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

CTC in the Context of Generalized Full-Sum HMM Training

Albert Zeyer, Eugen Beck, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52062 Aachen, Germany
{zeyer, beck, schlueter, ney}@cs.rwth-aachen.de

Abstract

We formulate a generalized hybrid HMM-NN training procedure using the full-sum over the hidden state-sequence and identify CTC as a special case of it. We present an analysis of the alignment behavior of such a training procedure and explain the strong localization of label output behavior of full-sum training (also referred to as peaky or spiky behavior). We show how to avoid that behavior by using a state prior. We discuss the temporal decoupling between output label position/time-frame, and the corresponding evidence in the input observations when this is trained with BLSTM models. We also show a way how to overcome this by jointly training a FFNN. We implemented the Baum-Welch alignment algorithm in CUDA to be able to do fast soft realignments on GPU. We have published this code along with some of our experiments as part of RETURNN, RWTH's extensible training framework for universal recurrent neural networks. We finish with experimental validation of our study on WSJ and Switchboard.

Index Terms: CTC, LSTM, full-sum, Baum-Welch

1. Introduction

In speech recognition, the connectionist temporal classification criterion (CTC) [1] with its associated training method is sometimes used instead of a hidden Markov model (HMM) with framewise training based on Viterbi alignments, esp. in the context of hybrid modeling with artificial neural networks (NN's). CTC usually works on single-state phoneme labels including a special blank symbol and its optimization criterion sums over all possible alignments, whereas a hybrid HMM-ANN usually has multiple states per phoneme and works with fixed Viterbi alignments.

We show that CTC training is a special case of a generalized HMM training procedure which we will formulate and work out in this paper. The CTC topology can be interpreted as a special HMM topology without transition probabilities. The CTC training corresponds to framewise training with Baum-Welch realignment, i.e. summing up the probabilities over all possible alignments. We will refer to this as full-sum training.

If the training of the model converges, the alignment will also converge. We refer to this as the optimal alignment by the model and we will discuss it under different conditions. We will explain the peaky behavior which is usually observed with CTC where the frame label with the highest probability is blank for almost all frames, except for very short peaks where the normal labels dominate. This is esp. true when trained with (bidirectional) long short-term memory ((B)LSTM) models. We also show methods to overcome this by jointly training a feed-forward NN (FFNN).

Depending on whether dividing by a state prior probability in the hybrid neural network emission model or not, a generative or a discriminative model is obtained. We briefly discuss the differences in the ensuing alignment behavior.

We implemented the alignment procedure using the Baum-Welch algorithm in CUDA as well as a Theano [2] wrapper around it. This allows us to perform training with only modest slowdown compared to using a fixed alignment. The whole training method is available in RETURNN, RWTH's extensible training framework for universal recurrent neural networks [3].

We present experiments on the Switchboard and WSJCam0 corpus with full-sum training from a randomly initialized model with the conventional HMM topology (3 states per tri-phone) instead of the CTC topology (1 state per phoneme + blank).

2. Related work

Alex Graves introduced CTC in [1] with more details in [4] for RNNs. Many works on CTC followed such as [5–13]. It has been shown that context-dependent (CD) phone models (i.e. 1 state per phone) as opposed to classic 3-state HMM models perform better with CTC [14]. Also, state-level minimum Bayes risk (sMBR) sequence-discriminative training on top of a Viterbi-trained hybrid HMM still performs better than CTC, but state-level minimum Bayes risk (sMBR) applied on top of CTC performs even better [12, 15, 16]. In addition, CTC seems to improve over the Viterbi-trained hybrid HMM only if enough training data is used [7, 11, 12, 17–19].

The first full-sum training with neural networks that we are aware of was demonstrated in [20]. Other similar work was shown in [21–24]. Flat-start training with frequent realignments was also discussed in [25–27]. The CTC/HMM expressiveness to map long input sequences to short output sequences can also be achieved by the even more powerful encoder-decoder model with optional attention [28, 29]. Other alternatives are segmental models which are described in [30–32]. The lattice-free MMI method described in [33] can be seen as an extension to CTC or full-sum HMM training without prior but with a phone-level language model. In [11, 34], a prior is included in decoding but not in training, i.e. not in the Baum-Welch alignment calculation.

