MULTIFRAME DEEP NEURAL NETWORKS FOR ACOUSTIC MODELING

Vincent Vanhoucke, Matthieu Devin, Georg Heigold

Google, Inc., USA

ABSTRACT

Deep neural networks have been shown to perform very well as acoustic models for automatic speech recognition. Compared to Gaussian mixtures however, they tend to be very expensive computationally, making them challenging to use in real-time applications. One key advantage of such neural networks is their ability to learn from very long observation windows going up to 400 ms. Given this very long temporal context, it is tempting to wonder whether one can run neural networks at a lower frame rate than the typical 10 ms, and whether there might be computational benefits to doing so. This paper describes a method of tying the neural network parameters over time which achieves comparable performance to the typical frame-synchronous model, while achieving up to a 4X reduction in the computational cost of the neural network activations.

Index Terms— deep neural networks, acoustic modeling

1. INTRODUCTION

Deep neural networks (DNNs) have become increasingly popular for acoustic modeling [1]. They make it possible to effectively use many more parameters than typical Gaussian mixture models (GMMs) in several ways:

1. use a large number of *shared* parameters across states: while GMM parameters are only exercised when their associated state is active, DNN parameters up to the last hidden layers are shared across all states [2],
2. use wider windows of context: while GMM systems rarely benefit from using more than 10 frames (100 ms) of context around the central frame, DNNs benefit from 20 (200 ms) and up to 40 (400 ms),
3. use a larger number of output states: it has been observed that DNN systems can typically take advantage of a much larger number of output states than comparable GMM systems.

The larger number of parameters that need to be evaluated at every frame has the disadvantage of making real-time inference more computationally challenging. There are several ways to mitigate this problem. One is to use GPUs, which are very effective at handling large matrix computations. Another approach is to quantize the networks and use fixed-point computations [3]. Another is to distribute the computation across multiple cores, or even machines [4]. To go beyond that, one might have to consider limiting the size of the networks or exploring alternative architectures.

This paper introduces another approach which takes advantage of the stationarity of the speech signal, and ties neural network parameters across frames, enabling the acoustic model to be run at a reduced frame rate. Rather than separating the model description from the experiments, we will use the experiments to guide the rationale behind the approach: Section 2 describes the baseline system and shows the performance/complexity tradeoff of a typical frame-synchronous acoustic model. Section 3 describes a simple asynchronous approach which performs remarkably well. Section 4 introduces the proposed method, and shows that it compares advantageously to both baselines in terms of accuracy and complexity. Section 5 demonstrates the speedups that can be obtained using this technique.

2. HYBRID DEEP NEURAL NETWORK SYSTEM AND COMPUTATIONAL COMPLEXITY

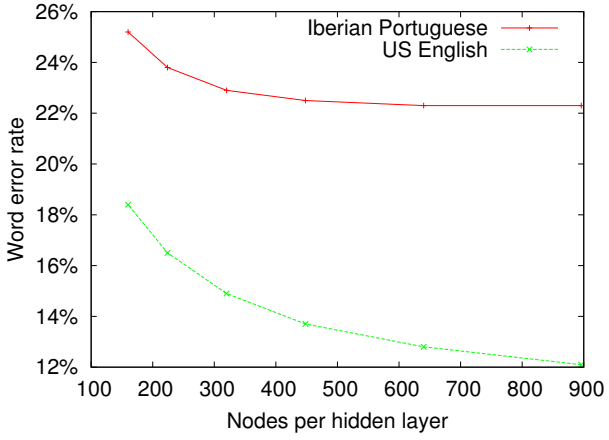

Fig. 1. Error rate against complexity of neural network acoustic models for US English, trained on thousands of hours of data, and Iberian Portuguese, trained on 100 hours of data.

