EXPLORING CTC-NETWORK DERIVED FEATURES WITH CONVENTIONAL HYBRID SYSTEM

Thai-Son Nguyen, Sebastian Stüker, Alex Waibel

Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology

ABSTRACT

Recently in automatic speech recognition (ASR) a lot of attention has been given to decoding optimization to boost the performance of connectionist temporal classification criterion (CTC) systems in all neural setups. Different from that, we investigated the use of the output of CTC network as input features to traditional HMM/ANN hybrid systems. By doing so, we benefit from the strengths of the CTC network at label discrimination and the highly optimized decoding stack of conventional hybrid systems. In a Switchboard setup, a feed-forward network system using our proposed CTC-network derived features with cross-entropy training outperforms a strong CTC baseline by a margin of 5% rel. in word error rate. With the same model, we achieved further improvements of 9% rel. when combining them with bottleneck features. Additionally, we revealed the possible elimination of the $blank$ label during decoding and the alignment relationship between the CTC model and the traditional HMM system.

Index Terms— CTC, posterior probability, feature extraction, feature combination

1. INTRODUCTION

The connectionist temporal classification criterion (CTC) [1, 2] and its associated training methods have received significant interest in speech recognition in recent years. Using recurrent neural networks (typically long short-term memory LSTM), CTC training can efficiently model the long-term dependencies between a small number of units (e.g., phonemes or characters) and speech frames. Utilizing the CTC optimization criterion which handles possible alignments of the units in a sequence label, CTC-based speech recognition systems can be trained in a straight manner, thereby eliminating many complex steps in the conventional hybrid Hidden Markov Modell / Artificial Neural Network (HMM/ANN) speech recognition pipeline, such as the definition of an HMM topology, finding context-dependent phonemes and modeling units, and the frame-wise alignment of HMM states and feature vectors. The way CTC was originally introduced motivates the development of such speech recognition systems in end-to-end fashion in which the language model or a vocabulary can also be omitted. However, to achieve state-of-the-art performance at par with the traditional hybrid HMM/ANN approach, an efficient decoding algorithm that uses additional knowledge sources (e.g., vocabulary, pronunciation lexicon and language model) is still required to transform the posteriors of the units modeled by the CTC network into word sequences.

Because the training criterion of the CTC model is to maximize the log posterior $P(z|X)$ of the target label z given acoustic features, it does not necessarily optimize the final recognition when decoding with an additional language model. To the best of our knowledge, a decoding with a weighted finite state transducer (WFST) built over a pronunciation lexicon and an n-gram language model applied to CTC posteriors is still the most successful approach with the best word error rate (WER). As observed in [3, 4], the performance of a CTC system can be better than that of a hybrid system trained with the cross-entropy criterion but is less than that of a hybrid system optimized with sequence training.

To benefit from the strengths of the CTC network at label discrimination on the one side and the highly optimized decoding stack of conventional hybrid systems on the other side, we investigate in this paper the use of CTC posterior probabilities as input features in hybrid HMM/ANN system to boost speech recognition performance.

2. RELATED WORK

In speech recognition, the posterior output of multiple-layer perceptrons (MLP) was originally proposed as input features to Tandem GMM models [5]. This approach of feature extraction was enhanced by combining MLP features with the original speech features [6]. Later, when the multiple HMM states per phone [7] and context-dependent states were introduced, bottleneck features [8], a small layer in the middle of the MLP, was used for instead. Bottleneck features usually performed best when being concatenated with traditional acoustic features in Tandem GMM systems [9, 10].

Our approach in this paper is similar to the Tandem approach as we also use the output of a neural network, an LSTM in our case, as input features to an HMM based speech recognition system, but it differs in that way that we use recurrent networks utilizing the CTC training criterion for creating the features, and that we are not training an HMM/GMM model but an HMM/ANN model instead.