3. Generalized HMM training

3.1. Model and Training Criteria

The likelihood of observing an acoustic vector sequence x_1^T given a word sequence w_1^N with state sequences s_1^T as hidden variable and model parameters θ is given by:

$$\begin{aligned} L(\theta) &:= p(x_1^T|w_1^N,\theta) \\ &= \sum_{s_1^T:w_1^N} p(x_1^T, s_1^T|w_1^N,\theta) \\ &\overset{\text{model}}{=} \sum_{s_1^T:w_1^N} \prod_{t=1}^{T} p_t(s_t|s_{t-1},w_1^N)\cdot p_t(x_t|s_t,\theta) \end{aligned} \tag{1}$$

http://dx.doi.org/10.21437/Interspeech.2017-1073

Using the maximum-likelihood criterion we obtain the following derivative:

$$\frac{\partial}{\partial\theta}\log L(\theta) = \frac{1}{L(\theta)}\frac{\partial}{\partial\theta}L(\theta)$$
$$= \frac{1}{L(\theta)}\sum_{t,s}\frac{\partial L(\theta)}{\partial p_t(x_t|s_t,\theta)}\cdot\frac{\partial p_t(x_t|s_t,\theta)}{\partial\theta}$$
$$= \frac{1}{L(\theta)}\sum_{t,s}\frac{\partial L(\theta)}{\partial p_t(x_t|s_t,\theta)}\cdot p_t(x_t|s_t,\theta)\cdot\frac{\partial}{\partial\theta}\log p_t(x_t|s_t,\theta)$$
$$\overset{\text{model}}{=} \frac{1}{L(\theta)}\sum_{t,s}\left[\sum_{s_1^T:w_1^N,s_t=s} p(s_1^T|w_1^N)\cdot\frac{\prod_{\hat{t}=1}^T p_{\hat{t}}(x_{\hat{t}}|s_{\hat{t}},\theta)}{p_t(x_t|s_t,\theta)}\right]$$
$$\cdot\, p_t(x_t|s_t,\theta)\cdot\frac{\partial}{\partial\theta}\log p_t(x_t|s_t,\theta)$$
$$= \sum_{s,t} q_t(s|x_1^T,w_1^N,\theta)\cdot\frac{\partial}{\partial\theta}\log p_t(x_t|s_t,\theta)$$

with

$$q_t(s|x_1^T,w_1^N,\theta) = \frac{\sum_{s_1^T:w_1^N,s_t=s} p(x_1^T,s_1^T|w_1^N\theta)}{\sum_{s_1^T:w_1^N} p(x_1^T,s_1^T|w_1^N\theta)}.$$

The quantity $q_t(s|x_1^T,w_1^N,\theta)$ can be efficiently computed using the Baum-Welch algorithm and is also known as soft-alignment. Alternatively, one can also do the maximum approximation and calculate a Viterbi alignment and encode $q_t(s)$ as a one-hot encoding of the Viterbi alignment.

For neural-network based models the probability $p_t(x_t|s_t,\theta)$ in the conventional hybrid modeling is modeled as

$$p_t(x_t|s_t,\theta) \sim \frac{p_t(s|x_1^T,\theta)^\beta}{p(s)^\gamma},$$

where β and γ are tunable hyperparameters, and contrary to our assumptions, using RNN, actually the dependence on the full sequence x_1^T is introduced instead of the single frame x_t. For estimating q_t, the remaining factor $p(x_1^T|\theta)$ will cancel out. For the gradient, we do the approximation to assume that $p(x_1^T|\theta)$ is constant w.r.t. θ, and $p(s)$ also is not modeled by θ, and set $\beta = 1$, thus we end up with

$$\frac{\partial}{\partial\theta}\log L(\theta) = \sum_{s,t} q_t(s|x_1^T,w_1^N,\theta)\cdot\frac{\partial}{\partial\theta}\log p_t(s|x_1^T,\theta).$$

If we consider that $q_t(s)$ is fixed / pre-calculated by a previous model, this is the same gradient as for the cross entropy criterion.

