SEQUENCE TRAINING OF MULTI-TASK ACOUSTIC MODELS USING META-STATE LABELS

Olivier Siohan

Google Inc., New York

ABSTRACT

In this paper, we describe a multi-task learning approach for acoustic modeling where the multiple output layers are used to predict context-dependent (CD) states from different state inventories. Unlike the traditional multitask learning approach which defines a primary and secondary output layers but discards the secondary output after training, we propose to use all output layers for recognition. This can be achieved by designing a decoding network operating on tuples of CD states and combining the scores of the different outputs during search. To support training such models using a sequence-based criterion, we propose to replace the multiple output layers with a single layer encoding the CD state tuples as "meta-states". Experimental results are given on a large Voice Search task evaluated on children's speech.

Index Terms— system combination, multitask learning, sequence training, children's speech

1. INTRODUCTION

In the machine learning community, the multitask learning (MTL) approach proposed in [1] attempts to learn multiple related tasks simultaneously. Within the context of neural network training, MTL can be implemented using a network sharing both its input features and hidden units but having multiple, task-specific output layers. Because the learning of one task may help learning the other tasks better, MTL was shown to learn a better hidden shared representation of the data and to improve generalization [1].

In speech recognition, state-of-the-art systems typically rely on some form of deep neural networks [2] to predict the posterior probabilities of context-dependent (CD) states given the acoustic feature vectors. Such acoustic models are then amenable to MTL training, requiring the definition of a main primary task and one or several related secondary tasks. In [3, 4, 5, 6, 7] a deep neural network (DNN) predicts context-dependent states as its primary task. The secondary task defines a second set of outputs to predict either context-independent (CI) states [6, 7], phone labels [3, 5], state or phone context [3], articulatory context [6], or gender targets [8]. In all those approaches but [7] the secondary outputs are discarded after the DNN model is trained and the recognition procedure only uses the primary outputs (providing an estimate of the CD state posteriors). In contrast, in [7], the network is structured such that the secondary CI state outputs are also used to predict the primary outputs, and in that sense, both primary and secondary outputs are used at run time. Similarly, in the approach proposed by [4] where the secondary task aims at predicting tri-grapheme classes, both the primary and secondary outputs are used at recognition time though independently: the CD state posteriors are fed to a triphone-based decoder while the tri-grapheme posteriors are used by a grapheme-based decoder. The recognition hypotheses provided by the two systems are then combined with ROVER [9].

In [10], we constructed an ensemble of acoustic models independently trained to predict CD states from different state inventories (e.g. a 12k CD states and a 1k CD states inventories). Rather than running multiple independent recognition systems and combining the resulting hypotheses post-recognition (e.g. using ROVER), we proposed instead to construct a single recognition system integrating the acoustic scores of the multiple systems at recognition time. To further reduce the computational complexity and avoid running inference in multiple neural networks simultaneously, we also proposed to share the input and hidden layers of those models, leading to a MTL architecture where both the primary and secondary outputs would predict CD states but based on distinct state inventories. As a result, in contrast with the MTL procedure of [3, 5, 6] which discards the secondary outputs after training, we used all primary and secondary outputs, treating them as an ensemble of classifiers. Unlike the approach in [4] which uses the primary and secondary outputs independently, we integrated all outputs at recognition time with a single recognition procedure using early acoustic score combination [11, 12].

The model in [10] were trained using a cross-entropy criterion. Because the acoustic scores from the primary and secondary outputs are combined at recognition time using a score combination function that may not be differentiable (e.g. maximum score combination), the model cannot be directly trained using a sequence training criterion. In this paper, we expand the work from [10] to enable sequence training of such models. We will show that by moving the derivation of the acoustic score of a tuple of CD states out of the search procedure onto the neural network and by defining a specialized output layer with an inventory of CD state tuples called meta-states, we can reformulate MTL training as a single task training procedure on output labels designed to encode the multi-task labels. As a result, those models can be trained using a traditional sequence training procedure [13].

We report experimental results on a large scale Voice Search task using convolutional, long short-term memory deep neural network (CLDNN) [14] trained on 2,100 hours of data and evaluated on child speech [15].