3. C-PHONE EXTRACTION

Assume that we use a set of labels L and we can always map the ground-truth transcript of an utterance X into a label sequence $z \in L^*$ (L^* meaning the Kleene closure over the alphabet L). A CTC path π (i.e., a sequence at frame level allowing repeated labels) represents an alignment of z. Denote y_t^π as the posterior probability that a recurrent neural network model assigns to the corresponding label of π at time t. By assuming the independent probabilities of all labels between frames, the CTC objective function solves all possible alignments as:

$$P(z|X) = \sum_{\pi} P(\pi|X) = \sum_{\pi} \prod_{t} y_t^\pi$$

For model optimization, [1, 2] proposed to use the forward-backward algorithm to maximize the likelihood of all the transcripts given the speech utterances in a training corpus. After training converges, we obtain an optimized model to predict the posteriors y_t of all labels at every time frame t.

By using independent phones as the set of labels (e.g., 45 English phones), we consider y_t as a phone information vector (so-called *C-Phone*) which indicates the occurrence of the phones in the frames. Since the posterior probabilities extracted from the softmax output usually have sharp distribution [5, 6], we transform them to the log domain for better modelling. During CTC training, the $blank$ label must be introduced to allow the optional occurrences of regular labels in the alignments. The probability of the $blank$ can be eliminated, i.e. removed from the C-Phone, when extracting the C-Phone vector since it does not map to any real acoustic event and has little meaning.

Different from the extracted posterior features [5, 6] or bottleneck features [8] where the extracting models are trained with fixed Viterbi alignments (e.g., the model exactly learns a feature transformation), C-Phone extraction is trained without any prior alignment (the CTC model needs learns the alignment by itself based on the label sequences). As observed in [1, 11, 12], the posteriors produced by the CTC model have peaky behaviors in which $blank$ has the highest probability in almost all frames, except for short peaks where regular labels dominate. This raises the question whether the phone probabilities assigned by the CTC model still correlate to the fixed labels of a traditional Viterbi alignment. In this study, we try to address this question by learning a feed-forward network transformation to bridge between C-Phones and the context-dependent phones labels in the conventional HMM system set-up.

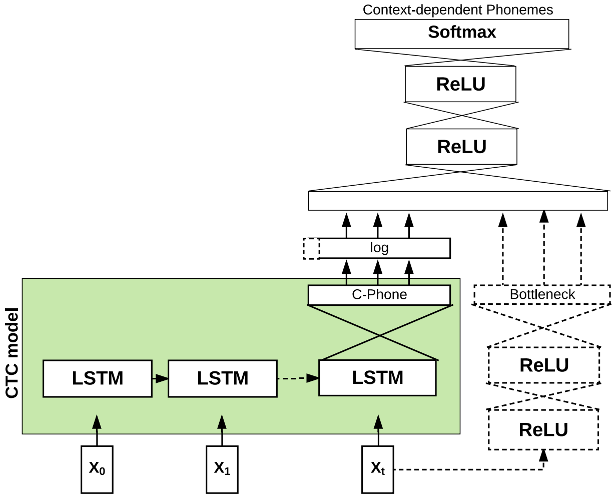

Fig. 1. Extracting and using C-Phone.

4. USING C-PHONE FEATURES

Figure 1 illustrates how we investigated the C-Phone features. We trained a CTC system with an LSTM model for feature extraction as explained in Section 3. The trained LSTM model is then used to produce posterior vectors for every frame. These posteriors are transformed into the log domain and the probability of the $blank$ label is eliminated to form the final C-Phone feature vectors. These features can be directly fed into a feed-forward network model to build a conventional hybrid HMM/ANN system. Furthermore, C-Phone features can also be augmented with additional features such as network-based bottleneck features.

5. EXPERIMENTAL SETUPS

Our experiments were conducted on the Switchboard-1 Release 2 (LDC97S62) training corpus which contains over 300 hours of speech. The Hub5'00 evaluation data (LDC2002S09) was used as test set. We used a 4-gram language model which was trained on the transcripts of the training data (3M words) and the transcripts (22M words) from the Fisher English Part 1 (LDC2004T19) and Part 2 (LDC2005T19) corpora. We further used the pronunciation dictionary that came with the Switchboard Corpus.

