FLAT START TRAINING OF CD-CTC-SMBR LSTM RNN ACOUSTIC MODELS

Kanishka Rao, Andrew Senior, Haşim Sak

Google

{kanishkarao,andrewsenior,hasim}@google.com

ABSTRACT

We present a recipe for training acoustic models with context dependent (CD) phones from scratch using recurrent neural networks (RNNs). First, we use the connectionist temporal classification (CTC) technique to train a model with context independent (CI) phones directly from the written-domain word transcripts by aligning with all possible phonetic verbalizations. Then, we devise a mechanism to generate a set of CD phones using the CTC CI phone model alignments and train a CD phone model to improve the accuracy. This end-to-end training recipe does not require any previously trained GMM-HMM or DNN model for CD phone generation or alignment, and thus drastically reduces the overall model building time. We show that using this procedure does not degrade the performance of models and allows us to improve models more quickly by updates to pronunciations or training data.

Index Terms— Flat start, CTC, LSTM RNN, acoustic modeling.

1. INTRODUCTION

Most modern large scale vocabulary speech recognition systems employ neural network acoustic models which are commonly feed-forward deep neural networks (DNNs) or deep recurrent neural network (RNNs) such as Long Short Term Memory (LSTM) [1, 2]. These 'hybrid' models assume a Hidden Markov Model (HMM) for which the neural network predicts HMM state posteriors [3]. A recent variation of LSTM-HMM, CLDNN [4], uses convolutional layer in addition to LSTM layers and has proven to perform better than LSTM RNNs. However, all these acoustic models trained with cross entropy (CE) loss require an alignment between acoustic frames and phonetic labels, which could be obtained from a Gaussian mixture model (GMM) [5, 6] or a neural network (initially aligned with a GMM-HMM). The bootstrapping model is used in two ways; for generating alignments and for building context dependency tree. GMM-HMM can be 'flat started' from the phonetic transcriptions [7], and the phone alignments from the initial GMM-HMM can be used to build context dependent phone models for improving accuracy.

The conventional neural network acoustic models require training a GMM-HMM and sometimes even an initial neural network to get better alignments. These iterations can take a long time, often a few weeks. A lengthy acoustic model training procedure not only delays the deployment of improved models but also hinders timely refresh of acoustic models. Being able to flat start an LSTM RNN is desirable since it eliminates the need for a GMM, simplifying and shortening the training procedure. A GMM-free training approach for DNN-HMM is described in [8] where DNNs are flat started and their alignments are used for building CD state-tying trees. In this paper, we describe a flat start procedure for LSTM RNNs trained with the CTC objective function.

The CTC technique has been shown to be very successful at phoneme recognition on the TIMIT dataset using deep bidirectional LSTM RNNs [9]. Unidirectional CTC based acoustic models have also been shown to outperform the state-of-the-art in large vocabulary speech recognition [10]. CTC models have the advantage of not needing alignment information as they can be trained directly with phonetic transcription. However, phonetic transcription of words cannot be obtained readily from the text transcription since there might be multiple verbalizations of the same word, e.g. '10' → 'ten' or 'one oh', and further each verbal word may have multiple valid pronunciations. Thus, a text transcription may have many valid phonetic transcriptions. The true spoken phoneme labels can be obtained by aligning the audio and the alternative pronunciations with an existing acoustic model, however, this relies on training a GMM-HMM or DNN-HMM which results in the same lengthy training procedure as with the conventional neural network models.

In this paper, we show that we can train RNN phone acoustic models using the CTC technique directly from transcribed audio and text data without requiring any fixed phone targets generated from a previous model. We also outline a mechanism to build a CD phone inventory using a CTC based phone acoustic model. Using these techniques we can flat start training a CTC phone model which is used to build a CD phone inventory, and finally, we can train a CTC CD phone model and show that it outperforms our previous best CLDNN models for various languages. We also show how this procedure can be useful to quickly refresh acoustic models whenever other components of the speech system (such as the pronunciations) are updated.

In section 2, we describe the CTC algorithm and how we adapt it for flat start. In section 3, we outline the end-to-end flat start CTC procedure for training acoustic models from scratch including generating the CD phones from a CTC CI model. Section 4 details our experimental setup with the results in section 5. Finally, in section 6 we discuss the results of the flat start CTC training approach.