In this generalized training procedure the following variations are possible:

- Baum-Welch (full-sum) vs. Viterbi.
- Realignment: per mini-batch/epoch or externally fixed.
- HMM topology, e.g. Bakis [35], one or more states, optionally with additional blank symbol.
- Full transition probabilities or simpler stationary models, and its scale α.
- State posterior probability model, e.g. a deep bidirectional LSTM, and its scale β.
- State prior probability model and its scale γ.

3.2. Training in Practice

In practice there are many possible variations on how to compute the (soft) alignment $q_t(s|x_1^T,\theta)$. Our group's standard procedure for classic systems is to steadily refine the alignment by starting with a linear alignment and then training increasingly powerful models which are then used to gradually obtain better alignments (monophone GMM $\rightarrow$ tied triphone GMM $\rightarrow$ speaker adapted GMM $\rightarrow$ framewise-trained NN).

In the full-sum training described in the previous subsection we do not necessarily need an initial alignment. Like CTC-training, this model is able to start from scratch. We start with random parameters θ and use them to compute an alignment in each mini-batch. To make the model converge to a good solution, we found it necessary to set the acoustic-model scale β initially to a small value. This will smooth-out the probability $p_t(s|x_1^T,\theta)$. Otherwise the resulting alignment will have low entropy (i.e. for each timeframe nearly all the probability mass is concentrated on one or two states), but will be wrong (i.e. not related to the evidence). We increase the value of β over time.

3.3. Relation to CTC

In this context, CTC can be seen as a special reduced HMM topology with a special blank state, no transition probabilities, no state prior probability model, trained with Baum-Welch soft alignments with realignments every mini-batch. The CTC loss L_{CTC} (2) is different but optimizing it is equivalent to the optimization of $L(\theta)$ (1) in this reformulation. In the original CTC formulation, the target sequence a_1^N over symbols from a set S is extended as $S' = S \cup \{\text{blank}\}$ and the sum is over all corresponding $s_1^T \in S'^T$ (in short we write $s_1^T : a_1^N$). In practice, the target sequence w_1^N is over words and the output labels S' are sub-units like phonemes with the additional blank symbol, thus the sum is over all $s_1^T \in S'^T$ with $s_1^T : w_1^N$. In CTC, there is the model assumption that $p(w_1^N|x_1^T)$ can be decomposed as $\sum_{s_1^T:w_1^N} p(s_1^T|x_1^T)$. Thus, in our notation, the CTC loss can be defined as:

$$\begin{aligned} L_{CTC}(\theta) &:= \log p(w_1^N|x_1^T,\hat{\theta}) \\ &= \log\sum_{s_1^T:w_1^N} p(s_1^T|x_1^T,\theta) \\ &= \log\sum_{s_1^T:w_1^N}\prod_t p_t(s_t|x_1^T,\theta) \end{aligned} \tag{2}$$

and for the gradient, we get:

$$\begin{aligned} \frac{\partial L_{CTC}}{\partial\theta} &= \frac{1}{p(w_1^N|x_1^T,\theta)}\cdot\frac{\partial p(w_1^N|x_1^T,\theta)}{\partial\theta} \\ &= \sum_{t,s}\frac{\frac{\partial p(w_1^N|x_1^T,\theta)}{\partial p_t(s|x_1^T,\theta)}}{p(w_1^N|x_1^T,\theta)}\cdot\frac{\partial}{\partial\theta}p_t(s|x_1^T,\theta) \\ &= \sum_{t,s} q_t(s|x_1^T,w_1^N,\theta)\cdot\frac{\partial}{\partial\theta}\log p_t(s|x_1^T,\theta) \end{aligned}$$

where:

$$q_t(s|x_1^T,w_1^N,\theta) = \frac{\sum_{s_1^T:w_1^N,s_t=s}\prod_{\hat{t}=1}^T p_{\hat{t}}(s_{\hat{t}}|x_1^T,w_1^N,\theta)}{\sum_{s_1^T:w_1^N}\prod_{\hat{t}} p_{\hat{t}}(s_{\hat{t}}|x_1^T,\theta)}$$

This is a special case of the full-sum HMM training as before, where the state-prior and transition model is omitted.

4. Optimal alignment properties

The CTC / full-sum training procedure implicitly converges to a well-localised soft alignment, which is optimal given the HMM-architecture and hyperparameters during training, even though no explicit alignment is provided in training. To understand the convergence behavior of the state posterior model, it is useful to study the behavior of the alignment under specific conditions.

