [1][2]. Such features perform well in when used in the acoustic feature vector to augment more traditional features such as cepstra based on perceptual linear prediction (PLP) [3]. However, training the MLPs is computationally expensive, especially if large networks and training sets are used. Furthermore, each time the acoustic front-end is changed the associated HMM-based acoustic models need to be re-trained with is also time consuming if discriminative training is used, as is now common-place in state-of-the-art systems. This paper investigates strategies to reduce the time required to first train MLPs on large data sets and then to make discriminative training of HMMs with these features more efficient.

MLP training for acoustic feature extraction tends to be very computationally expensive. It has been shown [2, 3] that MLP performance depends on the amount of available training data. However, for a fixed-size network, training time is roughly proportional to the amount of data. For large MLPs with millions of parameters, and hundreds of millions of training frames (hundreds or thousands of hours), MLP training can require many days or even weeks on modern multi-threaded multi-core computers. This time might be reduced by a more aggressive training schedule [2] but with the risk that this will lead to poorer performance from the final MLP. A drawback of a conventional MLP design is that if additional training data becomes available, a new network needs to be trained to incorporate it. One solution to address this training problem is to divide the training set into N subsets, train a separate MLP on each and then combine the outputs from these MLPs with a separate merger network. This technique allows the N subset MLPs to be trained in parallel reducing the overall latency for MLP construction by a factor of N. Furthermore, any additional data can be dealt with by training further subset MLPs and only the merger network needs to be retrained.

Once the MLP-based features have been derived for the entire training set, new HMM-based acoustic models must be estimated. Assuming that discriminative training is used, as is the case with most state-of-the-art systems, the main computational burden is the lattice generation step [4]. To reduce the computational load, lattices that have been previously generated for an alternative front-end can be used with the assumption that the sets of confusable hypotheses and model-level segmentations remain unaltered. The approach was introduced as a "rapid MLP system build" in [6], but no detailed performance contrasts were given. In this paper, the effect of discriminative training lattice mismatch is investigated.

In this paper, the various strategies for efficient MLP training and rapid acoustic model estimation are explored in the context a state-of-the-art Arabic large vocabulary continuous speech recognition (LVCSR) system developed for the DARPA GALE project.

2. MLP subset combination

This section assumes that MLPs are to be trained on individual subsets of the data. Issues that need to be addressed include the training of these subset MLPs and how these individual network outputs should be combined.

2.1. Parallel training of subset MLPs

In the parallel training stage, the training data has been divided into N sets and $N-1$ of these sets are used to train subset MLPs. The remaining one set of data is used for training the merger network. In this paper, 4-layer MLPs, with the MLP features generated at a 'bottleneck' layer [7], were used. The use of features generated at an intermediate bottleneck layer has several advantages over using posteriors at the MLP output. In particular, such features remove the need for an additional dimensionality reduction step. Furthermore, the merger network can take as input the outputs from the bottleneck layer from each of the subset MLPs. The data is partitioned using a random selection into N chunks. In this work a total of 1350 hours of Arabic acoustic data was available. It was divided into six sets of 200 hours each for MLP training, leaving 150 hours for the training of the merger network. All of subset MLPs are trained by wLP-TRAP

This work was in part supported by DARPA under the GALE programme via a subcontract to BBN Technologies. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

input feature used in [3][6]. Frame accuracies are given for each of the 200 hours subset MLPs in Table 1 and all subset networks give similar results.

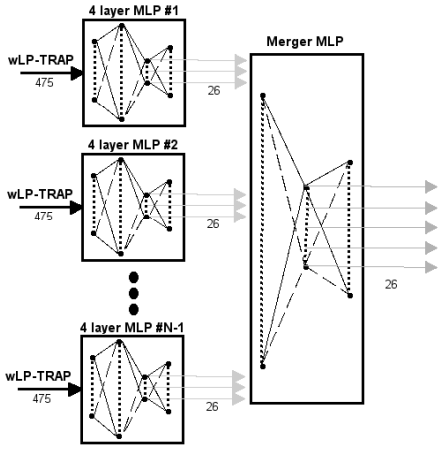

Figure 1: Configuration of the combination MLP.

2.2. Merger network to combine MLP outputs

Once the subset MLPs have been trained, the merger network must be designed. Since bottleneck features will be used as the input, it is convenient to use another MLP to combine these sets of features.