2. CONNECTIONIST TEMPORAL CLASSIFICATION

The connectionist temporal classification (CTC) approach is a learning technique for sequence labeling using RNNs [9]. It can learn an alignment between the input and target label sequences. Different from conventional alignment learning, the CTC introduces an additional *blank* output label which the model can choose to predict for relaxing the decision of labeling each input. It is ideal for acoustic modeling since labeling each acoustic frame phonetically is not required for speech decoding. A CTC based acoustic model may *listen* to several acoustic frames before outputting a non-*blank* label (phonetic unit in this case). A more detailed discussion of how CTC may

978-1-4799-9988-0/16/$31.00 ©2016 IEEE

be used for acoustic modeling is given in [11, 10].

The CTC loss function tries to optimize the total likelihood of all possible labelings of an input sequence with a target sequence. It calculates the sum of all possible path probabilities over the alignment graph of given sequences using the forward backward algorithm. The alignment graph allows label repetitions possibly interleaved with *blank* labels. When applied to acoustic modeling in this sequence labeling framework, this approach requires phonetic transcriptions of utterances which necessitates a previously trained acoustic model. Any significant change in training (such as updates to training data or word pronunciations) would require re-training all the acoustic models starting from the initial model used to obtain the phonetic transcriptions.

2.1. CTC for Flat Start

The CTC technique can be easily extended to align an input sequence with a graph representing all possible alternative target label sequences. This is useful, for instance, in flat start training of acoustic models where we have word level transcripts in written domain for training utterances but we do not know actual verbal forms of the words and phonetic transcriptions of an utterance. Note that there can be more than one possible verbal expansions of words and similarly phonetic pronunciations of words. We extend the CTC approach to learn a probabilistic alignment over all possible phonetic sequence representations corresponding to all the verbal forms of a written text transcript.

The conventional CTC technique can be implemented using finite state transducer (FST) framework by building an FST representation, P, for a given target phone sequence and another auxiliary transducer, C, allowing for optional *blank* label insertions and actual label repetitions. Then, the composed transducer $C \circ P$ represents a graph which can be used to align the input (see [11] for more details). We alter this prescription for flat start training of CTC models by using $C \circ L \circ V \circ T$, where T is the FST representation for the given target word level transcript, V is a verbalization FST [12], L is a pronunciation lexicon FST. Given the pronunciation and verbalization models as L and V we can train acoustic models directly from the acoustic data and corresponding word transcripts in the written form using the forward backward algorithm to align the input with this composed FST representation.

To ensure that flat start training of CTC acoustic models does not degrade the accuracy, we compared the performance of a CTC model trained with phonetic alignments generated by a DNN model and found it to be exactly the same as the CTC model trained with the flat start technique. The major advantage of flat start is that it does not require any previous model which is more convenient and reduces the overall training time.

3. CTC FLAT START TRAINING PROCEDURE

In this section we outline an end-to-end procedure to quickly train and refresh acoustic models using the flat start training of CTC models in the following steps:

1. A bidirectional LSTM RNN model, BLSTM-CTC-CI, is trained with the flat start CTC technique to predict phonemes. This model is used as an intermediate model since our objective is to train unidirectional models for real-time streaming speech recognition.
2. This BLSTM-CTC-CI is used to align the acoustic model training data to obtain the phonetic alignments and the statistics associated with the phone spikes are used to create context-dependent phones.
3. A unidirectional LSTM RNN, CD-CTC-sMBR, is trained with the flat start CTC technique to predict these context-dependent phones. This is the final model used for speech recognition.

3.1. Training BLSTM-CTC-CI Models

Speech recognition systems typically predict context-dependent labels such as triphones since the added context restricts the decoding search space and results in better word error rates. However, in order to build a CD phone inventory we first train a CI phone acoustic model.

We train a bidirectional LSTM (BLSTM) using flat start CTC to predict context-independent (CI) phone labels. As mentioned earlier, this step only requires a pronunciation model, a verbalization model and transcribed acoustic model training data. The performance of this BLSTM-CTC-CI model is measured by its phoneme error rate.

This bidirectional model is used only to generate statistics about context-dependent phone labels which can be used to establish a CD phone inventory. We train this CI model as a bidirectional network since they perform better than unidirectional models, are faster to train, and the alignments better matches the actual timing of acoustic frames. We cannot use this bidirectional model for speech recognition since we require unidirectional models for streaming recognition results for latency reasons.

