AN EMPIRICAL EXPLORATION OF CTC ACOUSTIC MODELS

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

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

ABSTRACT

The connectionist temporal classification (CTC) loss function has several interesting properties relevant for automatic speech recognition (ASR): applied on top of 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. CTC systems also do not require context decision trees for good performance, using context-independent (CI) phonemes or characters as targets. This paper presents an extensive exploration of CTC-based acoustic models applied to a variety of ASR tasks, including an empirical study of the optimal configuration and architectural variants for CTC. We observe that on large amounts of training data, CTC models tend to outperform state-of-the-art hybrid approach. Further experiments reveal that CTC can be readily ported to syllable-based languages, and can be enhanced by employing improved feature front-ends.

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

1. INTRODUCTION

The introduction of deep neural networks (DNNs) and recurrent neural networks (RNNs) as acoustic models has brought tremendous progress to automatic speech recognition (ASR) [1, 2, 3]. In the *hybrid* approach, DNNs/RNNs are used to classify speech frames to context-dependent (CD) states, i.e., senones. These states (and the corresponding training labels) are generally derived from a "seed" Gaussian mixture model (GMM) through forced alignment. Model training can then be carried out with the cross-entropy (CE) objective function. Connectionist temporal classification (CTC) [4] 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 connect 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-level labels and thereby the building of the initial GMMs. Used together with deep RNN models, CTC has been shown to achieve state-of-the-art performance on large-scale English acoustic modeling tasks [5, 6, 7, 8].

Despite showing convincing performance, CTC is still less well understood than the existing hybrid approach. For instance, the application of CTC on languages other than English has not been fully explored in the literature. This paper presents an empirical study to investigate how CTC behaves under various conditions. We focus on the following aspects:

Optimal configuration: CTC commonly uses deep RNNs with Long Short-Term Memory (LSTM) units as acoustic models. Motivated by past work on LSTMs [9], we initialize the bias vector of the LSTMs forget gates to larger values, which brings consistent gains for CTC training. Our experiments also reveal how the amount of training data affects the performance of CTC models.

Architectural variants: We study two architectural variants of CTC models. First, a convolution layer is added 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 dramatically worse than the bi-directional one.

Language Expansion: We report CTC results on a task of transcribing Chinese Mandarin conversational telephone speech [10]. By directly modeling thousands of Mandarin characters, CTC achieves competitive results on this task.

Front-Ends: Apart from the raw acoustic features (e.g., MFCCs, filterbanks), the hybrid approach can exploit advanced front-ends (e.g., fMLLRs, VTLNs). This paper empirically verifies the applicability of these front-ends in the context of CTC modeling.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. This research was performed as part of the Speech Recognition Virtual Kitchen project, which is supported by the United States National Science Foundation under grant number CNS-1305365. 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.

2. REVIEW OF CTC

Connectionist temporal classification (CTC) [4] is a loss function for sequence labeling problems where the inputs and the label sequences have variable lengths. When applied to acoustic modeling, 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 [5, 6, 8], the acoustic models used together with CTC are normally deep RNNs with LSTM units [11] (which we will consistently refer to as LSTMs). The nodes in the softmax layer correspond to the original labels, as well as a special *blank* label which estimates the probability of emitting no labels. CTC trains the LSTM model to maximize $\ln Pr(z|x)$, the log-likelihood of the reference 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 occurrences of blanks and repetitions of the non-blank labels. The label sequence z can be represented by a set of all the possible CTC paths that are mapped to z. The likelihood is then evaluated as an aggregation of the probabilities of the 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. $Pr(z|x)$ can now 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:

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

where the *forward variable* α_t^u represents the total probability of all CTC paths that end with label u at frame t, and can be recursively computed with the α values from the previous frame $t-1$. Similarly, the *backward variable* β_t^u carries the total probability of all CTC paths that starts with label u at t, and can be computed with the β values from the following frame $t+1$.

The loss function becomes differentiable with respect to the LSTM outputs. The quantity y_t^k represents the posterior of the label k outputted by the LSTM network. The gradients of $\ln Pr(z|x)$ with respect to y_t^k can be computed as

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

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 gradients are taken as the errors that will be back-propagated into the LSTM model for parameter updating.