First we look at our baseline model, which was trained frame-wise based on a good externally provided fixed Viterbi alignment.

For easier analysis, we took a short segment with only the single word ”possibly” with a single pronunciation ”p aa s ih b l iy” where each phoneme occurs once in the pronunciation. At the beginning and end there could optionally be silence. The triphone sequence is then also deterministic, as well as the corresponding sequence of generalized triphone states as obtained by CART, with optional silence at the beginning and end. For the analysis we only plot the state posterior scores for these labels in the figures. We see that the model learns the alignment quite well, with the per frame posterior mass mostly falling into a single phoneme state, each (cf. Fig. 1).

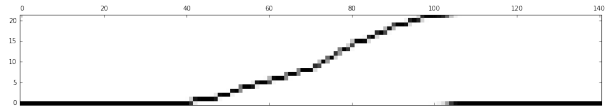

Figure 1: *BLSTM output, framewise trained based on fixed Viterbi alignment. X-axis shows time, y-axis shows states. Pixel intensity represents the value of the state-posterior. Note that silence is always state 0 in these plots.*

Without a state prior model, i.e. with a uniform distribution for the state prior model, a typical Viterbi alignment will look like in Fig. 2. Note that there are many alignments through the

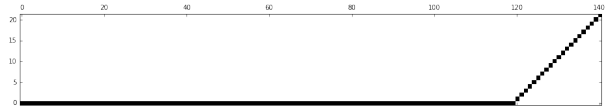

Figure 2: *Viterbi alignment based on hybrid HMM with uniform state posterior model.*

WFSA with the same probability, thus the choice of a particular alignment depends on the individual implementation.

Considering that our state posterior model is a bidirectional LSTM which has access to the full input sequence at any point in time, any monotonous target alignment would be trainable. Even a Gaussian mixture HMM does not provide actual phoneme boundaries, but with a BLSTM, nothing really enforces the model to learn an alignment reflecting anything near to actual phoneme boundaries that corresponds to the evidence in the input. The loss function simply has no way to account for that. It means that a BLSTM could just learn any monotonous alignment.

Without a state prior, the optimal score will be gained if the state posterior favors one class and provides high scores for it and the alignment in return favors also this same class. The optimal class for this behavior is one which can occur most often in the WFSA. For CTC, that is the blank, and for our conventional HMM, that is the silence. We can see that behavior in Fig. 3. This is also the behavior reported by many groups for CTC with

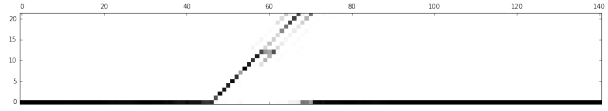

Figure 3: *BLSTM output, full-sum training without prior.*

the blank. The time when the output label occurs in the BLSTM output is not really correlated to the corresponding time frame it occurs in the audio. In [12], they even had to add a penalty to at least discourage the model from delaying the output for too long in a unidirectional LSTM.

Once the BLSTM tends to align in some way which does not correspond to the real alignment, all further training will only enforce this behavior. We argue that it is harder for a BLSTM to learn an unrelated alignment than to train it with a real (related) alignment where the input and output in one time-frame are highly related.

Also note that with a feed-forward neural network (FFNN), we enforce the model to correlate every input frame with the corresponding output, i.e. to do a proper alignment (or maybe with a fixed time-shift within the FFNN input window). That lead us to the idea to train a FFNN alongside with a BLSTM, to regularize the training to keep the alignment in the vicinity of the actual evidence within the audio data.

Note that even a FFNN without a prior will try to favor silence as much as possible as this optimizes the loss function — however, naturally the FFNN cannot learn that, although it tries, cf. Fig. 4. From personal communication, the authors of [36] report the same behavior for a convolutional FFNN. A FFNN

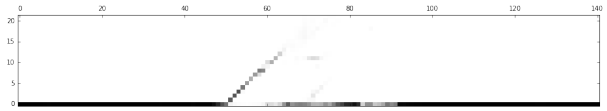

Figure 4: *FFNN output, training with full-sum without prior.*

with prior learns to align properly mostly, cf. Fig. 5.