For extensive background, a general introduction to hybrid DNN systems can be found in [1]. The goal of this paper is to look at the complexity/accuracy tradeoff of such a system. To explore this tradeoff, we trained a collection of systems of various complexities on two datasets: US English Voice Search [5] and voice typing, and Iberian Portuguese Voice Search, by varying the width of the hidden layers of the DNN acoustic model. The rest of the system was kept fixed: the frontend consists of 40 log-filterbank energies computed every 10 ms, stacked 20 frames in the past and 5 frames in the future to limit latency. The DNNs all have 3 hidden layers, sigmoid activations and 7969 softmax output classes for English (2960 for Portuguese), which are the leaves of a state-tying decision tree. They were trained using a distributed neural network infrastructure [4] using AdaGrad [6] and asynchronous parameter updates. The training data consists of more than 3000 hours of speech for English and approximately 100 hours for Portuguese. The evaluation is performed on 27327 held out utterances for English (11901 for Portuguese), using a fixed large-vocabulary language model. There are arguably many ways to sweep the algorithmic complexity of a system, and varying the width of the hidden layers is but one of them. Nevertheless, we find that this is an effective way to span a wide range of complexities without departing too far from the optimal operating point. Figure 1 shows how the word error rate of the evaluation set changes as we sweep the number of hidden nodes between 160 and 896 nodes per layer. Each datapoint corresponds to a doubling of the number of hidden parameters. It is evident from the graph that the performance falls off rapidly as the number of parameters decreases. The question is: can we do better?

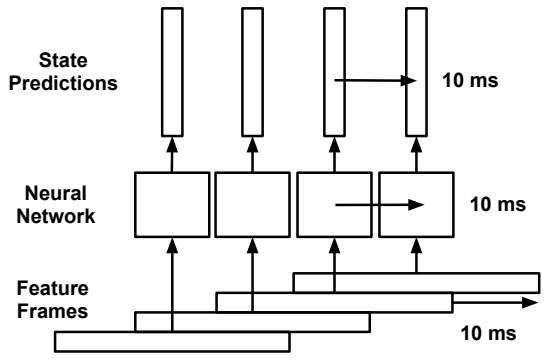

Fig. 2. Frame synchronous baseline approach: each acoustic model input is a window of multiple feature frames, shifted by one frame (10 ms) over time. A prediction is issued every input frame synchronously.

3. FRAME ASYNCHRONOUS MODEL

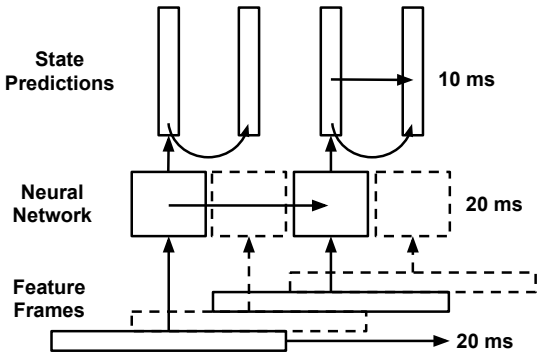

Fig. 3. Frame asynchronous approach, in the case of an acoustic model running at half the frame rate of the feature stream. Predictions are simply copied every other frame.

Speech is a rather stationary process when analyzed at a 10 ms frame rate. A much wider window of time (e.g. 275 ms in our experiments) is used to make a frame classification decision. The traditional approach is depicted in Figure 2: overlapping stacked frames are passed to the neural network to issue a prediction synchronously at every frame. It is natural to wonder whether one could simply use the predictions at time $t - K, k = 1, 2, \ldots$ to issue a prediction at time t. There are many models that attempt to take advantage of time correlations between feature frames. A naïve, but computationally inexpensive approach is to simply copy the predictions from previous frames as depicted in Figure 3. Since the alignments used to train the networks are inherently noisy, one can expect the neural networks to be very robust to alignments being off by several frames. This works surprisingly well, as illustrated in Figure 4: the graph depicts the performance of systems running the acoustic model at $1/2$ and $1/4$ the frame rate compared to frame-synchronous models of the same complexity. The approach is not novel, it has been used in GMM/HMM systems to trade off performance against speed, including more sophisticated variable-rate schemes [7, 8]. It is nonetheless interesting to note how well it performs in the context of a DNN on a very large task, and better so as the acoustic model and training data get larger. Note that because we copy the predictions for frame t to $t + 1$ (or $t + 1, t + 2, t + 3$ respectively), the decoder still runs at the same 10 ms frame rate in all cases. Based on the envelope of the resulting curves, it appears to always be a better tradeoff to oversize the network by a factor 2, and only compute acoustic scores every 2 frames. Computing acoustic scores every 4 frames did yield a better operating point on English, but not on Portuguese. Can we do even better?