Until recently, efficient decoding of CTC acoustic models has been a challenge because of the blank label. Our previous work [8] proposes a general decoding method based on weighted finite-state transducers (WFSTs) for CTC models. 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 blank label, while enabling the effective and efficient incorporation of word language models into CTC decoding.

3. OPTIMAL CONFIGURATION

3.1. Experimental Setup

Our experiments in this section are conducted on the Switchboard conversational telephone transcription task. 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 change of the accuracy of the hypothesis labels with respect to the reference label sequences. CTC training models context-independent (CI) phonemes. Totally, we have 46 labels including phonemes, noise marks and the blank.

Our decoding follows the WFST-based approach introduced in [8]. The label posteriors generated by the LSTM model are scaled with the label priors estimated from the (expanded) label sequences. A trigram language model is trained on the training transcripts, which is then interpolated with another language model 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 to 0 or small random weights. Although working

well on many applications, 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 [9]. In our experiments, we set the initial values of the forget gates bias vector uniformly to 1.0. From Tables 1 and 2, we can see that this initialization brings consistent improvement over the initialization with small random values. The word error rate (WER) is improved by 3.9% and 4.6% respectively on the 110-hour and 300-hour setups.

We compare the CTC models against hybrid HMM/DNN and HMM/LSTM models. These hybrid models have been built by following the standard Kaldi nnet1 recipes [12]. On the 110-hour setup, the DNN has 5 hidden layers each of which contains 1200 neurons. The LSTM model has 2 uni-directional LSTM layers where linear projection layers are applied over the hidden outputs. Each LSTM layer has 800 memory cells and 512 output units. Parameters of both the DNN and LSTM models are randomly initialized. On the 300-hour setup, the DNN model has 6 hidden layers each of which contains 2048 neurons. The LSTM model has 2 projected LSTM layers, where each LSTM layer has 1024 memory cells and 512 output units. The DNN is initialized with restricted Boltzmann machines (RBMs), while the LSTM model is randomly initialized. As with the CTC models, inputs of the hybrid models are filterbank features. More details about these hybrid models can be found in [13].

Tables 1 and 2 show 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. This comparison indicates that CTC training becomes more advantageous 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 Switchboard 110-hour training set, with different initializations for the forget-gate bias (FG Bias). "Small Random" refers to initialization with small random values, while "1.0" means that the bias vector is uniformly set to 1.0. M refers to million.*

Model	#Param	FG Bias	WER%
CTC	8M	Small Random	20.7
CTC	8M	1.0	19.9
Hybrid DNN	12M	—	20.2
Hybrid LSTM	8M	—	**19.2**

4. ARCHITECTURAL VARIANTS

In the hybrid approach, previous work [14] 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 size of the feature maps by 3 times, and finally by the LSTM hidden layers. From Table 3, we can see that this combined architecture, *ConvLSTM*, gives slight improvement over the pure LSTM. However, training of ConvLSTM is presently unstable, partly because the outputs from the convolution layer have a high dimension and therefore increase the size of the LSTM layers. This architectural combination will be further studied in our subsequent work.

Table 2. *Comparisons of the CTC, hybrid DNN and hybrid LSTM models on the Switchboard complete training set*

Model	#Param	FG Bias	WER%
CTC	11M	Small Random	15.7
CTC	11M	1.0	**15.0**
Hybrid DNN	40M	—	16.9
Hybrid LSTM	12M	—	15.8

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 3, 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 LSTMs causes 17.1% relative WER degradation (23.3% vs 19.9%), similar to what was observed in [7].

