INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Phone duration modeling for LVCSR using neural networks

Hossein Hadian[1], *Daniel Povey*[2,3], *Hossein Sameti*[1], *Sanjeev Khudanpur*[2,3]

[1]Department of Computer Engineering, Sharif University of Technology, Iran
[2]Center for Language and Speech Processing, Johns Hopkins University, USA
[3]Human Language Technology Center of Excellence, Johns Hopkins University, USA

hadian@ce.sharif.edu, dpovey@gmail.com, sameti@sharif.edu, khudanpur@jhu.edu

Abstract

We describe our work on incorporating probabilities of phone durations, learned by a neural net, into an ASR system. Phone durations are incorporated via lattice rescoring. The input features are derived from the phone identities of a context window of phones, plus the durations of preceding phones within that window. Unlike some previous work, our network outputs the probability of different durations (in frames) directly, up to a fixed limit. We evaluate this method on several large vocabulary tasks, and while we consistently see improvements in Word Error Rates, the improvements are smaller when the lattices are generated with neural net based acoustic models.

Index Terms: automatic speech recognition, neural networks, phone duration models, reproducible results

1. Introduction

Most speech recognition systems do not explicitly model the duration of the phones or words. However, empirical results from past studies show that explicit duration modeling of speech sounds improves recognition results [1] [2] [3]. In fact, most state-of-the-art speech recognition systems are based on HMMs which implicitly model the duration of each state using the transition probabilities, which in turn leads to a geometric probability distribution function [1], whereas the true distribution of speech sounds is closer to gamma or log-normal [3]. Duration modeling can be either done by directly assuming a state duration density for HMMs (for e.g. [1]) or by learning a separate duration model and rescoring the recognition lattice (or N-best list) with duration scores [4] [5] [2]. The first approach leads to a significant increase in computational complexity of HMM learning and decoding algorithms and therefore is not very efficient for ASR [1] [6]. Different such methods are described and tested on a small 9-hour task in [3], where the authors reported WER improvements but with a significant decoding slow-down. Nevertheless it is common in speech synthesis [7], where it is used to generate phones with natural duration.

In this paper we apply phone duration probabilities via lattice rescoring. Our phone duration model is a neural network which predicts the phone durations (in frames) and is trained using the cross entropy objective function. This work is inspired by [2]; the main difference from that previous work is that while they assume a log-normal output distribution, we make no parametric assumptions (at least, for durations below a specified maximum). WERs are improved by 1% to 3% relative versus that previous approach.

We conduct experiments and improve WER on 5 databases including, in total, 8 baseline ASR models, including HMM-GMM models, hybrid HMM-DNN models, and the state-of-the-art LSTM-based LF-MMI models [8].

One feature of our system that deserves mentioning is a score normalization technique, which we use to counteract a bias towards longer phones. It consists of subtracting, for each base phone, the average log of the duration probabilities for that phone. This gives a further improvement of 1% to 3% relative. Overall the results are better than the baseline ASR result by 1% to 8% relative; but disappointingly, the relative improvement is the smallest for the best baseline models (those based on LF-MMI).

In the following section, we describe our approach. In Section 3, the experimental setup and results are shown; and we conclude in Section 4.

2. Discrete Phone Duration Modeling

We use a neural network to model the duration of phones. The basic framework is to model $p(d)$ for each value of d, and to include $\log p(d)$ as one of the components of the score in the final lattice.

2.1. Network output

To give the network as much freedom as possible to model any distribution, we use a softmax layer at the end of network and each discrete distribution $d = 1, 2, 3, \dots$ is modeled as a separate output class. Since the number of output neurons must be finite, we must limit the number of phone duration values that we model by choosing an integer constant D (e.g. D = 50), and in the training stage, we map any duration larger than D to D.

In test time we allocate the probability mass for the final class appropriately to each individual $d \geq D$. If y_D is the network output for the D'th class, representing the probability for all $d \geq D$, then we let $p(d)$ for any $d \geq D$ be $(1 - \alpha)\alpha^{(d-D)} y_D$, for a suitably chosen $0 < \alpha < 1$ that defines a geometric distribution. For our experiments here we chose $\alpha = \exp(-1/D)$; this parameter is not very critical as few phone durations are that long.