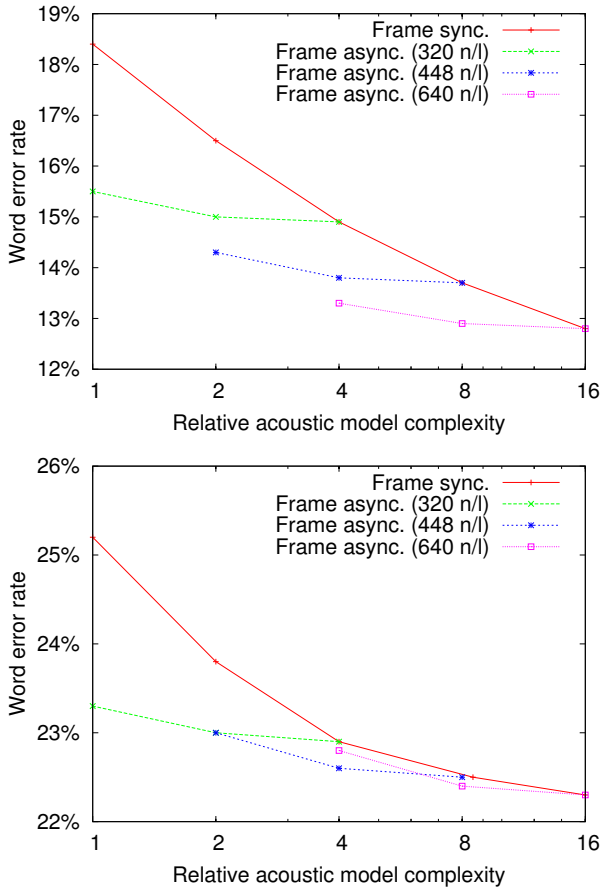

Fig. 4. Error rate against complexity of the neural network for US English (top) and Iberian Portuguese (bottom). Frame synchronous model complexity is controlled by doubling the network size between data points. Frame asynchronous models have a fixed size (320, 448 or 640 nodes per layer), but are computed every 1, 2 or 4 frames, resulting in complexities equivalent to, 1, 1/2 or 1/4 of the frame synchronous model complexity.

4. MULTIFRAME PREDICTION

Since the last layer of a DNN can be computed on-demand at decoding time and scores can be batched [3], there are fewer efficiency gains to be obtained from running the final layer of the DNN at a lower frame-rate. This suggests that training a DNN which shares all its hidden parameters, but uses frame-synchronous output layers might be a good tradeoff. The resulting architecture is depicted in Figure 5: the DNN has the same topology as our baseline system, but in addition to a softmax regression layer that predicts the frame label at time t, it also has an output layer trained jointly for labels $t-1$ up until $t-K$. In our experiments, we are looking at prediction of frames in the past $(t-1,\ldots,t-K)$, and not future frames $(t+1,\ldots,t+K)$, because the reference frame

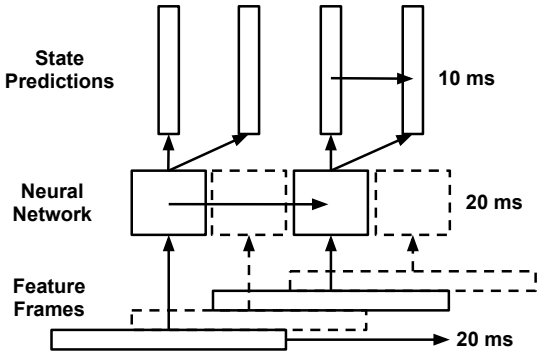

Fig. 5. Multiframe approach, in the case of an acoustic model running at half the frame rate of the feature stream. The neural network is trained to issue jointly a prediction for multiple consecutive frames.

has a much longer context window in the past (20 frames) compared to the future (5 frames), and hence past frames are provided a more balanced context than future frames. Note that this means that the effective input window for these predictions is $[t-20+K, t+5+K]$ instead of $[t-20, t+5]$. If K is large, this will have an impact on the overall latency of the system. In practice here, we will look at a worst case delay increase of 30ms. Training such DNNs can be performed by backpropagating the errors from both output layers jointly through the network, taking into consideration that due to the increased gradient magnitudes, the overall learning rate might have to be reduced. One interesting implementation point is that, for the decoder to operate in a frame-synchronous manner, it needs to presented first with the predictor for time t, followed by $t-K, t-K+1\ldots$.