There are several options for designing the merger MLP including the level of complexity and the number of hidden layers. In preliminary experiments, it was found that using a 4-layer bottleneck merger network was unnecessary and didn't outperform a much simpler 3-layer merger MLP, where the hidden layer is again a bottleneck structure. The size of the input layer of merger MLP depends on the number of subset MLPs and the size of their bottleneck layers. The size of the second bottleneck layer will determine the final number of MLP features and the third layer corresponds to the number of phones since the networks are trained to estimate phone posterior probabilities. This simple network can be trained very rapidly in comparison with the other networks in the system. Figure 1 presents the overall network structure for the combination network.

2.3. Selecting bottleneck output as input of merger

In order to train the merger network, an appropriate output of subset MLPs must be selected as input of merger network. As briefly mentioned subsection 2.1, taking bottleneck outputs as input of merger network has an advantage over phone posterior outputs in terms of dimensionality reduction. In this paper, one further option for selecting the bottleneck output is explored in which the bottleneck output from either before or after the sigmoid activation function is used. If the output is taken before the sigmoid, it is passed, after mean and variance normalisation, to the merger network ('Lin Output Merger', LOM). In this case, the inputs to the merger network are unbounded, and it was found to lead to some features with relatively small variances within the HMM acoustic models. The alternative to to take bottleneck outputs after the sigmoid activation functions from the subset MLPs ('Sig Output Merger', SOM). In this case no mean and variance normalisation is performed as these features are already bounded between zero and one. These sigmoid output features make training of merger network simple and perhaps handle outliers better than the LOM approach.

MLP config.	Training data	Train Acc.	Test Acc.
Bottleneck	1350hr	69.51%	65.61%
	1350hr_big	71.37%	67.57%
	200hr set #1	68.77%	63.92%
	200hr set #2	68.54%	63.66%
	200hr set #3	68.69%	63.86%
	200hr set #4	68.74%	63.91%
	200hr set #5	68.53%	63.71%
	200hr set #6	68.33%	63.32%
LOM	150hr	68.86%	65.42%
SOM	150hr	69.57%	66.09%

Table 1: Frame accuracies for the 1350 hours, 200 hours MLPs, and for the merger MLPs of the combination networks.

2.4. Evaluation of MLP frame accuracies

Table 1 contrasts a single MLP network trained on 1350 hours of audio with the 6 subset MLPs each trained on 200 hours of data. All these networks had 475 features in the input layer, 3500 hidden nodes, a bottleneck layer with 26 features. A further single MLP '1350hr_big' was trained with double the size of hidden layer (7000 nodes). The 200hr networks have an approximately 1.8% lower frame accuracy than the 1350hr network. However, combining the 200hr networks reduces the performance gap. Though the LOM configuration still performs 0.1% absolute poorer than the 1350hr network, the SOM configuration outperforms the 1350hr network by 0.5% in absolute frame accuracy. As the SOM performs better than the LOM network the LOM is not further used for this work. Finally, comparing the 1350hr_big network with the SOM network, an additional gain of approximately 1.5% in absolute testset frame accuracy is found.

For both acoustic model training and decoding, the data needs to be fed-forward through the MLPs. For the combination networks, this is computationally more expensive than for the single network. However, the computational load of this forward pass is quite small compared with either training the networks or to decoding. The parallel network build reduced the elapsed training time from 288 hours for a 3500 hidden node MLP, 720 hours for the larger 7000 hidden node network to a total of 60 hours (including data feed-forwarding) for training the subset MLPs and the combination network.

3. Rapid MLP system build

State-of-the-art LVCSR systems typically make use of discriminative training schemes such as Minimum Phone Error (MPE) training [5], and these often use lattices as a compact representation of all possible competing paths. The lattice paths will depend on the form of the acoustic models being used. Changing the acoustic front-end may change the set of paths, and may therefore have a significant impact on discriminative training methods. Incorporating MLP-features at the acoustic front-end should require new lattices. However, this is computationally expensive as thousand of hours of training data might be used in a large system.