2. SYSTEM COMBINATION AND MULTI-TASK LEARNING

2.1. System combination

In our previous work [10], we showed that the performance of a large-scale production-quality CLDNN model [14] could be improved by using an ensemble of CLDNN models trained on distinct CD states inventories constructed using randomized decision trees [16]. Specifically, we found that the different systems trained on those state inventories had near identical performance but would

lead to significant reduction of the word error rate, ranging from 5% to 7% relative depending on the test set, when combining their outputs using ROVER. To alleviate the computational load of running multiple decodings in parallel, we proposed to integrate the state-level acoustic scores of each system at run time using a single recognition procedure operating on a decoding graph defined on tuples of CD states from the multiple systems[1]. In the context of a finite state transducer (FST) based decoder, this only requires the construction of a specialized context-dependency transducer while keeping the decoding graph construction procedure unchanged. This approach is essentially similar to using tree arrays [17] in dynamic decoders which enable combining multiple acoustic models with different state inventories in a single decoding procedure by defining a virtual tree holding in each leaf a unique combination of leaves from the individual trees. Note that a similar recognition procedure was recently proposed in [18] but using multiple CD state inventories constructed to encourage diversity rather than relying on randomization.

Such a decoding approach corresponds to defining meta-states, also called virtual states in [18], which represent tuples of CD state symbols from the multiple state inventories that occur in identical phonetic contexts. Since the resulting decoding network is defined on meta-states labels, the search requires the derivation of the likelihood $p(X|\langle CD_i^1, CD_j^2\rangle)$ of an acoustic feature vector X for a given meta-state $\langle CD_i^1, CD_j^2\rangle$, here assuming that the meta-state only involves 2 sets of CD states, where CD_i^1 and CD_j^2 refers to the i-th (resp. j-th) CD state of the 1st (resp. 2nd) state inventory. This can be defined as a function of the likelihood of X for each individual CD state component, i.e. $p(X|\langle CD_i^1, CD_j^2\rangle) = f(p(X|CD_i^1), p(X|CD_j^2))$. When those likelihoods (or pseudo likelihood obtained by scaling the states posteriors by the state priors in the case of a neural network system) are represented as negative log-likelihood, or cost, the combination function $f()$ can be defined for example to select the minimum cost or to compute the average cost from the individual states of the tuple.

2.2. Multi-task learning

Unfortunately, such an approach remains expensive in a production setup as it requires running inference through multiple large acoustic models, in our case CLDNN models, not to mention training multiple independent models. To retain the benefit of exploiting multiple state inventories but at a reduced computational cost, we suggested in [10] to merge the input and hidden layers of all the models of the ensemble and only keep distinct output layers. This is equivalent to the MTL learning paradigm [1] where the network is expected to learn a shared representation of the input data for the different CD state classification tasks. However, unlike many of MTL approaches which discard the secondary outputs after training [3, 5, 6], we use all outputs at decoding time, retaining the early score combination described above within a single decoding procedure. The resulting system architecture involves running inference through a single MTL model.

One issue with this approach is that the MTL model is not directly amenable to be optimized using a sequence-training criterion with our standard procedure [13]. This is because the decoder operates on a specialized graph of meta CD states, while the neural network outputs the posterior probabilities of each state inventory in different output layers. In addition, the predefined minimum cost combination function $f()$ used in [10] is not differentiable, preventing back-propagating the gradient of the sequence-based loss function.

[1]Note that for efficiency reasons our decoding graph is constructed at the HMM level and HMM arcs are dynamically expanded into CD state arcs during the search. For the sake of the discussion, we will here assume that the graph is fully expanded to CD state arcs.

System	# CD states	# HMMs
#1	12,000	43,538
#2	6,000	29,456
#3	6,000	33,814
#4	2,000	11,414
#5	2,000	13,487
#6	2,000	13,893
#7	2,000	12,166
#8	1,000	4,461
Meta(#1, #8)	18,352	47,533
Meta(#2, #3)	25,567	50,356
Meta(#4, #5, #6, #7)	33,734	51,947

Table 1. Number of CD state and HMM symbols for 8 individual systems of various sizes corresponding number of meta CD states and meta HMM symbols obtained by constructing a meta-C transducer combining systems (#1, #8), systems (#2, #3) and systems (#4, #5, #6, #7).

2.3. Meta-state model

To alleviate the issues above, we propose to move the score combination from the search procedure onto the neutral network itself. This implies adding an extra output layer, as represented in Fig. 1 (b), constructed to provide an estimate of the posterior probability of a meta CD state, as well as estimating the meta CD state priors to properly enable the derivation of the pseudo meta-state likelihood. We considered 2 implementation choices, as represented in Fig 1 (b) and (c). In the first one, an extra output layer is constructed such that the node corresponding to the meta CD state $\langle CD_i^1, CD_j^2\rangle$ is only connected to node CD_i^1 from the first output and node CD_j^2 from the second output, enabling the learning of the score combination. However, we simplified this architecture even more by dropping the distinct inventory-specific output layers to adopt the architecture of Fig. 1 (c) which corresponds to learning a hidden representation to directly predict $p(\langle CD_i^1, CD_j^2\rangle|X)$, an encoding of the multi-target CD state outputs.