Table 1 compares the performance of the asynchronous and multiframe prediction approaches. The complexity comparison overlooks the fact that one has more effective parameters in the output layers than the other. The cost of these extra parameters is in practice a small increment over the cost of the rest of the model, and is very much implementation and task-dependent. The outcome is consistent with expectations: using a frame-synchronous output layer improves performance. What is somewhat surprising is that the performance of the resulting system seems to consistently be as good as the baseline system. The small performance gains against the frame-synchronous baseline for some of the Portuguese experiments were not found to be statistically significant. It is possible that joint multiframe training can help regularize the training in the presence of noisy alignments, but so far the evidence of any such effect is inconclusive. In any case, this demonstrates that the multiframe prediction architecture can compete with frame-synchronous systems with far fewer parameters.

Table 1. Word error rates (%) for neural networks trained as multiframe predictors. Multiframe acoustic model's hidden activations are computed every 2 or 4 frames, resulting in complexities approximately equivalent to 1/2 to 1/4 of the frame synchronous model complexity.

		Nodes / layer	$K = 1$ (baseline)	$K = 2$	$K = 4$
US English	Frame async.	640	12.8	12.9	13.3
	Multiframe	640	12.8	12.9	13.3
	Frame async.	448	13.7	13.8	14.3
	Multiframe	448	13.7	13.7	13.9
	Frame async.	320	14.9	15.0	15.5
	Multiframe	320	14.9	14.9	15.0
Iberian Portuguese	Frame async.	640	22.3	22.4	22.8
	Multiframe	640	22.3	22.3	22.4
	Frame async.	448	22.5	22.6	23.0
	Multiframe	448	22.5	22.4	22.2
	Frame async.	320	22.9	23.0	23.3
	Multiframe	320	22.9	22.6	23.0

5. DECODING SPEED

We evaluated the performance of the approach on a server-based recognizer running a 7-layer, 2000 nodes/layer US English system and a large vocabulary language model. The system implements the multiframe architecture, but for the purpose of benchmarking, the same output layer was used for each time step. For a system of that size trained on a large amount of data, the performance gain from training distinct layers is negligible.

A system which predicts jointly 2 frames at a time achieved a 10% improvement in the query processing rate at no cost in accuracy or median latency, compared to an equivalent frame synchronous system. A system which predicts jointly 4 frames achieved a further 10% improvement in the query processing rate at a cost of a 0.4% absolute increase in word error rate. Both multiframe systems also exhibit much better tail latency characteristics.

6. ACKNOWLEDGEMENTS

The authors would like to thank Rajat Monga for his invaluable help.

7. CONCLUSION

This paper presents a novel approach to training DNNs for hybrid systems which compares advantageously in terms of decoding complexity at equivalent accuracy to the standard approach. The method uses shared hidden layers across multiple output frames, making it possible to run the inference at a lower frame rate than the decoder while maintaining the same performance.

8. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Interspeech*, 2011, pp. 437–440.

[3] V. Vanhoucke, A. Senior, and M.Z. Mao, "Improving the speed of neural networks on CPUs," in *Deep Learning and Unsupervised Feature Learning Workshop, NIPS*, 2011.

[4] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large scale distributed deep networks," in *NIPS*, 2012.

[5] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, "Google Search by Voice: A case study," *Visions of Speech: Exploring New Voice Apps in Mobile Environments, Call Centers and Clinics*, vol. 2, pp. 2–1, 2010.

[6] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2010.

[7] KM Ponting and SM Peeling, "The use of variable frame rate analysis in speech recognition," *Computer Speech & Language*, vol. 5, no. 2, pp. 169–179, 1991.

[8] Q. Zhu and A. Alwan, "On the use of variable frame rate analysis in speech recognition," in *ICASSP*, 2000.