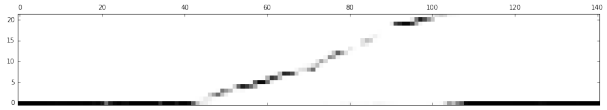

Figure 5: *FFNN output, full-sum training with prior scale 0.7.*

When we train a FFNN alongside with the BLSTM, we can train a model which is able to align more meaningfully w.r.t. the actual phoneme positions in the audio data, cf. Fig. 6. Interestingly, this was with a prior scale of 0.4 because we now had the opposite effect, that silence was scored too low by the model. You still see low silence scores in the output and the model prefers to output the first and last CART label of the word phonemes instead.

5. Baum-Welch CUDA implementation

The implementation of the Baum-Welch alignment in CUDA is not complex. For each sequence’s orthography we build a WFSA with RASR [37]. The edges of this WFSA are labeled with CART-labels and weights that are computed from a stationary transition probability. We transfer this data structure to the GPU and unfold it over time given the output of the neural network. A key ingredient to gain good performance was the usage of atomic Compare-and-Swap operations, which was inspired by [38]. The source code was published as part of RETURNN [3] and the configurations of the setups can be found at [39].

6. Experiments

6.1. Alignment results

We tested the alignment properties of this full-sum training on the Cambridge Wall Street Journal Corpus [40], as it provides manually created alignments. We trained a 4-layer BLSTM with 512 units per direction on 16 dimensional MFCC features. The prior during training was estimated iteratively using exponential decay with a decay factor of 0.9999. We start with an acoustic-model scale β of 0.10 and increase it by 0.05 each epoch until it reaches a value of 0.55. The transition-probability scale α was set to 0.25 for all results reported here. The output labels of the network were tied-triphone states, using a CART estimated using alignments generated by a monophone GMM system).

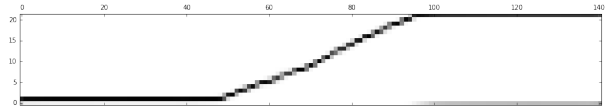

Figure 6: *BLSTM output, full-sum training with prior scale 0.4 alongside with a FFNN.*

Our system was trained using the CMU English pronunciation-lexicon [41], while the reference alignments use a different phoneme set from the beep lexicon [42]. We restricted ourselves to mapping phonemes that do not appear in the CMU lexicon to phonemes that are often hypothesized instead. Different transcriptions and pronunciation variants introduce further error. Thus these results should mainly be compared relative to each other. To compare two alignments we compute the ratio of correctly labeled audio data. For the NN-based alignments a point in time in an audio file is labeled with the label of the closest frame-center, as we use standard 25ms frames with a frame-shift of 10ms.

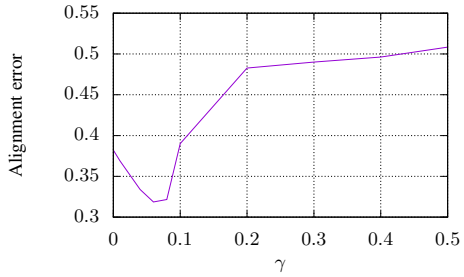

Figure 7: *Alignment error for different prior-scales γ*

Fig. 7 shows the alignment error (relative amount of time where the labels from our system do not agree with the target alignment) for different values of prior scales γ. Clearly, using a tuned state prior provides results than are closer to the human-generated alignment in comparison to not using a prior ($\gamma = 0$).

6.2. Switchboard

We use the 300h Switchboard-1 Release 2 (LDC97S62) corpus for training and the Hub5'00 evaluation data (LDC2002S09) is used for testing. We use a 4-gram language model which was trained on the transcripts of the acoustic training data (3M running words) and the transcripts of the Fisher English corpora (LDC2004T19 & LDC2005T19) with 22M running words. More details can be found in [43]. We use a 5 layer BLSTM. We have a reference Viterbi alignment from an existing tandem model. We use that Viterbi alignment for the framewise training as well as to calculate frame-error rates (FER) on a held-out cross-validation set. For the framewise training, we compare training either on full sequences or on chunks of these (see [44]). For full-sum training chunking is not straight-forward. Thus we always train on the complete feature sequence. Prior-estimation is done on-the-fly with exponential decaying average. The AM-scale β starts low at 0.05, going up to 0.3, never to 1. We jointly train a FFNN alongside with the BLSTM and use CE-smoothing for the BLSTM, i.e. we interpolate between posteriors to calculate the Baum-Welch alignment. The FFNN initially has more weight (0.8, going down to 0.5, never to 0). We do the comparison to leave out the FFNN. The results are in Table 1. First we notice that with framewise training we lose some WER performance by training on the full sequence instead of on chunks. This is a bit counterintuitive, although it probably leads to more stable training. This behavior is in line with [44].