It should be noted that the meta CD state inventory does not correspond to the full Cartesian product of the individual CD state inventories but only to a small subset since by design, a meta CD state is a tuple of individual CD states occurring in the same phonetic context. We report in Table 1 the size of a few meta CD state inventories given the size of the CD state inventory of the individual systems. For example, the combination of a system with 12k CD states with a system with 1k CD states leads to a 18k meta CD states inventory, while the combination of 4 $\times$ 2k CD states systems leads to a 34k meta CD states system. Note that for a given meta CD state inventory size, one cannot construct a regular single decision tree of similar size that will provide the same labeling of a given word sequence. That is, no single system of 18k CD states can lead to the same labeling as the one obtained by integrating the 12k and 1k systems onto their corresponding 18k meta-state labels.

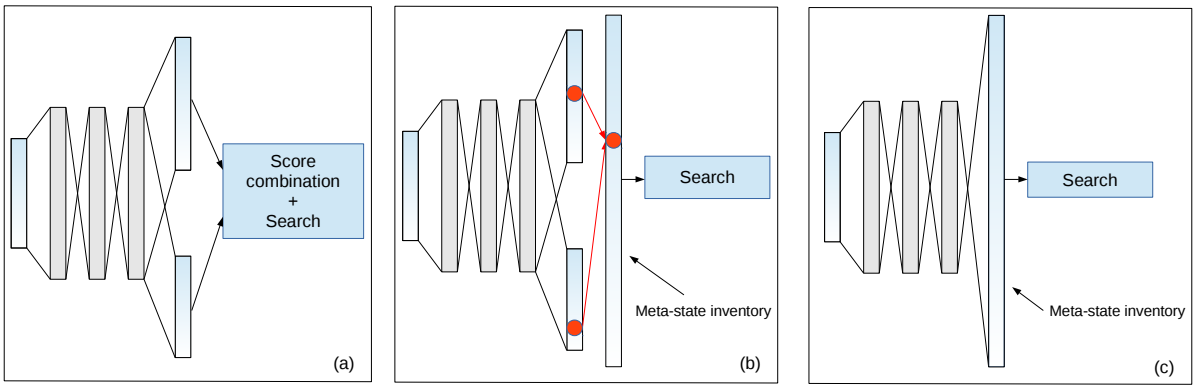

Fig. 1. (a) Multi-task training with multiple output layers defined on distinct CD state inventories and score combination carried out during search. (b) Adding a meta-state layer to perform CD state score combination during inference rather than search (c) Direct training of a model operating on meta-state labels encoding multiple outputs from the individual CD state inventories of (a).

3. EXPERIMENTS AND RESULTS

3.1. Task and Database

All experiments are carried out on a Voice Search (VS) task targeted at young speakers to support applications such YouTube Kids [19]. The training set consists of a mix of adult and child speech, in the order of 1.9M VS utterances from children and 1.3M utterances from the general VS traffic, totaling 2,100 hours of speech. We used 2 test sets for our evaluations, a set of 16k utterances from children and another of 25k utterances from adult speakers. Our training and test sets were manually transcribed and in accordance with our data retention policy, all data sets were anonymized. Further details on how the data sets were constructed are available in [15].

We trained all our acoustic models from scratch in multiple stages following the procedure described in [20]. We started by flat-starting a context-independent DNN model that was then used to construct a set of context-dependent states using Chou's partitioning algorithm modified to support randomization of the clustering procedure. We then trained a large context-dependent DNN model consisting of 8 hidden layers of 2560 nodes with cross-entropy before refining the model with sequence training. That procedure was used to train the 8 DNN models listed in Table 1 with CD states inventories ranging from 1k to 12k states. Note that the systems with identical number of CD states were constructed by randomizing the decision tree procedure to obtain different CD states inventories. Experimental results reported in [10] illustrate that this procedure is effective to construct a set of systems that can be combined either post or during recognition to improve performance. For reference, our production child-speech system uses about 13k CD states. Those models were then used to generate alignments used to bootstrap the training of CLDNN acoustic models, using either the MTL architecture of Fig 1 (a) operating on multi-outputs labels, or the architecture of Fig 1 (c) operating on meta-state labels. The topology of the CLDNN models is described in details in [15] and the models constructed in this paper only differ in terms of the number of state labels in the softmax output layers.

