AN EMPIRICAL EXPLORATION OF CTC ACOUSTIC MODELS

Yajie Miao[1], *Mohammad Gowayyed*[1], *Florian Metze*[1], *Xingyu Na*[2], *Tom Ko*[3], *Alex Waibel*[1]

[1]Carnegie Mellon University, Pittsburgh, PA 15213
[2]Institute of Acoustics, Chinese Academy of Sciences, Beijing, China
[3]Huawei Noahs Ark Research Lab, Hong Kong, China

ABSTRACT

Recent work has successfully applied the connectionist temporal classification (CTC) loss function to automatic speech recognition (ASR). Applied upon deep recurrent neural networks (RNNs), CTC learns the alignments between speech frames and label sequences automatically, which removes the need for pre-generated frame-level labels. Although showing promising performance, CTC has received insufficient investigation in comparison to the existing hybrid approach. This paper presents an extensive exploration of CTC-based model training. On large-scale acoustic modeling tasks, we empirically study the optimal configuration and architectural variants for CTC. With large amounts of training data, CTC models are observed to outperform the state-of-the-art hybrid systems. Also, detailed experiments reveal that CTC can be readily ported to other languages other than English, and can be enhanced by employing improved feature front-ends.

Index Terms— CTC, LSTM, RNNs, acoustic modeling, speech recognition

1. INTRODUCTION

The introduction of deep neural networks (DNNs) and recurrent neural networks (RNNs) as acoustic models has achieved tremendous success for automatic speech recognition (ASR) [1, 2, 3, 4]. In the *hybrid* paradigm, DNNs/RNNs are used to classify speech frames into labels which are normally clustered context-dependent (CD) states. These labels are pre-generated by an initially trained Gaussian mixture model through forced alignment. Model training can then be carried out with the cross-entropy objective function which is likely to be followed by sequence training. Recently, an alternative loss function, connectionist temporal classification (CTC) [5], has been proposed for sequence labeling problems with variable-length inputs and outputs. With "blank" symbols inserted between labels, CTC constructs frame-level paths as intermediate representations to bridge frame-level network outputs with label sequences. When applied to acoustic modeling, CTC automatically learns the alignments between speech frames and labels. Thus, CTC removes the need for pre-generated frame-label labels and thereby the building of the initial GMMs. Used together with deep RNN models, CTC has been shown to achieve the state-of-the-art performance on various large-scale acoustic modeling tasks [6, 7, 8, 9].

Although performing promisingly, CTC has received substantially less investigation than the hybrid HMM/DNN approach. Most of the existing CTC work is constrained to particular tasks/scenarios. For instance, although showing promising results on English, the application of CTC on other languages has not been reported in literature. In this paper, we present an extensive empirical study to investigate how CTC training behaves under various conditions. Our exploration focuses on the following aspects:

- Optimal configuration. CTC commonly uses deep RNNs with the Long Short-Term Memory (LSTM) units as acoustic models. Motivated by past work on LSTMs [10], we initialize the bias vector of the LSTMs forget gates to larger values (e.g., 1.0 and 2.0). This initialization is observed to bring consistent gains for CTC training. Also, our experiments reveal how the amounts of training data affect the performance of CTC models.

- Architectural variants. We study two architectural variants of CTC models. First, a convolution layer is added following the input features and prior to the LSTM layers. The resulting ConvLSTM architecture achieves slight improvement over the vanilla LSTM. Second, we compare a uni-directional LSTM model with the bi-directional LSTM, and observe that the uni-directional model performs 10% worse than the bi-directional one.

- Language Expansion. Due to language diversity, it is intriguing to study how CTC works on various languages other than English. In this work, we port CTC to a task of transcribing Mandarin conversational telephone speech [11]. By directly modeling thousands of Mandarin characters, CTC achieves the state-of-the-art results on this task.

- Front-ends. Apart from the raw acoustic features (e.g., MFCCs, filterbanks), the HMM/GMM and HMM/DNN paradigms exploit advanced front-ends (e.g., fMLLRs, VTLNs). This paper empirically verifies the applicability of these front-ends in the context of CTC models.

2. REVIEW OF CTC

The connectionist temporal classification (CTC) approach [5] is a loss function for sequence labeling problems where the inputs and the label sequences have variable lengths. Instead of employing pre-generated frame-level labels, CTC automatically learns the alignments between speech frames and their label sequences (e.g., phonemes or characters). In previous work [6, 7, 9], the acoustic models used together with CTC are normally deep RNNs architectures using the LSTM units [12] (which we will consistently refer to as LSTMs). The nodes in the softmax layer of the LSTM model correspond to the original labels, as well as a special *blank* label which estimates the probability of emitting no label at a time step. CTC trains the LSTM model to maximize $\ln Pr(z|x)$, the log-likelihood of the label sequence z given the inputs x.

