Improvements to the Pruning Behavior of DNN Acoustic Models

Matthias Paulik

Apple Inc., 1 Infinite Loop, Cupertino, CA 95014

mpaulik@apple.com

Abstract

This paper examines two strategies that improve the beam pruning behavior of DNN acoustic models with only a negligible increase in model complexity. By augmenting the boosted MMI loss function used in sequence training with the weighted cross-entropy error, we achieve a real time factor (RTF) reduction of more than 13%. By directly incorporating a transition model into the DNN, which leads to a parameter size increase of less than 0.017%, we achieve a RTF reduction of 16%. Combining both techniques results in a RTF reduction of more than 23%. Both strategies, and their combination, also lead to small but statistically significant word error rate reductions.

Index Terms: speech recognition, DNNs, acoustic modeling

1. Introduction & Related Work

In voice enabled applications, such as Siri, user experience is influenced by both the accuracy and latency of the underlying large vocabulary continuous speech recognition system. Unfortunately, these two optimization criteria often display an inverse correlation. For example, a more aggressive pruning beam typically improves the real time factor (RTF) of the speech recognition system, but it also typically increases the word error rate (WER). And while a more complex acoustic model (AM) might improve the WER, it often results in an increased RTF, due to an increase in the computational need for likelihood estimation. However, there are cases where a more complex AM can significantly reduce the overall RTF, despite the need to spend more time in likelihood computation. In such cases, search (Viterbi decoding) is sped up because the 'sharper' AM allows pruning of incorrect hypotheses much earlier in search.

In this paper we are investigating two strategies that are aimed at improving the general pruning behavior of DNN acoustic models [1, 2, 3, 4, 5], without increasing the model complexity (number of parameters). By general pruning behavior we mean that we do not adapt the DNN AM to a specific task or speaker [6, 7, 8] to achieve any speedups. While AMs that display a better pruning behavior often times also yield better WERs when decoding with the same beam pruning thresholds, we do not specifically seek such WER improvements. However, both techniques described in this paper result in small, but consistent and statistically significant improvements in WER.

Beam pruning finds the score s of the best scoring state at time t and prunes from the active search space all states whose scores are worse than b times s. The parameter b thus controls the width of the beam. It is obvious that the better the distribution of scores over all the active states at time t approximates to a concentration on a single 'correct' state the more effectively beam pruning will work. Given a hard labeling of the frames, the usual cross-entropy provides a measure of such concentration, or 'sharpness', for an AM. Thinking in these terms, it seems that frame level cross-entropy training of DNN AMs should yield optimally sharp models. However, this formulation naturally ignores how we construct the search space during decoding. Both language model and HMM topology influence which acoustic states are active at any given frame in Viterbi decoding with beam pruning. One could argue that lattice based sequence training [9, 10] of DNN AMs addresses this issue, and in fact sequence training typically yields significant WER improvements over cross-entropy training. However, as we will see in Section 3, at identical pruning thresholds, we can observe a worse pruning behavior for sequence trained models compared to cross-entropy trained models. We use the boosted maximum mutual information (bMMI) criterion [11] in the sequence training stage. To counter the negative effect on pruning behavior of sequence trained DNNs, we propose to add the weighted cross-entropy error to the bMMI loss function, similar to [12]. However, in contrast to [12], we provide a detailed analysis of the influence this approach has on Viterbi decoding with beam pruning. We will show that this approach can speed up decoding significantly.

It is well known that beam pruning interacts with word and phone transitions due to the associated fan-out at such transition points. A stronger transition model (TM) might help to reduce confusion about when to cross into a new phone as opposed to staying within the current phone. To this end, we propose the incorporation of a simple transition model directly into the DNN acoustic model. We are not aware of any previous work that attempts anything similar. We incorporate the transition model into the DNN acoustic model by adding a small number (four) of output targets to the DNN and dividing the output layer during training into two regions, one corresponding to the clustered tri-phone state targets and one corresponding to the aforementioned four transition model targets. This approach hardly increases the total amount of parameters in our DNN at all – the total parameter size increase is less than 0.017%. More details on the proposed transition model are given in Section 4. Adding the transition model to the DNN acoustic model yields another significant improvement in RTF, because of favorable pruning effects.