All our systems were trained on the same training data and use the same vocabulary and 4-gram language model. The dictionary used for decoding includes 43 English phonemes and 2 noise models. For the CTC training, $blank$ is used as additional label while for the hybrid HMM/ANN system we use $silence$ instead.

The CTC systems used for C-Phone extraction was trained with Eesen [13]. We used a bi-directional LSTM with 5 layers of 320 units, and a uni-directional LSTM containing 640 units per layer (see Section 6.3). The training schedule adopts an initial learning rate of 0.00004 for every training. A decay with a factor of 4 was applied when the cross validation error degraded after 12 epochs.

We used Janus Recognition Toolkit (JRTK) [14] to train and decode the feed-forward neural network (FFNN) systems. A FFNN architecture of 7 layers of 1600 units is used for all hybrid HMM/ANN systems. The training of FFNN models uses new bob learning rate schedule with an initial rate of 0.02.

Similar to other FFNNs, the bottleneck extraction network is also trained on 11 frames of log mel filter-bank features which are normalized per conversation. The bottleneck layer contains 40 units which is the same as the number of filter-bank coefficients. The extraction network also has 7 layers and the 2 last layers are removed after the training. A feature-space Maximum Likelihood Linear Regression (fMLLR) transformation was estimated from the manual transcripts during the training and from high-confidence decoding transcripts during testing.

6. RESULTS

6.1. C-Phone Features

Table 1 compares the results of multiple systems using C-Phone features against conventional hybrid systems with log mel filter-bank (FBank), bottleneck features (BNF) and fMLLR features which are estimated on top of the BNF. For fair comparison, all the systems share the same feed-forward neural network (FFNN) architecture for classifying 8000 context-dependent phonemes on a fixed Viterbi alignment. We use a popular FFNN as the baseline which was trained with the cross-entropy criterion on 11 frames of FBank coefficients. The referenced CTC system is trained using Eesen [13] and also uses Eesen’s WFST functionality to decode on the same posteriors as used for C-Phone extraction. The same 4-gram language model is employed in all systems. The results are reported on the full Hub5’00 test set. We noticed in our experiments that our baseline CTC system performs slightly better than a very similar system recently reported in [15].

We experimented with 3 variants of C-Phone features. The first variant is the direct posterior probabilities (C-Phone-P) while the second variant (C-Phone-L) is obtained after transforming the softmax output to the log domain. The third variant (C-Phone-NB) is the same as C-Phone-L before eliminating the probability of the $blank$ unit. In our setup, the training of the FFNN systems on C-Phone-P features did not converge. However when we switched to C-Phone-L or C-Phone-NB, our training converges well for all inputs of different context sizes and without applying further feature normalization techniques.

Even when using only a singe C-Phone vector as input, an FFNN can even be trained well. This reveals an additional aspect to the peaky behavior observed in CTC training [1, 11, 12], e.g., even for the frames when no (regular) label has its peak probability, the posteriors vector still contains meaningful information for classifying phonemes (or even context-dependent phonemes) labeled in the fixed alignment manner.

Model	Features	Window	Hub5’e (SWB)
FFNN	FBank	11	22.4 (15.8)
CTC	FBank	-	19.9 (14.1)
FFNN	C-Phone-P	-	-
	C-Phone-L	1	19.3 (13.7)
	C-Phone-L	7	19.0 (13.6)
	C-Phone-L	11	**18.9 (13.5)**
	C-Phone-L	15	19.3 (13.8)
	C-Phone-NB	1	19.3 (13.8)
	C-Phone-NB	7	19.0 (13.6)
	C-Phone-NB	11	19.1 (13.6)
	BNF	1	22.7 (16.0)
	BNF	7	21.8 (15.3)
	BNF	11	21.5 (15.1)
	BNF	15	21.5 (15.1)
	fMLLR-BNF	11	21.0 (14.6)
GMM	C-Phone-L	1	20.9 (15.7)
	C-Phone-L	11	20.0 (14.5)
	BNF	11	22.1 (15.7)