Table 3. *Comparisons of architectural variants with CTC training on the 110-hour Switchboard setup.*

Model	WER%
LSTM	19.9
ConvLSTM	19.6
Uni-directional LSTM	23.3

5. LANGUAGE EXPANSION

5.1. Mandarin

We evaluate CTC training on the HKUST Mandarin Chinese conversational telephone ASR task [10]. In our experiments, the training and testing sets contain 174 and 5 hours of speech respectively. The acoustic model contains 5 bi-directional LSTM layers, each of which has 320 memory cells in both the forward and the backward sub-layers. Instead of phonemes, CTC on this setup models characters directly. Data preparation gives us 3667 labels including English characters, Mandarin characters, noise marks and the

blank. A trigram language model is employed in the WFST-based decoding. From Table 4, we can see that CTC training achieves a CER of 39.70%. This number is comparable to the hybrid HMM/DNN system (39.42%) which is trained with the CE objective and over the speaker-adaptive (SA) features, as reported in the Kaldi repository [12]. This observation is on the contrary to [15] where CTC is found to perform much worse than the hybrid models, due to the lack of word language models in decoding.

Table 4. *Comparisons of the CTC model and the hybrid HMM/DNN model (whose number is reported in the Kaldi repository) on the HKUST Mandarin corpus. The evaluation metric is character error rate (CER).*

Model	CER%
CTC	39.70
Hybrid DNN	39.42

6. FRONT-ENDS

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

6.1. Speaker Adaptive Features

When building GMM models, we can estimate linear transforms to project the original acoustic features into a SA feature space. Two most commonly used types of transforms are vocal tract length normalization (VTLN) and feature-space maximum likelihood linear regression (fMLLR). In the hybrid approach, the effectiveness of fMLLR and VTLN features has been sufficiently verified for DNN models. In [13], the hybrid LSTM model with VTLN-transformed filterbanks performs consistently better than the model with the original 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 the VTLN-trasformed filterbanks. Table 5 presents the results of the CTC models with different front-ends on the Switchboard setups. As with the hybrid models, the VTLN-filterbank front-end also generates better WERs than the original filterbank 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 5. *Comparisons of various front-ends with CTC training on the Switchboard setups.*

Set	Model	Feature	WER%
110-hour	CTC	filterbank	19.9
	CTC	VTLN-filterbank	19.2
300-hour	CTC	filterbank	15.0
	CTC	VTLN-filterbank	14.5

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) [16]. On our Mandarin setup (Section 5), we incorporate the pitch features into CTC model training. The pitch features are extracted using the method described in [16]. On each frame, appending the 3-dimensional pitch to the 40-dimensional filterbank features gives us a 43-dimensional feature vector. From Table 6, we observe that the CTC model with the appended features obtains the CER of 38.67%, outperforming the CTC model only with filterbanks.

Table 6. *%CER of the CTC model on the HKUST Mandarin corpus with different features.*

Feature	CER%
filterbank	39.70
filterbank+pitch	38.67

7. CONCLUSIONS AND FUTURE WORK

In this paper, we conduct an extensive study of the CTC approach to training acoustic models. We present several improvements and observations: 1) Initializing the bias vector of the LSTMs forget gates to large values (1.0) improves performance. Also, the advantage of CTC gets more pronounced 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 LSTMs. Switching from bi-directional to uni-directional LSTMs degrades recognition accuracy significantly. 3) The performance of CTC models can be further improved by speaker adaptive front-ends, or by front-ends enriched with additional feature types. 4) Using characters as targets, CTC achieves competitive performance on a Mandarin ASR task.

For future work, we will investigate how to perform adaptive training [17, 18] and speaker adaptation [13] for CTC acoustic models. Also, we would like to extend the convolution in the ConvLSTM architecture to both the time [19] and the frequency dimensions.

8. 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] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[3] 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.

[4] 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*, 2006, pp. 369–376.

[5] 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.

[6] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, et al., "Deepspeech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[7] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2015.

[8] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015.

[9] 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.

[10] 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. 2006.

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

[12] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, et al., "The Kaldi speech recognition toolkit," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 1–4.

[13] 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.

[14] 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.

[15] Jie Li, Heng Zhang, Xinyuan Cai, and Bo Xu, "Towards end-to-end speech recognition for chinese mandarin using long short-term memory recurrent neural networks," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2015.

[16] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, et al., "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.

[17] Yajie Miao, Hao Zhang, and Florian Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*. ISCA, 2014.

[18] Yajie Miao, Hao Zhang, and Florian Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 1938–1949, 2015.

[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.