2.2. Network input

We aim for the model to predict the sequence of durations from the sequence of phones. This implies that we can use both left and right context for the phone identity, but must choose ei-

This work was partially supported by NSF CRI Award no. 1513128 and DARPA LORELEI Contract No HR0011-15-2-0024. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

http://dx.doi.org/10.21437/Interspeech.2017-1680

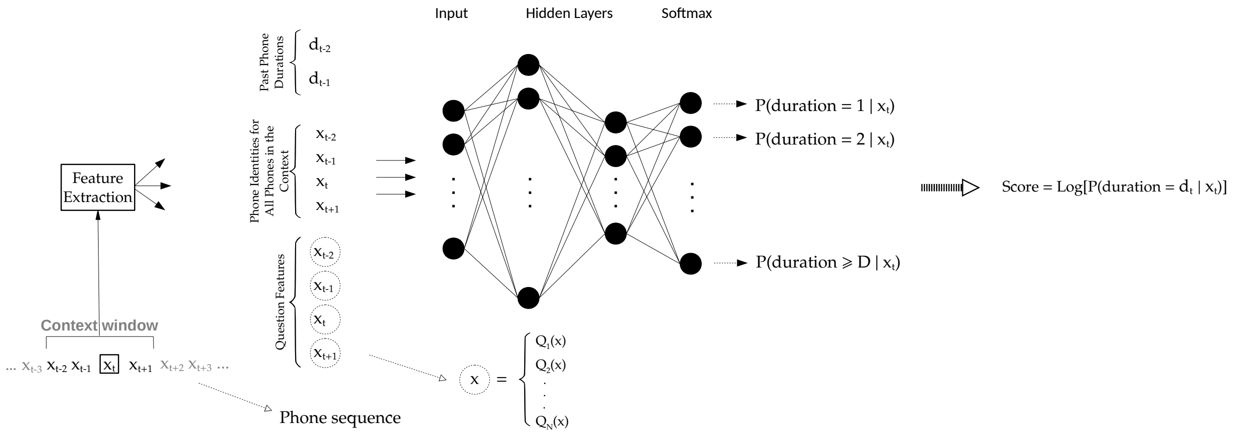

Figure 1: *The overview of the neural network in our approach, along with inputs and outputs. A context size of $L = 2$ and $R = 1$ is assumed.*

ther left or right context for the phone durations (otherwise the predictions for the sequence would depend on each other circularly). We choose left context for the phone durations.

Choose left and right context widths $L \geq 0$ and $R \geq 0$, e.g. $L = R = 3$, and bear in mind that right context will only be for phone identities. The features are as follows:

- For each context offset $-L \leq i \leq R$, the phone identity at that position, as a one-hot encoding (1 for the correct phone, zero for others). Total dimension is (number of phones) $\times(L + R + 1)$. We omit the word-position-dependent tags on phones, and any word stress information, for purposes of determining phone identity at this stage, so the dimension is the number of "real" phones, i.e. about 40 or so. We use an extra phone identity for unavailable context (i.e. at the edges).
- The next features depend on the phone sets that are used in the questions for the phonetic-context decision tree (in Kaldi, the questions.int file). These sets are automatically generated based on clustering the phones acoustically, but then we add in predetermined questions about vowel stress (in the WSJ system only) and word boundary information. For each context offset $-L \leq i \leq R$ and for each phone-set, 1 if the phone at position i is in the set, and zero otherwise. Total dimension is: (number of questions) $\times(L + R + 1)$.
- For each negative context offset $-L \leq i < 0$, the duration of the phone at this position. Total dimension is: L. Similar to [2], we normalize the duration values (which are in frames) using a sigmoid-like function to bound them to the finite range $(0, 1)$:

  $$d' = \frac{2}{1 + e^{-0.01d}} - 1,$$

  where d is the duration in frames, and d' is the normalized duration. For unavailable context (i.e. at the edges), we use duration zero.

The input and output information above is obtained from alignments of the training data, that is generated using the same model we intend to decode with.