The remainder of this paper is organized as follows. Section 2 describes our experimental setup and discusses how performance is measured. In Section 3, we take a closer look at how sequence training influences the pruning behavior of our acoustic models, and we show results for smoothing the sequence training objective function with the frame level cross-entropy error. Section 4 gives a detailed description of our

standard transition model and of the newly proposed transition model, which is directly integrated into the DNN acoustic model. Section 5 presents WER/RTF trade-off curves and the final results on our evaluation set. In Section 6 we discuss our results and we conclude with a short summary in Section 7.

2. Experimental Setup

2.1. Data Sets

For acoustic model training, we use 1,200 hours of manually transcribed, US English audio data. 30 hours of that training set is held-out for cross evaluation purposes, i.e. to adjust the learning rate and the number of iterations in DNN training. Our language model is estimated from a very large, automatically transcribed speech corpus. Our development (dev) and evaluation (eval) sets each comprise 10 hours of audio data. All the data sets used in the work described here were anonymized.

2.2. Baseline System and Performance Measurements

Weighted Finite State Transducer (WFST) based speech recognition systems [13, 14, 15, 16] have gained great popularity over the last decade. For our experiments, we use a WFST based decoder employing the difference LM principle, similar to [17]. The language models are class-based and the decoder uses on-the-fly compiled, user dependent intra-class language models to allow for user specific vocabularies. We trained a baseline DNN AM, first using frame level cross-entropy training, followed by boosted MMI sequence training. The input to this DNN consists of global mean normalized, spliced filter bank features of dimension 40. We use a splicing of -12/+6 frames, resulting in an overall input dimension of 760. The DNN has 6 hidden layers with 1024 sigmoid activation functions each. The last hidden layer is connected to the 10,201 dimensional output layer (clustered tri-phone state targets) via a 512 dimensional linear bottleneck layer. The bottleneck layer helps to reduce the overall parameter size of the DNN, which comes to 11.7852 million parameters. The decoding dictionary has 523.6K entries and the entropy pruned 4-gram language model has 16 million entries.

All RTF numbers reported below are computed on the author's desktop (an Apple iMac), over a 300 utterance subset extracted from the dev set. We arrive at these RTF values by averaging over RTF values obtained from decoding that subset three times. Our RTF computation does not consider the complete dev set and suffers from some minor noise due to background processes. However, as we will see below, the reported RTF values correlate very well with the average amount of active tokens (AT) per frame, which is always computed on the complete data set under consideration and is therefore an accurate measurement.

3. X-Entropy Error & Sequence Training

Table 1: *XEnt and bMMI training (dev set)*

	WER	RTF	AT	FA	FA_c
XEnt	7.8	0.16	2023	62.3	65.5
bMMI	7.1	0.174	2185	52.6	56.8
bMMI+XEnt	7.0	0.15	1818	60.1	62.9

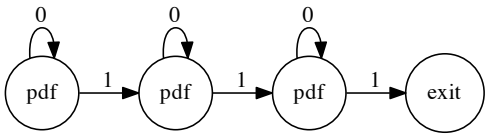

Figure 1: *3-state Bakis topology with non-emitting exit state*

Table 1 lists the WER of our baseline DNN AM on the dev set after cross-entropy training (XEnt) and sequence training (bMMI). All decoding runs shown in the table use the same pruning thresholds. The table also shows the RTF values, and the average AT counts per frame. Note first that sequence training results in a strongly improved WER, but slightly worse RTF. Given that the the parameter size of the DNN is unchanged, i.e. the time spend in feed-forward remains constant, any degradation in RTF has to be attributed to time spend in Viterbi decoding. This observation is supported by the increase in AT. The last columns of Table 1 show the frame accuracy (FA) on our 30 hour cross evaluation set. We compute the FA in two ways, once using the initial training alignments and once using alignments computed with the current, newly trained DNN (FA_c). Perhaps not surprisingly, optimizing towards the bMMI loss function results in an increased cross-entropy error, which in turn leads to a degradation in frame accuracy[1]. As already argued in the introduction, it seems plausible that the average frame accuracy interacts with beam pruning. We therefore experiment with augmenting the bMMI loss function with the cross-entropy error:

$$L_{bMMI+XEnt} = L_{bMMI} + w * L_{XEnt}$$

The third row in Table 1 lists the result when weighting the cross-entropy error by $w = 0.5$. The WER is reduced by 0.1% absolute; a small, but statistically significant ($p = 0.95$) change. More interestingly, we observe a reduction in active token count of 16.8% relative, which translates into a reduction in the RTF of 13.8% relative.

4. A Simple DNN Transition Model

We use two HMM topologies in our acoustic model: a typical 3-state Bakis topology without skip transitions and a 4-state topology with skip transitions. Both these topologies have an additional, final, non-emitting exit state, as depicted in Figure 1. Each emitting state has two transitions in the 3-state topology and four transitions in the 4-state topology. Each transition can be uniquely identified by the state identifier of the emitting state together with the index i of the transition, with $i \in [0, 1]$ or $i \in [0, 1, 2, 3]$ depending on the topology. The standard transition model is a simple maximum likelihood estimate over the count statistics for how frequently we see each transition when doing Viterbi decoding in training. The transition probabilities from the standard TM are directly represented in our WFST decoding graph.

On top of the standard transition model, we propose to make use of another, much simpler transition model that is directly combined with the DNN acoustic model. We propose to extend the output layer of our DNN acoustic model by four

[1] We refer the reader to Section 6 on this topic.

additional targets encoding the transition index $i \in [0, 1, 2, 3]$. In training, we divide the output layer into two regions, one corresponding to the clustered tri-phone state targets (pdf index) and one corresponding to the aforementioned four transition model targets. For back propagation, we compute two independent error values, one for each region, and then back propagate the weighted sum of both. Note that this approach does not treat speech frames that belong to a state from the 3-state topology any different than states that belong to the 4-state topology and that any correlation between pdf index and transition index has to be learned implicitly by the DNN. Nevertheless, we observe an average transition index prediction accuracy of more than 80%. Almost half of all the speech frames in our training data correspond to states from the 4-state topology.

During decoding, as well as alignment and lattice generation for training, we compute the acoustic score from the DNN logit values (the pseudo log likelihoods before the softmax activation) in the following way:

$$score_{AM} = acwt * (logit_{pdf\_i} + tmwt * logit_{trans\_i})$$

That is, we multiply the logit value of the DNN output corresponding to a specific transition index by a global transition model weight $tmwt$ and add the resulting value to the logit of the clustered tri-phone state under consideration. This sum is weighted by the global acoustic model weight $acwt$.

The rows marked with TM in Table 2 list the results obtained on the dev set when using a DNN with the integrated transition model. We use a transition model weight of $tmwt = 1.0$ during decoding. As in previous experiments, all results are obtained by running the decoder with exactly the same pruning values. Note that using the proposed transition model already has a positive impact in the frame level cross-entropy training stage: both, WER and RTF/AT are reduced. The same trend can be observed for the bMMI sequence trained AM. An even stronger reduction in RTF and active token count can be seen when the cross-entropy error is once again added to the bMMI loss function. Overall, we observe a relative reduction in the average number of active tokens per frame of more than 30%, compared to the bMMI sequence trained baseline system. This reduction in AT corresponds to a 23% relative reduction in RTF[2]. In addition to the reduction in RTF, we obtain a small, but statistically significant ($p = 0.95$) reduction in WER.