To bridge the frame-level LSTM outputs with the utterance-level label sequences, CTC introduces an intermediate representation called the *CTC path*. A CTC path is a sequence of labels at the frame level, allowing repetitions of the blank to be inserted between labels. The label sequence can be represented by a set of all the possible CTC paths that are mapped to it. The likelihood of z is then evaluated as an aggregation of the probabilities of its CTC paths:

$$Pr(z|x) = \sum_{p \in \Phi(z)} Pr(p|x) \tag{1}$$

where $\Phi(z)$ is the set of CTC paths corresponding to z. With this formulation, $Pr(z|x)$ can be evaluated using a forward-backward algorithm over a trellis that compactly encodes $\Phi(z)$. The likelihood of the label sequence **z** is then computed as:

$$Pr(z|x) = \sum_{u=1}^{|z|} \frac{\alpha_t^u \beta_t^u}{y_t^u} \tag{2}$$

where the variable α_t^u represents the total probability of all CTC paths that end with label l_u at frame t, and can be recursively computed from α_{t-1}^u and α_{t-1}^{u-1}. Similarly, the backward variable β_t^u carries the total probability of all CTC paths that starts with label l_u at t and reaches the final frame T. The quantity y_t^u represents the posterior of the label u outputted by the LSTM network.

The loss function becomes differentiable with respect to the LSTM outputs. The gradients of the loss function with respect to the outputs y_t^k of the LSTM softmax layer can be computed as

$$\frac{\partial \ln Pr(z|x)}{\partial y_t^k} = \frac{1}{Pr(z|x)} \frac{1}{(y_t^k)^2} \sum_{u \in \Upsilon(l,k)} \alpha_t^u \beta_t^u \tag{3}$$

where $\Upsilon(z,k) = \{u|z_u = k\}$ defines an operation on the label sequence that returns the elements of z which have the value k. These errors are back-propagated through the softmax layer and further into the LSTMs to update the model parameters.

When CTC is applied to acoustic modeling, incorporating lexicons and language models into decoding has been a challenge. Our previous work [9] proposes a generalized decoding method based on weighted finite-state transducers (WFSTs). In this method, individual components (CTC labels, lexicons and language models) are encoded into WFSTs, and then composed into a comprehensive search graph. The WFST representation provides a convenient way of handling the CTC *blank* label and enabling beam search during decoding.

3. OPTIMAL CONFIGURATION

We first explore the optimal configuration of the LSTM models for CTC training. Our experiments in this section are conducted on the Switchboard conversational telephone transcription task.

3.1. Experimental Setup

We use Switchboard-1 Release 2 (LDC97S62) as the training set which contains over 300 hours of speech. For fast turnarounds, we also select 110 hours from the training set and create a lighter setup. CTC training uses a deep bi-directional LSTM architecture as the acoustic model. On the 110-hour and 300-hour setups, the LSTM network consists of 4 and 5 bi-directional LSTM layers respectively. At each layer, both the forward and the backward sub-layers contain 320 memory cells. Inputs of the LSTM model are 40-dimensional filterbank features together with their first and second-order derivatives. The features are normalized via mean subtraction and variance normalization on the speaker basis. Initial values of all the model parameters are randomly drawn from a uniform distribution with the range [-0.1, 0.1]. Model training adopts an initial learning rate of 0.00004 which is decayed based on the variation of the accuracy of the hypothesis labels with respect to the reference label sequences. CTC training models context-independent (CI) phones. Totally we have 46 labels including phones, noise marks and the blank.

Our decoding follows the WFST-based approach presented in [9]. The label posteriors generated by the LSTM model are normalized with the label priors estimated from the training transcripts. A trigram language model (LM) is trained on the training transcripts. This LM is then interpolated with another trigram LM trained on the Fisher English Part 1 transcripts (LDC2004T19). We report results on the Switchboard part of the Hub500 (LDC2002S09) test set.

3.2. Results

Table 1 presents the results of the resulting CTC-trained acoustic models under various settings. A key configuration of LSTM models is the initialization of the forget gate bias vector. Most of the existing work has simply initialized the bias vector with small random weights. Although working well on many application, this initialization effectively decays the gradients back-propagated at each time step. This issue can be resolved simply by initializing the bias to a large value [10]. In our experiments, we set the initial values of the forget gates bias vector uniformly to 1.0. From Table 1, we can see that this initialization brings consistent gains over the initialization with small random values. The WER is improved by 3.9% and 3.8% respectively on the 110-Hour and 300-Hour setups.