Table 1. Performance in word error rate (WER) of multiple HMM/ANN systems with different input features such as C-Phone, FBank, BNF, and fMLLR-BNF

Interestingly, the performance of the systems with C-Phone-L and C-Phone-NB are almost identical for the same configurations. This may indicate that the probability of the $blank$ does not carry any useful information for phoneme classification, and thus can be eliminated during decoding. This observation consolidates the identification in [16].

In terms of word error rate (WER), the FFNN systems trained on C-Phone outperform FBank by a large margin (15.6% rel.) and clearly improve over the other network-based extracted features such as BNF or fMLLR. The improvement of stacking longer context of C-Phone vectors appears small but is still effective. The extracted C-Phone features also show their usefulness over other features when switching to GMMs instead of FFNNs.

In our experiments, the FFNN systems trained on the C-Phone outperform the referenced CTC system even when using only one feature verctor per frame. It is worth noting that the superior performance of the FFNNs is observed here with cross-entropy training (further improvement is expected when optimizing the FFNNs with sequence training). This result can be explained by either the introduction of context-dependent phonemes, that helps improving the classification, or by the current decoding approach of the CTC system not being as good as that of conventional HMM system.

6.2. Feature Stream Combination

As shown in Section 6.1, C-Phone is a compact vector which contains excellent features for phonemes classification. As a typical approach of feature engineering, one can wonder if the recognition performance can be further improved by combining additional features to C-Phone. In this section, we investigate the combination of C-Phone and FBank, BNF and fMLLR features. Table 2 presents the results of different combinations. We report only with C-Phone-L features but other variants have the same results. The *Window* column shows the number of consecutive C-Phone vectors and additional feature vectors (*+Feature*) fed into the FFNNs. Basically, we allow only two different input streams and the center of the context window is always the current frame.

+Features	Window	Hub5'e (SWB)
FBank	1/1	23.0 (17.7)
	3/3	18.9 (13.6)
	5/5	19.1 (13.7)
	1/5	19.1 (13.7)
BNF	1/1	18.4 (13.1)
	2/2	18.2 (12.9)
	3/3	18.4 (13.1)
	5/5	18.6 (13.3)
	1/5	18.5 (13.1)
fMLLR-BNF	1/1	**18.1 (12.8)**
	2/2	18.2 (13.0)
	3/3	18.2 (13.1)
	5/5	18.3 (13.2)
	1/5	18.2 (12.9)

Table 2. Results in WER of different feature combinations.

Combining C-Phone with FBank features has almost the same result as using single C-Phone features. This result is different from [6] where the combination of PLP features and their derived multiple layer perceptron (MLP) features gave improvements. This indicates that C-Phone does not need the complementary information from the original speech features for phoneme classification.

We found that the other network-based extracted features such as BNF features can supplement C-Phone and result in a better recognition performance (4.2% rel.). This can be explained by the fact that BNF which is extracted from a wide context window contains additional information for context-dependent phoneme classification.

When transforming BNF into the fMLLR feature space which has less speaker variability, we achieved remarkable result (see Table 1). However, the recognition performance stays more or less the same when combining with C-Phone with BNF or fMLLR features. This observation can be explained as the analysis in [6] where the extracted posteriors features reduce the variation among speakers, and thus have similar effects as fMLLR.

In many modern speech recognition systems, i-vector [17] which contains the information about the speaker and environment in a short vector usually helps supplementing the traditional features such as FBank or fMLLR [18, 19] in a speaker adaptation manner. Unfortunately, we was not able to provide results with i-vector adaptation due to our i-vector training setup could not employ Switchboard (or also Fisher) corpus to produce efficient i-vectors. We also found the same observation as reported in [20].