Table 2: *DNN transition model (dev set)*

	WER	RTF	AT
XEnt	7.8	0.16	2023
TM, XEnt	7.6	0.153	1802
bMMI	7.1	0.174	2185
TM, bMMI	7.0	0.146	1756
TM, bMMI+XEnt	7.0	0.133	1474

[2]Note that all RTF values include the constant overhead from DNN feedforward computation.

5. Final Results

So far, we have explored the performance of the techniques presented only for one specific 'operating point', i.e. one particular beam pruning value. Figure 2 shows how the WER varies in relation to the RTF for the techniques presented. The plot was obtained by computing the WER/RTF values at different beam pruning settings $b \in [9.0, 9.5, ..., 13.5, 14.0]$. Figure 3 was obtained in the same manner, but lists the average number of active tokens on its x-axis. The plots look virtually identical. This not only demonstrates how well RTF and AT correlate, but also gives a clear indication of the positive impact the techniques presented have in combination with beam pruning. Overall, we can see that both techniques individually result in approximately the same WER/RTF behavior and that by combining the techniques, a superior WER/RTF trade-off can achieved.

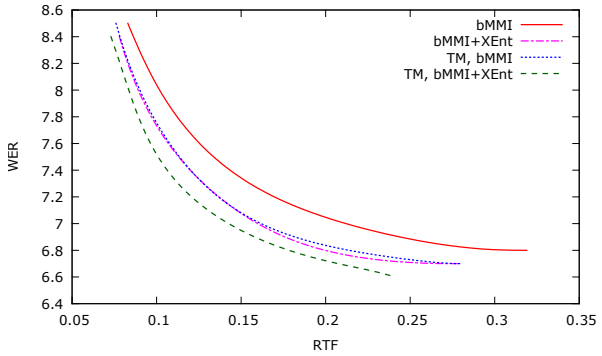

Figure 2: *WER vs. RTF (dev set)*

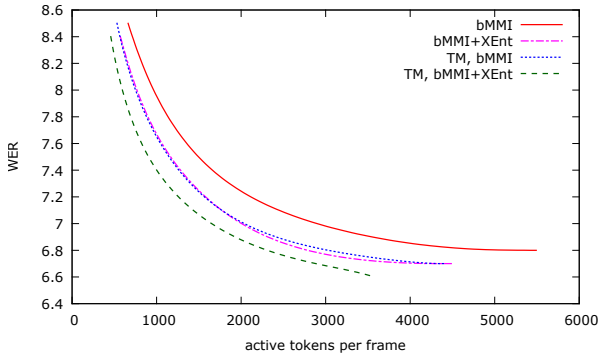

Figure 3: *WER vs. AT (dev set)*

Table 3 lists the final results on the 10-hour evaluation set at our preferred operating point. Given the availability of the accurate measure of average active token counts per frame, we omitted the somewhat tedious computation of RTF values. We see the same behavior as observed on our development set. Both techniques independently achieve approximately the same reduction in AT at a slightly improved WER. Combining both techniques yields the best result, with a relative reduction in AT of more than 32% and a relative WER reduction of 2.9%.

Table 3: *Final results (eval set)*

	WER	AT
bMMI	6.9	2175
bMMI+XEnt	6.8	1805
TM, bMMI	6.8	1744
TM, bMMI+XEnt	6.7	1459

6. Discussion

At first sight, the improvements in beam pruning behavior by adding the cross-entropy error to the bMMI loss function in sequence training seem intuitive: a sharper acoustic likelihood distribution between active acoustic states with different underlying pdfs should help pushing incorrect states outside the search beam. However, and as already indicated in the introduction, one could argue that lattice based sequence training should have the advantage of 'respecting' how we construct the search space during decoding. In this light, the disadvantage of the sequence trained models with respect to pruning behavior at identical pruning settings seems much less obvious, especially given the large improvements in the WER sequence training yields. In this context, we would like to quote [12], which refers to "the unavoidable sparseness of word lattices" as a motivation for smoothing the sequence training objective with the frame level objective. In contrast to [12], we give detailed results for the run-time behavior of models trained with a smoothed sequence training objective. Reference [12] simply cites the WER improvements compared to training without smoothing, and it remains unclear at what RTF the various decoding runs operate.