2.3. Lattice rescoring

Phone duration log-probabilities are computed for the test-set lattices, scaled by a constant that is tuned on dev data, and added in with the acoustic and language model scores. To be able to compute these scores we need to be able to identify sufficient left and right context. We first modify the lattices so that the arcs correspond to phones. To simplify the task of expanding the lattices to provide sufficient context, we need to ensure that each phone in the lattice has a unique left context of $L + R$ phones, and add to the score of each phone in the lattice, the score for the phone that occurred R phones in the past, if there was one; then at lattice final states, we add in the score for the last few phones. This can be done by composing with a special FST that "remembers" the previous $L + R$ phones, with states that correspond to sequences of phones; this FST is constructed on-demand so that it does not consume much memory.

Figure 1 shows the overall approach of phone duration modeling using neural networks in this paper.

To increase generalization power of the network, we make the last hidden layer very small (10 neurons) so that the whole output distribution is learned with very few degrees of freedom. This is shown empirically in section 3. We use an additional technique to better model the phone duration values, which is explained in the following subsection.

2.4. Score normalization with priors

We found (see results) that it is helpful, in the lattice rescoring stage, to subtract the expected score for each phone from its score. By "score", we mean the log-probability of the duration. This helps to counteract out a bias towards paths with fewer phones. To be even more specific: for each phone p we compute the average, over all the training examples where p was the central phone, of the log-probability that the model assigned to the duration of that particular training example. So we store P values, where P is the number of phones; programmatically, it's very similar to the process of dividing by the class prior in hybrid DNN systems.

Table 1: *The two databases that we used for initial experiments, with their baseline WER (i.e. before rescoring).*

Database (ASR model)	Baseline WER
SWBD (HMM-GMM)	26.7
WSJ (TDNN HMM-DNN+CE)	6.77

Table 2: *Results of experimenting with maximum duration value D. Each value shows the WER after rescoring using a phone duration model with the specified D, with context $(3, 3)$, and without score normalization.*

Database	D						
	15	30	50	70	100	150	200
SWBD	26.1	26.0					
WSJ	6.60	6.46			6.55		

3. Experiments

As mentioned before, we used Kaldi [9] to run our experiments[1]. In all the experiments, unless otherwise stated, we assume D is 50, context window is $(L, R) = (3, 3)$ and the network has two hidden layers with ReLU activations. Assuming we have Q decision tree questions and P phones, the feature vector will have a dimension of $I = (L+R+1)*(P+Q)+L$. We choose the first hidden layer size to be $3I$ (i.e. 3 times the input dimension) and the second hidden layer size to be 10. These values worked best in most cases.

The databases we used for evaluation are 300-hour Switchboard (with the entire Hub5 '00 set as the evaluation set; also called eval2000) [10], AMI [11], TED-LIUM [12], Wall Street Journal (WSJ) [13], and Farsdat [14]. Farsdat is a Persian ASR database which consists of 27 hours of recorded speech from 100 speakers. We list two ASR models which we used for our initial experiments (to determine D, network size, etc.) along with their baseline word error rates (i.e. WER before rescoring using phone duration model) in Table 1.

3.1. Max duration

First we present the results of our experiments on D. Table 2 shows the results of our experiments with different values for D on Switchboard and Wall Street Journal (WSJ). It can be seen that the model is almost independent of the value of maximum duration D, but works best in the range 40 to 70. We set it to 50 for the rest of our experiments.

3.2. Context size

The most important factor in our experiments was found to be the context size. We investigated both the total context size (i.e. $L + R$) and the importance of left vs. right context. Table 3 compares different symmetric context sizes. We can see that the performance is improved as the context becomes larger up to $(3, 3)$. Using the context size $(4, 4)$ degrades the performance.

Table 4 shows the results of comparing effect of left and right context sizes when the total context size is fixed to 6. It seems that more symmetric context is better, although the results are not very conclusive.