In Table 1, we also compare the CTC models against the hybrid HMM/DNN and HMM/LSTM models. The hybrid systems are constructed by following the standard Kaldi recipes [13]. As with CTC models, inputs of the hybrid model are filterbank features as well. For space limit, we are not describing the details of the hybrid model training. Interested readers can refer to [14] for more details. We observe that on the 110-hour setup, the CTC model performs slightly better than the hybrid DNN model, but is still behind the hybrid LSTM model. In contrast, when we switch to the complete 300-hour setup, the CTC model outperforms both hybrid models by a large margin. This comparison preliminarily shows that the advantage of CTC training becomes more obvious when the amount of training data increases. The validity of this observation needs to be further verified on even larger datasets.

Table 1. *Comparisons of the CTC, hybrid DNN and hybrid LSTM models on the two training sets and with different initializations for the forget-gate bias (FG Bias). "Small random" refers to initialization with small random values, while "constant 1.0" means that the bias vector is set to 1.0.*

Set	Model	FG Bias	WER%
110-Hour	CTC	Small Random	20.7
	CTC	1.0	**19.9**
	Hybrid DNN	—	20.2
	Hybrid LSTM	—	19.2
Complete	CTC	Small Random	15.7
	CTC	1.0	**15.1**
	Hybrid DNN	—	16.9
	Hybrid LSTM	—	15.8

4. ARCHITECTURAL VARIANTS

This section focuses on investigating two architectural variants of the LSTM model. Within the hybrid approach, previous work [15] shows the benefits of combining convolutional neural networks (CNNs) and DNNs with LSTMs. In this paper, we examine this combination in the context of CTC training. Specifically, a 1-dimensional convolution layer along the frequency axis is placed over the input features (i.e., prior to the LSTM layers). This convolution layer is followed by a max-pooling layer which shrinks the sizes of feature maps by 3 times, and finally by the LSTM hidden layers. From Table 2, we can see that this combined architecture, *ConvLSTM*, gives slight improvement (0.3% absolute) over the pure LSTM. However, training of ConvLSTM is observed to be not stable, partly because the outputs from the convolution layer have a high dimension and therefore increase the size of the LSTM layers. More optimization and extensive results will be conducted in our subsequent studies.

As with most of the CTC work, we have used bi-directional LSTMs for CTC training. A criticism of the bi-directional structure lies in the temporal latency, which hampers the deployment in real-world applications. In Table 2, we also present the result when our acoustic model is constructed with uni-directional LSTMs. In this case, the dimension of the memory cell is 640, making the uni-directional model have approximately the same size as the bi-directional network. Applying uni-directional LSTM causes XX.X% relative WER degradation (XX.X vs 19.9%).

Table 2. *Comparisons of various network architectures with CTC training on the 110-Hour Switchboard setup.*

Model	WER%
LSTM	19.9
ConvLSTM	19.6
Uni-directional LSTM	XXX

5. LANGUAGE EXPANSION

We further evaluate CTC training on the HKUST Mandarin Chinese conversational telephone speech recognition task [11]. The training and testing sets contain 174 and 5 hours of speech respectively. The LSTM model contains 5 bi-directional LSTM layers, each of which has 320 memory cells in both the forward and the backward sub-layers. On this setup, CTC models the characters directly. Data preparation gives 3667 labels including English characters, Mandarin characters, noise marks and the blank. Table 3 reveals that CTC training achieves the CER of 35.47%. This number is superior than the numbers (both hybrid HMM/DNN and HMM/LSTM) reported in the Kaldi repository https://github.com/kaldi-asr/kaldi.

Table 3. *%CER of the CTC model achieved on the HKUST Mandarin corpus, and the comparison with the best number reported in the Kaldi repository.*

Model	CER%
CTC	35.47
Kaldi's best number [1]	35.93

6. FRONT-ENDS

In the existing hybrid approach, the inputs of the DNN or LSTM models are enhanced by feature learning based on the GMM models, or by feature enrichment with additional features. This section focuses on more advanced front-ends in addition to the aforementioned filterbank coefficients.