3.2. CLDNN training on meta-state labels

The DNN models constructed based on the procedure described above were then used to realign and label the training data with tuples of CD states for different system combinations. We considered 3 system configurations, the first one combining a 12k with a 1k CD states inventories, the second combining 2 $\times$ 6k CD states inventories, and the last one combining 4 $\times$ 2k CD states inventories. Those alignments were used to derive the prior probabilities of the CD state tuples defining the meta CD state inventory.

Given the tuples of CD states, we first trained a CLDNN model similar to the architecture of Fig 1 (a) using a cross-entropy criterion. Results are reported in Table 2 and for each trained MTL model, we ran recognition using either each individual output layer or by combining the acoustic costs of each output layer using a minimum cost combination. In all cases, decoding by combining the scores from the multiple output layers provides only marginal improvement over decoding from only one of the output layers. In addition, we observed in separate experiments that single-task training of a 6k CD states system and of a 12k CD states led in both cases to 10.0% WER on the Child test set. This indicates that the multi-style training paradigm does not lead to any significant improvements over single-task training, unlike what is reported in [5]. We hypothesize that MTL training is mostly effective when using smaller amount of training data but provides diminishing return with increasing amount of data. Those results also confirm that large CLDNN models are not very sensitive to the size of the CD state inventory, as the 6k and 12k CD states model delivered similar performance.

It should be noted however that when the sizes of the output layers differ significantly, as for example when training an MTL model with a 12k and 1k output layers, MTL training provides a regularization effect: decoding using only the 1k CD state output gives 10.9% WER on Child while a single-task model with 1k CD states gives 11.8% WER. Conversely, decoding using only the 12k CD state output of the MTL model gives 10.2% WER on Child, while a single-task 12k CD states model gives 10.0% WER. In other words, MTL training improves the quality of the 1k CD state output layer over single-training a 1k CD state model, but slightly degrades the quality of the 12k CD state output over single-training a 12k CD state model.

The results also indicate that reducing the size of the output layers to 2k CD states noticeably degrades performance, despite increasing the number of output layers to 4. This is consistent with separate experiments where we observed a degradation in performance using single-task systems with state inventories below 4k CD states.

Next, we trained a CLDNN model following the meta-state ar-

System	Child	Adult
Softmax#2 (6k states)	10.0%	12.6%
Softmax#3 (6k states)	10.1%	12.7%
MinCost(#2,#3)	9.9%	12.5%
Softmax#1 (12k states)	10.2%	12.8%
Softmax#8 (1k states)	10.9%	13.6%
MinCost(#1, #8)	10.1%	12.6%
Softmax#4 (2k states)	10.7%	13.2%
Softmax#5 (2k states)	10.7%	13.3%
Softmax#6 (2k states)	10.8%	13.5%
Softmax#7 (2k states)	11.0%	13.5%
MinCost(#4, #5, #6, #7)	10.6%	13.5%

Table 2. Cross-entropy MTL training for various state inventories and combinations for the system architecture in Fig. 1 (a). Top part: MTL system with 2 outputs layers of 6k states each (state inventory #2 and #3 from Table 1). Middle part: MTL system with 2 outputs layers of 12k and 1k. Bottom part: MTL system with 4 outputs layers of 2k states each. The 'Softmax' lines refers to decoding with a single output from the MTL model. The 'MinCost' line refers to decoding using minimum cost combination.

chitecture of Fig. 1 (c) and operating on multiple configurations corresponding to the last 3 lines of Table 1, that is, a first system with a meta-state layer constructed from a 12k/1k state tuple, another with a meta-state layer constructed from $2 \times$ 6k state tuples, and the last one from $4 \times$ 2k states tuples. The models were first trained using cross-entropy training and then refined with sMBR sequence-training. Results are given on Table 3. One can observe that the meta-state does not match the performance of the MTL training which we attribute to the much larger size of the softmax output layer and a propensity to overtrain. Nevertheless, the approach enables sequence training the model providing a significant improvement over the CE MTL model. We expect that on applications with a smaller amount of training data where the effectiveness of MTL training was demonstrated [5, 6], the proposed use of meta-state label can facilitate sequence-training the model.