To address this, a 'fast' build was proposed in [6]. The MLP features were constrained to share as much of the standard PLP-feature configuration as possible. The acoustic models, which used PLP+MLP-features or pure MLP-features, used the same decision tree and linear feature-transform as the underlying PLP-system. As a starting point *single-pass retraining* (SPR) from the standard PLP-system to the PLP+MLP-system was used to

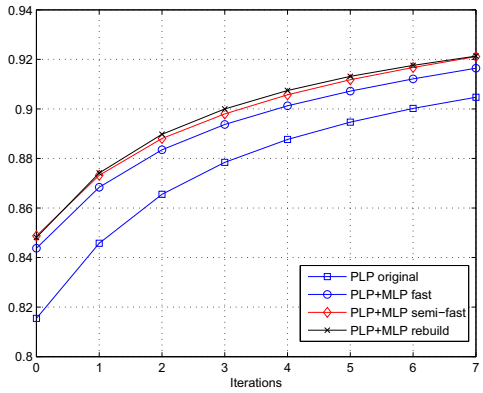

Figure 2: Normalised expected phone accuracies for different MPE training schedules.

make the lattices more appropriate. For the discriminative training stage, the PLP+MLP-system share lattices from the PLP-system. The main training steps are described in [6].

The 'fast' system performance was compared to 'semi-fast' and 'rebuild' approaches. In case of the 'semi-fast' system, the original lattices were re-phonemarked; in the 'rebuild' case they were rebuilt from scratch. The PLP+MLP$_{1350hr}$ front-end and the 'unadapted' decoding configuration were used (see Section 4.1). To permit a fast turn-around cycle, the training data for the acoustic models was reduced to 180 hours.

Figure 2 shows a significant improvement in terms of normalised expected phone accuracy, as computed for the MPE objective function, for the PLP+MLP systems compared to the standard PLP system. However, the difference between the PLP+MLP-systems is small.

System			WER	
Training	Front End	Lattices	dev07	dev08
MPE	PLP	Original	20.5	22.8
	PLP+MLP$_{1350hr}$	Fast	17.7	20.0
		Semi-fast	17.7	19.9
		Rebuild	17.7	19.9

Table 2: Contrast of the 'fast', 'semi-fast', and 'rebuild' training procedures, (WER in %).

Table 2 shows that all three systems achieve nearly identical performance. This indicates that the mismatch in the front-end between the models used for lattice generation and for model re-estimation doesn't harm the final model quality. Therefore, for all further experiments the 'fast' system build was used to build PLP+MLP models.

4. Experiments

4.1. System Description

Graphemic and phonetic systems based on the 36 graphemes and 39 phonemes (including 3 short-vowels) were built as baseline Arabic PLP-based acoustic systems [8]. These used a 39-dimensional PLP-based front-end that was derived using 13-dimensional PLP, including the Cepstra with first, second and third delta parameters followed by an HLDA projection from 52-dimensions down to 39-dimensions. Cepstral mean normalisation was applied. The baseline system used approximately 1500 hours of acoustic training data. Cross-word decision-tree state-clustered triphones were built with about 9000 states and an average of 36 components per state. MPE was used for discriminative training. Both gender-independent (GI) and gender-dependent (GD) models were then constructed for evaluating cross-adaptation performance.

Two decoding configurations are used, 'unadapted' and 'adapted'. 'Unadapted' is a simple 1-pass with a trigram LM and no acoustic model adaptation. 'Adapted' is a 3-pass decoding (P3). The first two passes implement lattice generation and lattice rescoring with a four-gram LM. Pass three performs acoustic lattice rescoring, applying constrained Maximum Likelihood Linear Regression (MLLR) followed by lattice-MLLR and Confusion Network (CN) decoding. Further details on the training and decoding configuration can be found in [9].

Four different network configurations are investigated: the single bottleneck MLP trained on 1350 hours data (MLP$_{1350hr}$), a 200 hours subset MLP (MLP$_{200hr}$)[1] the double-sized hidden layer bottleneck MLP trained on 1350 hours data(MLP$_{1350hr_big}$) and the 1350 hours network combination configurations with the sigmoid output merger (MLP$_{SOM}$).

The system performance was evaluated on three testsets dev07 (2.58 hours), dev08 (3.04 hours) and a set not used for development eval07[2] (2.85 hours). All testsets consist of Broadcast News/Conversation style data. The LM was trained using approximately 1G words. 24 source-specific components (4 STT acoustic data sets, 6 newswire sources, and 14 webdata sets) are built and merged. Out-of-vocabulary (OOV) rates for the 350k wordlist and the testsets are approximately 1.2%.