The full-sum experiments clearly show that without the FFNN, the BLSTM has troubles in learning an optimal alignment. Also, we see that this leads to worse WER, as we expected. We notice that the FER is still off from the framewise trained system, which indicates still some instability during the training, which leads to worse WERs compared to the framewise trained system. This maybe can be overcome by more careful tuning of the AM-scale scheduling, prior estimation and other hyper parameters.

Table 1: *Results on Switchboard. We show the WER of epoch 5, 20, and the best epoch. We show the FER of the best epoch. CE training uses an alignment by a previous system, full-sum is training from scratch*

train criterion	chunking	with FFNN	WER [%] of ep. 5, 20, best total	SWB	CH	FER [%]
framewise CE	yes	no	25.4/19.0/15.4	18.8/13.3/10.3	31.9/24.7/20.4	22.5
	no	no	27.8/21.5/16.5	19.4/14.2/10.7	36.2/22.5/22.3	20.8
full-sum	no	no	91.0/24.5/19.5	89.0/16.6/12.2	93.0/32.2/26.7	84.5
	no	yes	27.5/19.3/17.5	19.0/12.2/10.8	35.9/26.2/24.2	48.8

7. Conclusion & Outlook

The analysis presented in this work allowed us to better understand some of the properties of CTC training and full-sum training in general. We have shown that CTC is a special case of full-sum HMM training and that in order to obtain alignments that more closely correspond to the evidence in the audio data a state prior is essential. Our results on the Switchboard corpus show that it is possible to train a good ASR system from scratch without alignments from previous systems, although there is still a small gap between the framewise trained system and our new system. This might be overcome by more tuning of the hyperparameters, which is ongoing work.

One point that was not addressed by this work is the tying of the triphone states by a CART. We used tyings obtained from alignments of previous systems. It would be desirable to do without state-tying in the future or at least learn it together with the alignment. We also did not perform any experiments with the original CTC topology in this work. Section 4 suggests to also use a prior in that case, which will be done in future work.

8. Acknowledgments

The research was partially supported by a Google PhD fellowship grant and has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694537). The work reflects only the authors' views and the European Research Council Executive Agency is not responsible for any use that may be made of the information it contains. It was also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract no. W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: 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 IARPA, DoD/ARL, or the U.S. Government.

9. References

[1] A. Graves, S. Fernández, F. Gomez, and J. 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] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[3] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "Returnn: the rwth extensible training framework for universal recurrent neural networks," in *ICASSP*, 2017.

[4] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 385. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24797-2

[5] S. Fernández, A. Graves, and J. Schmidhuber, "Phoneme recognition in TIMIT with BLSTM-CTC," *arXiv preprint arXiv:0804.3269*, 2008.

[6] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[7] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, T. Jebara and E. P. Xing, Eds. JMLR Workshop and Conference Proceedings, 2014, pp. 1764–1772.

[8] T. Grósz, G. Gosztolya, and L. Tóth, "A sequence training method for deep rectifier neural networks in speech recognition," in *Speech and Computer*. Springer, 2014, pp. 81–88.

[9] T. Bluche, J. Louradour, M. Knibbe, B. Moysset, M. F. Benzeghiba, and C. Kermorvant, "The A2iA Arabic handwritten text recognition system at the OpenHaRT2013 evaluation," in *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*. IEEE, 2014, pp. 161–165.

[10] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[11] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 167–174.

[12] A. Senior, H. Sak, F. de Chaumont Quitry, T. N. Sainath, and K. Rao, "Acoustic modelling with CD-CTC-sMBR LSTM RNNs," in *ASRU*, 2015.