System	Child	Adult
MTL (6k, 6k)	9.9%	12.5%
Meta (6k, 6k)	10.3%	12.7%
Meta seq (6k, 6k)	8.7%	11.6%
MTL (12k, 1k)	10.1%	12.6%
Meta (12k, 1k)	10.8%	13.0%
Meta seq (12k, 1k)	8.7%	11.5%
MTL (2k, 2k, 2k, 2k)	10.6%	13.5%
Meta (2k, 2k, 2k, 2k)	11.4%	13.9%
Meta seq (2k, 2k, 2k, 2k)	9.5%	12.5%

Table 3. Multitask (MTL) and meta-state training for various state inventories and combinations.

4. CONCLUSION

Multi-task learning traditionally involves learning a primary and secondary task jointly, but only uses the primary outputs at recognition time. In contrast, we propose in this paper a MTL approach to learn multiple CD state inventories in which all outputs are used at run-time by combining them using an integrated decoding procedure operating on tuples of CD states occurring in identical phonetic contexts. To support training such a model architecture using a sequence-based criterion, we propose to move the score combination into the neural network, which can be further simplified by designing the output layer to directly predict meta-state labels. We have found that while this procedure effectively enables the training of a model operating on a large meta-state inventory, it does not significantly outperform a baseline single task system. In particular, we note that in contrast with the results presented in [5, 6], MTL training only provides marginal improvements and speculate that this is due to the large amount of training data used in our experiments. Nevertheless, we believe that when the amount of training data is limited [5, 6], MTL training can improve performance and that the proposed approach enables sequence-training such models.

5. REFERENCES

[1] Rich Caruana, *Multitask learning*, Ph.D. thesis, Carnegie Mellon University, 1997.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Michael Seltzer and Jasha Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[4] Dongpeng Chen, Brian Mak, Cheung-Chi Leung, and Sunil Sivadas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[5] Peter Bell and Steve Renals, "Complementary tasks for context-dependent deep neural network acoustic models," in *Conference of the International Speech Communication Association (InterSpeech)*, 2015.

[6] Peter Bell and Steve Renals, "Regularization of context-dependent deep neural networks with context-independent multi-task training," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 2015.

[7] Pawel Swietojanski, Peter Bell, and Steve Renals, "Structured output layer with auxiliary targets for context-dependent acoustic modelling," in *Conference of the International Speech Communication Association (InterSpeech)*, 2015.

[8] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. H. Tham, P. Green, and V. Wan, "Multitask learning in connectionist speech recognition," in *Proc. Australian International Conference on Speech Science and Technology*, 2004.

[9] Jonathan G. Fiscus, "A post-processing system to yield reduced word error rates:recognizer output voting error reduction (ROVER)," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 1997.

[10] Olivier Siohan and David Rybach, "Multitask learning and system combination for automatic speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, To appear. 2015.

[11] Peter Beyerlein, "Discriminative model combination," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998, vol. 1, pp. 481–484.

[12] Andras Zolnay, *Acoustic Feature Combination for Speech Recognition*, Ph.D. thesis, RWTH Aachen-University, 2006.

[13] Georg Heigold, Erik McDermott, Vincent Vanhoucke, Andrew Senior, and Michiel Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[14] Tara N. Sainath, Oriol Vinyals, Andrew Senior, and Hasim Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[15] Hank Liao, Golan Pundak, Olivier Siohan, Melissa Carroll, Noah Coccaro, Qi-Ming Jiang, Tara Sainath, Andrew Senior, Francoise Beaufays, and Michiel Bacchiani, "Large vocabulary automatic speech recognition for children," in *Conference of the International Speech Communication Association (InterSpeech)*, 2015.

[16] Olivier Siohan, Bhuvana Ramabhadran, and Brian Kingsbury, "Constructing ensembles of ASR systems using randomized decision trees," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2005.

[17] Hagen Soltau, George Saon, and Brian Kingsbury, "The IBM Attila speech recognition toolkit," in *IEEE Workshop on Spoken Language Technology (SLT)*, 2010.

[18] Hainan Xu, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Modeling phonetic context with non-random forests for speech recognition," in *Conference of the International Speech Communication Association (InterSpeech)*, 2015.

[19] "Introducing the newest member of our family, the YouTube Kids app—available on Google Play and the App Store," 2015, http://youtube-global.blogspot.com/2015/02/youtube-kids.html.

[20] Michiel Bacchiani, Andrew Senior, and Georg Heigold, "Asynchronous, online, GMM-free training of a context dependent acoustic model for speech recognition," in *Conference of the International Speech Communication Association (InterSpeech)*, 2014.