6.1. Speaker Adaptive Features

When building GMM models, we can estimate linear transforms to project the original acoustic features into a speaker adaptive (SA) feature space. Two most commonly used types of transforms are vocal tract length normalization (VTLN) [16] and feature-space maximum likelihood linear regression (fMLLR) [17]. In the hybrid approach, the effectiveness of fMLLR and VTLN features has been sufficiently verified for DNN models. In [14], the hybrid LSTM model with VTLN-transformed filterbanks performs consistently better than the model with the raw filterbanks. In this section, we study the utility of SA features for CTC model training. Specifically, we transform the filterbank features with VTLNs estimated by a GMM model. The LSTM model in CTC is trained over these VTLN-trasformed filterbanks. On the Switchboard setups, Table 4 presents the results of the CTC models with different front-ends. As the case with hybrid systems, for CTC models, the VTLN-FBank front-end generates better WERs than the original FBank features. This confirms that SA features are also applicable to CTC training. Estimating the SA feature with VTLN has the drawback that CTC training now has dependency on GMM models. However, in practice, we may have access to user attributes, such as gender and age, to replace the VTLN factors. These attributes can be exploited to obtain SA features and thus improve CTC acoustic models.

Table 4. *Comparisons of various front-ends with CTC training on the Switchboard setups.*

Set	Model	Feature	WER%
110-Hour	CTC	FBank	19.9
	CTC	VTLN-FBank	19.2
Complete	CTC	FBank	15.1
	CTC	VTLN-FBank	14.6

6.2. Pitch Features for Tonal Languages

Another way to enhance speech front-ends is to integrate different types of features together. In particular, the Pitch features have been found to be beneficial for tonal languages (e.g., Mandarin, Cantonese and Vietnamese) [18]. On our Mandarin setup (Section 5), we propose to incorporate the Pitch features into CTC model training. The Pitch features are extracted using the method described in [18]. Append the 3-dimensional Pitch and the 40-dimensional FBank features gives us a 43-dimensional feature vector at each frame. On the Mandarin test set, the CTC model with these appended features obtains the CER of 34.79%, outperforming the CTC model only with FBank.

Table 5. *%CER of the CTC model on the HKUST Mandarin corpus with different features.*

Feature	CER%
FBank	35.47
FBank+Pitch	34.79

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an extensive study regarding the CTC technique for training acoustic models. To be aligned with the aspects listed in Section 1, our conclusions can be summarized as follows. 1) Initializing the bias vector of the LSTMs forget gates to a large value (1.0) is found to give nice gains. Also, the advantage of CTC gets more obvious on larger amounts of training data. 2) The ConvLSTM architecture, with a convolution layer inserted before the LSTM layers, achieves slight improvement over the vanilla LSTM. Switching from the bi-directional to the uni-directional LSTM degrades the recognition accuracy by 10%. 3) The performance of CTC models can be improved by speaker adaptive front-ends, or by front-ends enriched with additional feature types. 4) The application of CTC results in the state-of-the-art performance on the HKUST Mandarin corpus.

For our future work, we are interested to investigate the characteristics of CTC in the decoding stage, e.g., how to perform speaker adaptation [14] for CTC models. Also, we would like to extend the convolution in the ConvLSTM architecture to both the time [19] and the frequency dimensions.

8. ACKNOWLEDGEMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.This work was partially funded by Facebook, Inc. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Facebook, Inc

9. REFERENCES

[1] George E Dahl, Dong Yu, Li Deng, and Alex Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

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

[3] Alex Graves, Navdeep Jaitly, and A-R Mohamed, “Hybrid speech recognition with deep bidirectional LSTM,” in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[4] Hasim Sak, Andrew Senior, and Françoise Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[5] 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.

[6] Alex Graves and Navdeep Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1764–1772.

[7] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., “Deepspeech: Scaling up end-to-end speech recognition,” *arXiv preprint arXiv:1412.5567*, 2014.

[8] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, “Fast and accurate recurrent neural network acoustic models for speech recognition,” *arXiv preprint arXiv:1507.06947*, 2015.

[9] Yajie Miao, Mohammad Gowayyed, and Florian Metze, “Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding,” *arXiv preprint arXiv:1507.08240*, 2015.

[10] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2342–2350.

[11] Yi Liu, Pascale Fung, Yongsheng Yang, Christopher Cieri, Shudong Huang, and David Graff, "HKUST/MTS: a very large scale mandarin telephone speech corpus," in *Chinese Spoken Language Processing*, pp. 724–735. Springer, 2006.

[12] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý, "The Kaldi speech recognition toolkit," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 1–4.

[14] Yajie Miao and Florian Metze, "On speaker adaptation of long short-term memory recurrent neural networks," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2015.

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

[16] Puming Zhan and Alex Waibel, "Vocal tract length normalization for large vocabulary continuous speech recognition," Tech. Rep., DTIC Document, 1997.

[17] Mark JF Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[18] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2494–2498.

[19] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang, "Phoneme recognition using time-delay neural networks," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, 1989.