3.2. Building CD Phones

Once the BLSTM-CTC-CI has reached a reasonable phoneme error rate (which typically takes less than 1 day), we re-align the data to generate the CD phones. Previously, it was shown that it is possible to build context dependent whole-phone models, and that for LSTM-HMM hybrid speech recognition, these models can give similar results to context dependent HMM state models, provided that a minimum duration is enforced [13]. We repeat that procedure, using the hierarchical binary divisive clustering algorithm [7] for context-tying. Using the trained BLSTM-CTC-CI, we do a Viterbi forced alignment to get a set of frames with phone labels (and many frames with blank labels), and find sufficient statistics for all the frames with a given phone label and context. The sufficient statistics are the mean and diagonal covariance of input log-mel filterbanks features for labelled frames. If two or more frames are aligned for a phoneme we only use the initial frame to generate statistics, variations of this approach were tried (such as using all frames) but these did not affect the performance of the system. One tree per phone is constructed, with the maximum-likelihood-gain phonetic question being used to split the data at each node. The forest consisting of all phone-trees is pruned to a fixed number of CD phones by merging the two CD phones with the minimum gain. We find that beyond a certain number (500 for Russian) having more CD phones does not improve the accuracy (see Table 1) and thus pick the smallest CD phone inventory with the best performance.

Although the CTC does not guarantee the alignment of phone spikes with the corresponding acoustic input frames, we find the alignments of a bidirectional model to be generally accurate. However, this is not true for the unidirectional phone models that we trained which generally choose to delay its phone predictions (typically around 300 ms). Figure 1 shows such an alignment for an example utterance. The CTC phone spikes are close to the phone

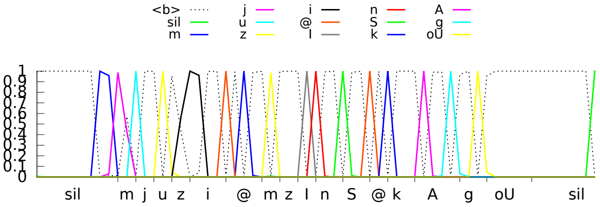

Fig. 1: The timing of CI phone spikes from a BLSTM-CTC-CI model for an example utterance with the transcript 'museums in Chicago'. The x-axis shows the phonetic alignments as obtained with a DNN model and y-axis shows the phone posteriors as predicted by the CTC model. The CTC phone spikes are found to be close to the time intervals of DNN phone alignments.

time intervals obtained by aligning with a DNN model that uses a large context window (250 ms).

Number of CD Phones	WER (%)
50	17.0
100	16.9
500	16.7
1000	16.7
2000	16.7
5000	16.7

Table 1: The WERs for CTC models trained with various numbers of CD phones for Russian (without sequence discriminative training).

3.3. Training CD-CTC-sMBR Models

Using the generated CD phone inventory, we train a unidirectional CTC model predicting these CD phones. We build a context dependency transducer, D, from the CD phone inventory, that maps CD phones to CI phones. Then, we can repeat the flat start technique with CTC for CD phones using the composed transducer graph $C \circ D \circ L \circ V \circ T$. After the CTC model training converges fully, we further improve it by training with the sMBR sequence discriminative criterion as described in [2, 11, 10].

One may choose to train a bidirectional CD phone model, however, such a model does not allow for streaming recognition results. In this paper, we do not consider the bidirectional models for speech recognition and only compare the WERs of unidirectional models.

4. EXPERIMENTAL SETUP

All the LSTM networks are trained on a 3 million utterance dataset consisting of anonymized and hand-transcribed audio utterances. To ensure our approach is language-independent, we repeat our experiment with Hindi, Russian and Brazilian Portuguese. We compute acoustic features as the 80-dimensional log mel filterbank energy every 10ms, eight such features are stacked resulting in a 640-dimensional input feature vector for the CTC models. We skip two in every three such vectors. This results in a single input feature vector every 30ms. This mechanism of frame stacking and skipping has been optimized for CTC acoustic models and is identical to the setup in [10].

We clip the activations of memory cells to range [-50, 50], and their gradients to [-1, 1] This makes training with CTC models stable. For the BLSTM-CTC-CI model we use a deep LSTM RNN with 5 layers of forward and backward layers of 300 memory cells, the CD-CTC-sMBR LSTM is a 5-layer deep RNN with forward layers of 600 memory cells. CTC training for all models is done with a learning rate of 0.0001 with an exponential decay of one order of magnitude over the length of training.