Table 3: *Comparison of symmetric context sizes in phone duration modeling.*

Database	Context size (L, R)				
	(0, 0)	(1, 1)	(2, 2)	(3, 3)	(4, 4)
SWBD	26.7	26.3	26.1	26.0	26.1
WSJ	6.73	6.60	6.56	6.46	6.59

Table 4: *Effect of left context versus right context when total context size is $L + R = 6$.*

Database	Context size (L, R)				
	(5, 1)	(4, 2)	(3, 3)	(2, 4)	(1, 5)
SWBD	26.0	26.0	26.0	25.9	26.1
WSJ	6.61	6.50	6.46	6.49	6.50

3.3. Network size

We tried different number of hidden layers and also experimented the effect of a final bottleneck hidden layer. The results are presented in Table 5. It can be seen that the model with two hidden layers has performed better than the model with one or three hidden layers. Although the differences are not significant in the case of WSJ. Besides, final bottleneck hidden layer has helped consistently.

3.4. Score normalization

As explained in Section 2.4, we applied a score normalization to decrease word deletion rate. This normalization was very effective and improved the results by 0.2% to 0.4% in almost all cases. The results which show the effect of score normalization are presented in Table 6.

3.5. Performance on various ASR models

Finally we present the results using the best setup for all the ASR models. This means using $D = 50$, a network with 2 hidden layers where the second one is bottleneck, and with score normalization as explained before. Table 7 shows the WERs after rescoring with a phone duration model with the setup explained. The baseline WER (i.e. before rescoring) and the WER after rescoring with log-normal objective function are also included. Furthermore, the results of score normalization applied to log-normal objective function are displayed for comparison. We can see that score normalization is more effective on our method versus log-normal objective function.

It can also be seen that the improvement (due to duration modeling) is consistently lower when more powerful DNNs are used for acoustic modeling. For example, the LF-MMI method has been improved only 1% relatively. This might suggest that DNN-based acoustic models - especially those with sequence-level objective functions - implicitly model phone durations better.

Table 5: *Comparison of different network sizes and effect of a final bottleneck layer. "1H" means 1 hidden layer and so on. "2H+bottleneck" means there are two hidden layer and the second one is bottleneck with 10 neurons.*

Database	Network setup				
	1H	2H	3H	2H+bottleneck	3H+bottleneck
SWBD	26.5	26.1	26.2	26.0	26.1
WSJ	6.56	6.52	6.50	6.46	6.49

[1] The code is available in Kaldi's github page, and the results are reproducible

Table 6: *Effect of score normalization with priors. The last two columns show the WER after rescoring with duration model, without score normalization and with score normalization respectively.*

Database	ASR Model	Baseline	w/o snorm	with snorm
SWBD	HMM-GMM	26.7	26.0	25.6
	TDNN HMM-DNN+CE	17.9	17.7	17.4
	BLSTM LF-MMI [8]	15.9	15.9	15.7
WSJ	HMM-GMM	9.69	9.09	8.92
	TDNN HMM-DNN+CE	6.77	6.46	6.24
TED-LIUM	TDNN HMM-DNN+CE	11.9	11.4	11.1
AMI	BLSTM HMM-DNN+CE	40.08	39.78	39.40
Farsdat	HMM-GMM	8.27	7.85	7.74

Table 7: *WER improvements on all evaluated models using the best setup. The numbers are word error rates on the evaluation set of the corresponding databases. The "logn" column shows the results of log-normal objective function (i.e. previous work :[2]), and "logn + snorm" shows the results of log-normal objective function with our score normalization technique applied.*

Database	ASR Model	Baseline	Our approach	logn	logn + snorm
SWBD	HMM-GMM	26.7	25.6	26.2	25.9
	TDNN HMM-DNN+CE	17.9	17.4	17.7	17.5
	BLSTM LF-MMI [8]	15.9	15.7	15.9	15.9
WSJ	HMM-GMM	9.69	8.92	9.07	8.95
	TDNN HMM-DNN+CE	6.77	6.24	6.54	6.30
TED-LIUM	TDNN HMM-DNN+CE	11.9	11.1	11.3	11.1
AMI	BLSTM HMM-DNN+CE	40.08	39.40	39.64	39.54
Farsdat	HMM-GMM	8.27	7.74	7.92	7.89

3.6. Comparison of predictive duration distributions

We have plotted the predictive distributions for a few test examples using the log-normal and cross-entropy models in Figure 2. In most cases the distributions are similar as in 2a. However in many cases, the cross-entropy model predicts more peaky distributions around the true duration as in 2b. Besides, there are a few cases where the cross-entropy model is obviously superior in modeling: Figures 2c and 2d show two such cases where the input phone seems to have a multi-modal distribution of duration. It can be seen in these figures that the cross-entropy model has predicted a multi-modal duration distribution which can give good scores to the true duration value.
Briefly put, we can see from these figures that (1) our non-parametric model is capable of learning the distributions smoothly and can generalize, and (2) there are examples of phones (with specific contexts) where the duration distribution is multi-modal and our non-parametric approach handles them better than a parametric unimodal model.