So far, all our experiments make use of the standard transition model, which is directly incorporated in the WFST decoding graph in the form of fixed graph costs. In order to examine the importance of the standard TM, we remove any transition model graph costs from the search graph and re-decode our dev set using our preferred operating point. Somewhat surprisingly, the WER remains unchanged. However, time spend in Viterbi decoding is strongly affected, as can be seen from the results in Table 4. For the bMMI trained baseline system, the number of active tokens more than doubles and even the system with the newly proposed DNN TM sees an increase in AT of 33% relative. Further, we note that without the standard TM, the DNN TM system runs at only a 7% relative increased AT count, compared to the bMMI baseline system with the standard transition model (2185 vs. 2337 active tokens). The results show that combining both transition models provides the best performance but that the simple DNN TM alone can provide a performance that is quite close to the standard TM.

Table 4: *Influence of the standard TM on AT (dev set)*

	with sTM	without sTM
bMMI	2185	4780
TM, bMMI	1756	2337

Finally, we wanted to take a closer look at the role of the DNN transition model weight $tmwt$. Given the cross-entropy trained DNN, we optimized $tmwt$ using a grid search. The resulting optimal value of $tmwt = 1.0$ was then used for any subsequent training and decoding runs. Whereas all of our RTF/AT trade-off curves presented so far were computed by varying the beam pruning value b at a constant transition model weight $tmwt = 1.0$, Figure 4 now shows the RTF/AT trade-off curve for our best available model when varying $tmwt \in [0.0, 0.5, ..., 6.0]$ at a constant beam value of $b = 11.5$. For comparison, the figure also shows the curves for various other models within the region of interest, once again obtained by varying the beam pruning value b at a constant transition model weight $tmwt$. Note that by varying the TM weight at a fixed beam pruning value, only a slightly better RTF/AT trade-off can be achieved within the region of between approximately 900 and 1500 active tokens per frame.

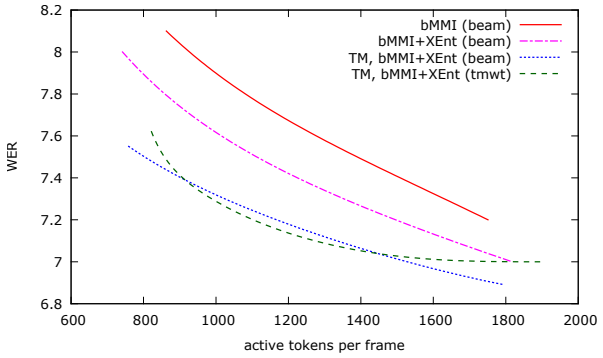

Figure 4: *WER vs. AT when varying tmwt (dev set)*

Our approach to learning clustered tri-phone state targets and transition model targets in parallel, using a shared underlying model can be viewed as a variation of the well-known multitask learning concept [18]. In this context, it should be noted that we observed degradations in accuracy compared to the baseline model, when decoding with the multi-task learned DNN and setting the transition model weight $tmwt$ to zero.

7. Summary

We have presented two strategies that improve the beam pruning behavior of DNN acoustic models, with only a negligible increase in the parameter size of the model. These methods are (A) smoothing the bMMI objective function with the frame level cross-entropy error; and (B) incorporating a simple yet effective transition model into the DNN acoustic model. Both methods improve the WER/RTF trade-off by reducing the average amount of active tokens per frame in Viterbi decoding with beam pruning. The techniques can be easily combined and their combination yields another significant improvement in WER/RTF trade-off.

8. Acknowledgements

The author would like to thank Henry Mason for valuable discussions. Thanks also go to the numerous other Siri speech team members, particularly Melvyn Hunt, who took the time to provide feedback and to carefully proofread this paper.