4.2. Unadapted decoding results

In a first test series the use of MLP features is evaluated within the 'unadapted' decoding setup applying ML and MPE trained acoustic models. Table 3 gives results comparing a PLP based system with four PLP+MLP systems and a pure MLP system.

On the ML stage all PLP+MLP systems exhibit reductions in WER of 11.3%-16.7% relative compared to the PLP system. PLP+MLP$_{1350hr}$ outperforms PLP+MLP$_{200hr}$ but is itself outperformed by PLP+MLP$_{SOM}$ and PLP+MLP$_{1350hr_big}$. This indicates that an improved system performance can be obtained by applying more network parameters and by using more acoustic data for network training.

System		WER		
		dev07	eval07	dev08
ML	PLP	21.1	22.9	25.1
	MLP$_{1350hr}$	20.2	21.2	22.6
	PLP+MLP$_{200hr}$	18.7	20.1	21.6
	PLP+MLP$_{1350hr}$	18.6	19.8	21.2
	PLP+MLP$_{1350hr_big}$	18.0	19.5	20.9
	PLP+MLP$_{SOM}$	18.2	19.2	20.9
MPE	PLP	15.8	17.7	18.8
	MLP$_{1350hr}$	16.7	18.0	18.9
	PLP+MLP$_{200hr}$	14.7	15.7	16.9
	PLP+MLP$_{1350hr}$	14.6	15.7	16.8
	PLP+MLP$_{1350hr_big}$	14.1	15.7	16.4
	PLP+MLP$_{SOM}$	14.1	15.6	16.6

Table 3: Contrast of the PLP, MLP, and the PLP-MLP front-ends for 'unadapted' decoding using graphemic models, (WER in %).

In the case of MPE trained models similar patterns to ML training are found, although the gains with respect to the PLP case

[1] The best performing 200 hours subsets MLPs.

[2] The GALE `eval07` non-sequestered testset version is used.

were reduced to 6.9%-11.3% relative. This reduction is attributed to the fact that MLP features already exploit a discriminative parameter estimation scheme. This reasoning is supported by comparing the pure PLP and MLP systems in the ML and in the MPE case. In the ML case MLP_{1350hr} outperforms PLP, but in the MPE case PLP performs better than MLP_{1350hr}. Thus, MPE training on a PLP system is more efficient than in the MLP case.

In summary, in case of unadapted decoding a clear advantage for the PLP+MLP_{SOM} system is found. Performing as well as the most advanced single network system, its modular design facilitates the network build process significantly while greatly reducing the training costs.

4.3. Adaptation and System Combination

The first two blocks of Table 4 give P3 adapted decoding results. For lattice generation, a graphemic standard PLP system is used. Pass 3 decoding performs lattice adaptation to the various PLP+MLP, MLP, and PLP systems. All passes apply MPE trained models.

As in the unadapted case, the standard PLP systems are outperformed by the PLP+MLP systems. Reductions in relative WER of 4.5%-8.1% in the graphemic case and of 0.7%-2.6% in the phonetic case are found. The larger gains in the graphemic case are attributed to the MLP features which are trained on phonetic targets, therefore enhancing the graphemic systems by implicit graphemic knowledge. However, in contrast to the unadapted case PLP+MLP_{1350hr_big} outperforms PLP+MLP_{SOM}

In a 2-way Confusion Network Combination (CNC) experiment (first CNC block of Table 4) it is investigated whether the PLP+MLP front-end combination can further be improved by additional CNC with the PLP system. These setups are further compared to a straight forward CNC of the pure PLP and MLP systems. Comparing G3a⊕G3f to G3f only shows that the additional CNC with the pure PLP system gives gains of 0.1%-0.3% in absolute WER. Comparing G3a⊕G3e with G3a⊕G3f confirms the necessity to combine the features at the front end, as this gives gains of up to 0.2% in absolute WER.

In case of 2-way combination of the graphemic PLP and PLP+MLP systems with their phonetic counterparts (second CNC block of Table 4), the pure PLP combination (G3a⊕V3a) is always outperformed by the PLP+MLP combinations. The best performance is obtained by the PLP+MLP_{1350hr_big} combination (G3c⊕V3c). Combining the two PLP+MLP_{SOM} (G3d⊕V3d) does not give any improvements over combining the pure PLP systems.