6.3. Using Uni-directional LSTM

So far for all the results, we used the bi-directional LSTM to train C-Phone extraction. However the bi-directional LSTM structure which requires the temporal context of a whole utterances is not an optimal choice for real-time and low latency applications. In Table 3, we present the results of the systems using the C-Phone features which are extracted from a uni-directional LSTM. Switching to the uni-directional structure, the performance of the CTC system and the FFNN with C-Phone features degrades severely (20% and 26% rel.) and becomes lower than that of the conventional FFNN system. We only achieve some small improvement (3.2% rel.) when using C-Phone to complement the bottleneck features.

Model	+Features	Window	Hub5'e (SWB)
CTC		-	25.4 (17.8)
FFNN		1	35.4 (28.2)
		11	25.5 (18.6)
	FBank	2/2	25.4 (18.8)
	FBank	3/3	24.8 (18.2)
	BNF	1/1	21.9 (15.8)
	BNF	2/2	21.2 (15.3)
	BNF	3/3	21.0 (15.0)

Table 3. Results of the systems using C-Phone extracted from uni-directional LSTM.

7. CONCLUSION AND FUTURE WORK

We have investigated the use of the posterior probabilities extracted from a CTC network as features in a conventional hybrid HMM/ANN speech recognition system. Our experiments show that the extracted posteriors are excellent features for content-dependent phonemes classification and speech recognition. While we experimented only with a phone set as CTC labels, future work can explore the performance of different label sets. We are also going to examine the gain when performing sequence training as well as the performance of the presented systems on different training data sets.

8. REFERENCES

[1] 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.

[2] 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.

[3] Haşim Sak, Félix de Chaumont Quitry, Tara Sainath, Kanishka Rao, et al., "Acoustic modelling with cd-ctc-smbr lstm rnns," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 604–609.

[4] Naoyuki Kanda, Xugang Lu, and Hisashi Kawai, "Minimum bayes risk training of ctc acoustic models in maximum a posteriori based decoding framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4855–4859.

[5] Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*. IEEE, 2000, vol. 3, pp. 1635–1638.

[6] Qifeng Zhu, Barry Chen, Nelson Morgan, and Andreas Stolcke, "On using mlp features in lvcsr," in *Eighth International Conference on Spoken Language Processing*, 2004.

[7] Petr Schwarz, Pavel Matějka, and Jan Černockỳ, "Towards lower error rates in phoneme recognition," in *Text, Speech and Dialogue*. Springer, 2004, pp. 465–472.

[8] Frantisek Grézl, Martin Karafiát, Stanislav Kontár, and Jan Cernocky, "Probabilistic and bottle-neck features for lvcsr of meetings," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, 2007, vol. 4, pp. IV–757.

[9] Dong Yu and Michael L Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[10] Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3377–3381.

[11] Haşim Sak, Andrew Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4280–4284.

[12] Albert Zeyer, Eugen Beck, Ralf Schlüter, and Hermann Ney, "Ctc in the context of generalized full-sum hmm training," 2017.

[13] 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, pp. 167–174.

[14] Michael Finke, Petra Geutner, Hermann Hild, Thomas Kemp, Klaus Ri es, and Martin Westphal, "The karlsruhe VERBMOBIL speech recognition engine," in *Proc. of ICASSP*, 1997.

[15] Kartik Audhkhasi, Bhuvana Ramabhadran, George Saon, Michael Picheny, and David Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," *arXiv preprint arXiv:1703.07754*, 2017.

[16] Zhehuai Chen, Yimeng Zhuang, Yanmin Qian, and Kai Yu, "Phone synchronous speech recognition with ctc lattices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 90–101, 2017.

[17] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[18] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors.," in *ASRU*, 2013, pp. 55–59.

[19] Andrew Senior and Ignacio Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[20] Yajie Miao, Lu Jiang, Hao Zhang, and Florian Metze, "Improvements to speaker adaptive training of deep neural networks," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 165–170.