9. References

[1] Seide F., Li G., Yu D., "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks", Interspeech, 2011, Florence, Italy.

[2] Sainath T.N., Kingsbury B., Ramabhadran B., Fousek P., Novak P., Mohamed A., "Making Deep Belief Networks Effective for Large Vocabulary Continuous Speech Recognition", ASRU, December 2011, Big Island, Hawaii, USA.

[3] Dahl G., Yu D., Deng L., Acero A., "Context-Dependent Pre-Trained Deep Neural Networks for Large Vocabulary Speech Recognition", IEEE Trans. on Audio, Speech, and Language Processing, vol. 20, no.1, pp. 30-42, 2012.

[4] Mohamed A., Dahl G., Hinton G., "Acoustic Modeling using Deep Belief Networks", IEEE Trans. on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 14?22, 2012.

[5] Hinton G., Deng L., Yu D., Dahl G., Mohamed A.-R., Jaitly N., Senior A., Vanhoucke V., Nguyen P., Sainath T., Kingsbury B., "Deep Neural Networks for Acoustic Modeling in Speech Recognition", IEEE Signal Processing Magazine, 2012.

[6] Yu D., Yao K., Su H., Li G., Seide F., "KL-divergence Regularized Deep Neural Network Adaptation for Improved Large Vocabulary Speech Recognition", ICASSP, May 2013, Vancouver, BC, Canada.

[7] Saon G., Soltau H., Nahamoo D., Picheny M., "Speaker Adaptation of Neural Network Acoustic Models using I-Vectors", ASRU, December 2013, Olomouc, Czech Republic.

[8] Xiao Y., Zhang Z., Cai S., Pan J., Yan Y., "A Initial Attempt on Task-Specific Adaptation for Deep Neural Network based Large Vocabulary Continuous Speech Recognition", Interspeech, September 2012, Portland, OR, USA.

[9] Bridle J.S., Dodd L., "An Alphanet Approach to Optimising Input Transformations for Continuous Speech Recognition", ICASSP, April 1991, Toronto, ON, Canada.

[10] Kingsbury B., "Lattice-Based Optimization of Sequence Classification Criteria for Neural-Network Acoustic Modeling", ICASSP, April 2009, Taipei, Taiwan.

[11] Povey D., Kanevsky B., Kingsbury B., Ramabhadran B., Saon G., Visweswariah K., "Boosted MMI for Model and Feature-Space Discriminative Training", ICASSP, 2008, Las Vegas, NV, USA.

[12] Su H., Li, G., Yu D., Seide F., "Error Back Propagation for Sequence Training of Context-Dependent Deep Networks for Conversational Speech Transcription", ICASSP, May 2013, Vancouver, BC, Canada.

[13] Mohri M., Pereira F., Riley M., "Weighted Finite-State Transducers in Speech Recognition", Computer Speech and Language 16.1 (2002): 69-88.

[14] Moore D., Dines J., Magimai Doss M, Vepa J., Cheng O., Hain T., "Juicer: A Weighted Finite-State Transducer Speech Decoder", Machine Learning for Multimodal Interaction,Springer Berlin Heidelberg, 2006. 285-296.

[15] Dixon P. R., Oonishi T., Iwano K., Furui, S., "Recent Development of WFST-based Speech Recognition Decoder", Asia-Pacific Signal and Information Processing Association, October 2009.

[16] Povey D., Ghoshal A., Boulianne G., Burget L., Glembek O., Goel N., Hannemann M., Motlicek P., Qian Y., Schwarz P., Silovsky J., Stemmer G., Vesely K., "The Kaldi Speech Recognition Toolkit", ASRU, December 2011, Big Island, Hawaii, USA.

[17] Dolfing H., Hetherington , I., "Incremental Language Models for Speech Recognition using Finite-State Transducers", ASRU, December 2001 Madonna di Campiglio, Trento, Italy.

[18] Caruana R., "Multitask Learning", Ph.D. thesis, Carnegie Mellon University, September 1997.