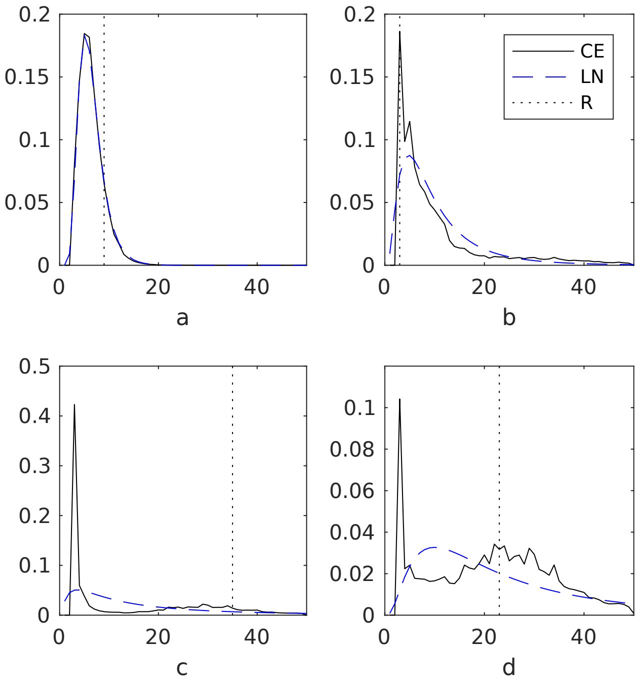

Figure 2: *Probability distributions predicted by our model (CE), and log-normal model (LN) for 4 test examples. The horizontal axis shows the duration in frames. The dotted line shows the reference (i.e. true) duration for that phone.*

4. Conclusion

In this research, we investigated the effect of explicit phone duration modeling using neural networks on performance of ASR models. Unlike some previous work which assumed a log-normal distribution over phone durations, we did not assume any prior distribution and modeled phone durations discretely with a softmax layer. We evaluated our approach on various speech databases and ASR models, to make sure the improvements are not noise. 1 to 8 percent relative improvement was achieved in all cases.

5. References

[1] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] T. Alumäe, "Neural network phone duration model for speech recognition." in *INTERSPEECH*, 2014, pp. 1204–1208.

[3] J. Pylkkönen and M. Kurimo, "Duration modeling techniques for continuous speech recognition." in *INTERSPEECH*, 2004.

[4] A. Anastasakos, R. Schwartz, and H. Shu, "Duration modeling in large vocabulary speech recognition," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1. IEEE, 1995, pp. 628–631.

[5] V. R. Gadde, "Modeling word duration for better speech recognition," in *Proceedings of NIST Speech Transcription Workshop*, 2000.

[6] M. Russell and R. Moore, "Explicit modelling of state occupancy in hidden markov models for automatic speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, vol. 10. IEEE, 1985, pp. 5–8.

[7] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for hmm-based speech synthesis." in *ICSLP*, vol. 98, 1998, pp. 29–32.

[8] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Interspeech*, 2016.

[9] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[10] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ser. ICASSP'92. Washington, DC, USA: IEEE Computer Society, 1992, pp. 517–520. [Online]. Available: http://dl.acm.org/citation.cfm?id=1895550.1895693

[11] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The ami meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.

[12] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the tedlium corpus with selected data for language modeling and more ted talks." in *LREC*, 2014, pp. 3935–3939.

[13] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 357–362. [Online]. Available: http://dx.doi.org/10.3115/1075527.1075614

[14] J. Sheikhzadegan and M. Bijankhan, "Persian speech databases," in *2nd Workshop on Persian Language and Computer*, 2006, pp. 247–261.