[13] K. Hwang and W. Sung, "Sequence to sequence training of ctc-rnns with partial windowing," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 2178–2187.

[14] H. Sak, A. Senior, K. Rao, and F. Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4280–4284.

[16] H. K. Naoyuki Kanda, Xugang Lu, "Minimum bayes risk training of ctc acoustic models in maximum a posteriori based decoding framework," in *ICASSP*, 2017.

[17] G. Pundak and T. Sainath, "Lower frame rate neural network acoustic models," in *INTERSPEECH*, 2016.

[18] Y. Miao, M. Gowayyed, X. Na, T. Ko, F. Metze, and A. Waibel, "An empirical exploration of CTC acoustic models," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2623–2627.

[19] J. Li, H. Zhang, X. Cai, and B. 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*, 2015.

[20] P. Haffner, "Connectionist speech recognition with a global mmi algorithm," in *EUROSPEECH*, 1993.

[21] A. Senior and T. Robinson, "Forward-backward retraining of recurrent neural networks," *Advances in Neural Information Processing Systems*, pp. 743–749, 1996.

[22] J. Hennebert, C. Ris, H. Bourlard, S. Renals, and N. Morgan, "Estimation of global posteriors and forward-backward training of hybrid hmm/ann systems." 1997.

[23] Y. Yan, M. Fanty, and R. Cole, "Speech recognition using neural networks with forward-backward probability generated targets," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 4. IEEE Computer Society, 1997, pp. 3241–3241.

[24] X. Li and X. Wu, "Labeling unsegmented sequence data with dnn-hmm and its application for speech recognition," in *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, 2014, pp. 10–14.

[25] C. Zhang and P. Woodland, "Standalone training of context-dependent deep neural network acoustic models," in *ICASSP'14*, 2013.

[26] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, "GMM-free DNN acoustic model training," in *ICASSP'14*, 2014.

[27] M. Bacchiani, A. W. Senior, and G. Heigold, "Asynchronous, online, GMM-free training of a context dependent acoustic model for speech recognition," in *INTERSPEECH*, 2014, pp. 1900–1904.

[28] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[29] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *arXiv preprint arXiv:1508.04395*, 2015.

[30] L. Lu, L. Kong, C. Dyer, N. A. Smith, and S. Renals, "Segmental recurrent neural networks for end-to-end speech recognition," 2016.

[31] L. Kong, C. Dyer, and N. A. Smith, "Segmental recurrent neural networks," *arXiv preprint arXiv:1511.06018*, 2015.

[32] D. Y. O. Abdel-Hamid, L. Deng and H. Jiang, "Deep segmental neural network for automatic speech recognition," in *Proc. of Interspeech, Lyon, France*, 2013.

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

[34] H. K. Naoyuki Kanda, Xugang Lu, "Maximum a posteriori based decoding for ctc acoustic models," 2016.

[35] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *The Journal of the Acoustical Society of America*, vol. 59, no. S1, pp. S97–S97, 1976.

[36] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Laurent, Y. Bengio, and A. Courville, "Towards end-to-end speech recognition with deep convolutional neural networks," 2016.

[37] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014, pp. 3313–3317.

[38] J. Kim and I. Lane, "Accelerating large vocabulary continuous speech recognition on heterogeneous cpu-gpu platforms," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 3291–3295.

[39] "GitHub repository with config files for experiments in RETURNN," 2016. [Online]. Available: https://github.com/rwth-i6/returnn-experiments/tree/master/2016-ctc-paper

[40] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "Wsjcam0: A british english speech corpus for large vocabulary continuous speech recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*. IEEE, 1995, pp. 81–84.

[41] "Svn repository of the cmu pronouncing dictionary," 2016. [Online]. Available: http://svn.code.sf.net/p/cmusphinx/code/branches/cmudict-new/

[42] "Beep dictionary," 1996. [Online]. Available: http://svr-www.eng.cam.ac.uk/comp.speech/Section1/Lexical/beep.html

[43] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Speaker adaptive joint training of gaussian mixture models and bottleneck features," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Scottsdale, AZ, USA, Dec. 2015, pp. 596–603.

[44] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, "A comprehensive study of deep bidirectional lstm rnns for acoustic modeling in speech recognition," in *ICASSP*, 2017.