We ensure robustness to background noise and reverberant environments by synthetically distorting each training example in a room simulator with a virtual noise source. Noise is taken from the audio of YouTube videos. Each training example is randomly distorted to get 20 variations. This 'multi-condition training' also prevents overfitting of CTC models to training data. To estimate the performance of acoustic models we create *Noisy* versions of our test sets similarly.

The final trained models are evaluated in a large vocabulary speech recognition system on a test set of roughly twenty thousand hand-transcribed, anonymized utterances. For all the decoding experiments, we use a wide beam to avoid search errors. After a first pass of decoding using the CTC models with a 5-gram language model heavily pruned, lattices are rescored using a large 5-gram language model.

All models are evaluated based on their word error rate (WER) on the clean and noisy test sets.

5. RESULTS

We compare the models obtained by the flat start CTC training procedure to our state-of-the-art CLDNN models. The training and evaluation datasets for both systems are identical.

5.1. Word Error Rate on Test Sets

We compare the performance of the CD-CTC-sMBR models obtained using the flat start CTC to CLDNN-sMBR models. The flat start CTC models generally outperform the CDLNN in terms of WER for the languages we tested; Russian, Hindi and Brazilian Portuguese. The improvements in WER are similar for clean and noisy test sets. For Brazilian Portuguese we found a CD phone inventory of size 2000 performed best, while Hindi and Russian only required 500 CD phones. Table 2 reports the final WER after sequence discriminative training.

	CLDNN-sMBR		CD-CTC-sMBR LSTM	
TestSet	Clean	Noisy	Clean	Noisy
Hindi	27.4	35.6	26.7	33.9
Russian	14.7	25.4	14.7	24.4
Brazilian Portuguese	11.8	20.4	11.7	19.5

Table 2: WER for the CLDNN-sMBR versus the CD-CTC-sMBR on clean and noisy test sets for Russian, Hindi and Brazilian Portuguese.

5.2. Impact on Real Traffic

To measure the impact of the flat start CTC models beyond the offline test sets, we recognize utterances from real traffic using the baseline model (CLDNN-sMBR) and the CD-CTC-sMBR model. From these we randomly sample 1000 utterances with different recognition results from these two systems and ask human raters to label each result as either *Nonsense*, *Unusable*, *Usable* or *Exact*. Figure 2 shows the distributions of these ratings for both systems for Hindi traffic. The CD-CTC-sMBR model rated higher with more

Exact and *Usable* recognitions and fewer *Nonsense* and *Unusable* recognitions compared to the CLDNN-sMBR.

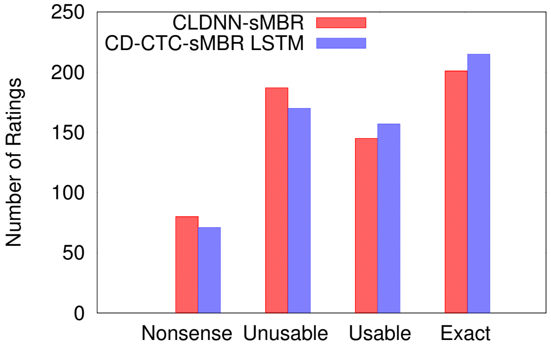

Fig. 2: Human ratings for randomly sampled Hindi utterances recognized by the CD-CTC-sMBR versus the CLDNN-sMBR.

6. DISCUSSION

6.1. Learning Multiple Pronunciations

With flat start CTC we no longer provide fixed phoneme targets during CTC training, instead, all possible valid pronunciations for the given transcript are available to the network. The network can *decide* which of the valid pronunciations to predict for a given training example. To confirm that the network is indeed able to utilize these multiple valid pronunciations we count the usage of pronunciation variants for a few example words, see Table 3.

Word	*either*		*gyro*	
Frequency	128		77	
Pronunciations	*i D @'*	*aI D @'*	*dZ aI r oU*	*j i r oU*
Frequency	111	17	66	11

Table 3: The frequency of example words with multiple valid pronunciations in the training data and the frequency of the pronunciation outputted by the flat start CTC model.