Finally, 4-way CNC is performed by combining the two standard PLP branches, G3a and V3a with two corresponding PLP+MLP branches. The results (third CNC block of Table 4), indicates no significant difference between these three systems. Furthermore, the best 4-way result (G3a⊕V3a⊕G3c⊕V3c) is only marginally better than the best 2-way result (G3c⊕V3c). However, it is remarkable that the use of the PLP+MLP_{200hr} front-end (G3b⊕V3b) results in a nearly as good performance as the use of one of the more sophisticated PLP+MLP front-ends.

5. Conclusions

It has been shown that training sets of MLPs on data subsets and merging the outputs can be effective. The use of MLP features derived in this way can provide an reduction in WER as well as efficiency in training complex MLPs with very large amounts of data. However, when used with complex adaptation, final WER reductions were not obtained. A number of system combination experiments explored how HMMs using MLP features could be combined with PLP-based systems. The best overall system for Arabic speech recognition was a 4-way CN combination of systems with acoustic models with either PLP-based or PLP+MLP-based features and either graphemic or phonetic models.

System		WER		
		dev07	eval07	dev08
G3a	PLP	13.5	14.3	15.4
G3b	PLP+MLP_{200hr}	12.7	13.4	14.7
G3c	PLP+MLP_{1350hr_big}	12.4	13.3	14.4
G3d	PLP+MLP_{SOM}	12.9	13.5	14.7
G3e	MLP_{1350hr}	14.1	14.7	15.7
G3f	PLP+MLP_{1350hr}	12.5	13.6	14.6
V3a	PLP	11.4	12.8	13.9
V3b	PLP+MLP_{200hr}	11.2	12.5	13.6
V3c	PLP+MLP_{1350hr_big}	11.0	12.3	13.6
V3d	PLP+MLP_{SOM}	11.4	12.5	13.8
CNC	G3a⊕G3e	12.6	13.4	14.5
	G3a⊕G3f	12.4	13.3	14.5
	G3a⊕V3a	11.0	12.4	12.9
	G3b⊕V3b	10.9	11.9	12.9
	G3c⊕V3c	10.5	11.7	12.7
	G3d⊕V3d	11.0	12.0	13.1
	G3a⊕V3a⊕G3b⊕V3b	10.6	11.8	12.6
	G3a⊕V3a⊕G3c⊕V3c	10.6	11.6	12.5
	G3a⊕V3a⊕G3d⊕V3d	10.6	11.6	12.6

Table 4: Contrast of the PLP, MLP, and the PLP-MLP front-ends for 'adapted' P3 and CNC decoding, (WER in %).

6. References

[1] H. Hermansky, D.P.W. Ellis, & S. Sharma, "Connectionist feature extraction for conventional HMM systems", Proc. ICASSP'00.

[2] Q. Zhu, A. Stolcke, B.Y. Chen, & N. Morgan, "Using MLP features in SRI's conversational speech recognition system", Proc. Interspeech'05.

[3] P. Foursek, L. Lamel & J. Gauvain, "Transcribing broadcast data using MLP features", Proc. Interspeech'08.

[4] P. Woodland & D. Povey, "Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition," Computer Speech and Language, vol. 16, no. 1, pp. 25–47.

[5] D. Povey & P.C. Woodland. "Minimum Phone Error and I-Smoothing for Improved Discriminative Training", Proc. ICASSP'02.

[6] J. Park, F. Diehl, M.J.F. Gales, M. Tomalin, & P.C. Woodland, "Training and adapting MLP features for Arabic speech recognition", Proc. ICASSP'09.

[7] F. Grézl, M. Karafiát, S. Kontár, & J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings", Proc. ICASSP'07.

[8] M.J.F. Gales, F. Diehl, C.K. Raut, M. Tomalin, P.C. Woodland, & K. Yu, "Development of a phonetic system for large vocabulary Arabic speech recognition", Proc. ASRU'07.

[9] F. Diehl, M.J.F. Gales, M. Tomalin, & P.C. Woodland, "Phonetic pronunciations for Arabic Speech-To-Text systems", Proc. ICASSP'08,