6.2. Acoustic Model Refresh

Three major components comprise a speech recognition system; acoustic, pronunciation and language models. Although during inference all these models are used together they are generally trained and improved independently, and often an improvement in one system may necessitate refreshing the others. A common example of this when new word pronunciations are added, the acoustic models may need to be refreshed to take advantage of them. In this section, we examine one such scenario for Hindi.

We find a WER regression (27.4 → 28.2) when we add 40,000 new human-transcribed Hindi pronunciations to our system, this can happen if there is a mis-match between the pronunciations used during evaluation and those used during acoustic model training. The pronunciations used during acoustic model training may contain incorrect pronunciations (possible if they are generated using an automated tool), then the acoustic model will learn to predict these incorrect phonetic transcriptions. If, at a later time, these incorrect pronunciations are corrected then a WER regression may result from a mis-match between the pronunciation and acoustic models. A refresh of the acoustic model is required to take advantage of the 40,000 new Hindi pronunciations. We can re-train the CLDNN-sMBR with the new pronunciations, however, this would first require us to re-train the GMM since the alignments would be different with the new pronunciations. Instead, we use the flat start CTC procedure to quickly update our acoustic models, which does not require a GMM. We flat start two CTC CD phone models using the baseline and new pronunciations. The CTC CD model with the baseline pronunciations improves on the CLDNN-sMBR trained with the same pronunciations, see Table 4, this improvement is due to the CTC-CD-sMBR versus CLDNN-sMBR technology (as already discussed in section 5). However, the CTC CD model flat started with the new pronunciations further improves the recognition (26.7 → 26.4) showing that these new pronunciations are indeed beneficial and required an acoustic model refresh. We expect to see similar improvements if we refreshed the CLDNN-sMBR model, however, here we show how flat start CTC makes the refresh simpler and faster, without the need for re-training a GMM.

Acoustic Model	Pronunciation Model	WER
CLDNN-sMBR	Baseline	27.4
CLDNN-sMBR	Updated	28.2
CD-CTC-sMBR LSTM	Baseline	26.7
CD-CTC-sMBR LSTM	Updated	26.4

Table 4: The performance of CLDNN-sMBR and flat started CD-CTC-sMBR LSTM models for Hindi with a baseline pronunciation model and an updated one where 40,000 new pronunciations are added. It should be noted that the same CLDNN-sMBR is shown with the baseline and update pronunciations while a new CD-CTC-sMBR LSTM is flat started for each set of pronunciations.

7. CONCLUSION

We have extended the CTC training technique to allow training of phoneme models directly from written transcripts. We use this mechanism to train a bidirectional CTC phone model which is used only to generate a CD phone inventory. We then train a CD-CTC-sMBR LSTM RNN model using this CD phoneme inventory and show that they perform better than the current state-of-the-art CLDNN-sMBR models. We have shown that this approach is language independent with improvements for all languages tested; Russian, Hindi, Brazilian Portuguese. The end-to-end flat start CTC training procedure is faster than training a GMM-HMM model to bootstrap and train a neural network model. Using this flat start CTC procedure one can train and refresh state-of-the-art acoustic models from scratch in a relatively short time.

8. REFERENCES

[1] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH*. 2014, pp. 338–342, ISCA.

[2] Haşim Sak, Oriol Vinyals, Georg Heigold, Andrew Senior, Erik McDermott, Rajat Monga, and Mark Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *INTERSPEECH*, 2014, pp. 1209–1213.

[3] N. Morgan and H. Bourlard, "Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach," *IEEE Signal Processing Magazine*, vol. 12, no. 3, pp. 25–42, 1995.

[4] T. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[5] H. Hermansky, D.P.W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000.

[6] T.N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[7] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. ARPA Human Language Technology Workshop*, 1994.

[8] Andrew Senior, Georg Heigold, Michiel Bacchiani, and Haitao Liao, "Gmm-free dnn acoustic model training," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5602–5606.

[9] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[10] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *INTERSPEECH*. 2015, pp. 1468–1472, ISCA.

[11] Haşim Sak, Andrew Senior, Kanishka Rao, Ozan İrsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *ICASSP*, 2015, pp. 4280–4284.

[12] Hasim Sak, Françoise Beaufays, Kensuke Nakajima, and Cyril Allauzen, "Language model verbalization for automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8262–8266.

[13] Andrew Senior, Hasim Sak, and Izhak Shafran, "Context dependent phone models for lstm rnn acoustic modelling," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 4585–4589.