Sequence to Sequence Training of CTC-RNNs with Partial Windowing – Supplementary Material –

Kyuyeon Hwang KYUYEON.HWANG@GMAIL.COM

Wonyong Sung WYSUNG@SNU.AC.KR

Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, 08826 Korea

Appendices

A. Derivation of the CTC-EM Equations

In the maximization step, the objective is to obtain the derivative of $Q_\tau(\mathbf{w}|\mathbf{x},\mathbf{z},\mathbf{w}^{(n)})$ with respect to the input of the softmax layer, a_k^t, at time t. We first differentiate Q_τ with respect to y_k^t at $\mathbf{w}=\mathbf{w}^{(n)}$:

$$\left.\frac{\partial Q_\tau(\mathbf{w}|\mathbf{x},\mathbf{z},\mathbf{w}^{(n)})}{\partial y_k^t}\right|_{\mathbf{w}=\mathbf{w}^{(n)}} = \sum_{m=0}^{|\mathbf{z}|} p(\mathbf{z}_{1:m}|Z,\mathbf{x}_{1:\tau},\mathbf{w}^{(n)}) \frac{\partial \ln p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}{\partial y_k^t}. \tag{1}$$

With Bayes' rule, we obtain

$$p(\mathbf{z}_{1:m}|Z,\mathbf{x}_{1:\tau},\mathbf{w}^{(n)}) = \frac{p(\mathbf{z}_{1:m},Z|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}{p(Z|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})} = \frac{p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}{p(Z|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}, \tag{2}$$

and with simple calculus,

$$\frac{\partial \ln p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}{\partial y_k^t} = \frac{1}{p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})} \frac{\partial p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}{\partial y_k^t}. \tag{3}$$

Then, (1) becomes

$$\left.\frac{\partial Q_\tau(\mathbf{w}|\mathbf{x},\mathbf{z},\mathbf{w}^{(n)})}{\partial y_k^t}\right|_{\mathbf{w}=\mathbf{w}^{(n)}} = \frac{1}{p(Z|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})} \sum_{m=0}^{|\mathbf{z}|} \frac{\partial p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}{\partial y_k^t} \tag{4}$$

$$= \frac{1}{p(Z|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})} \frac{\partial p(Z|\mathbf{x}_{1:\tau},\mathbf{w}^{(n)})}{\partial y_k^t}. \tag{5}$$

If we define the loss function to be minimized as

$$\mathcal{L}_\tau(\mathbf{x},\mathbf{z}) = -\ln p(Z|\mathbf{x}_{1:\tau}), \tag{6}$$

then its derivative equals to (5) with the opposite sign:

$$\frac{\partial \mathcal{L}_\tau(\mathbf{x},\mathbf{z})}{\partial y_k^t} = -\left.\frac{\partial Q_\tau(\mathbf{w}|\mathbf{x},\mathbf{z},\mathbf{w}^{(n)})}{\partial y_k^t}\right|_{\mathbf{w}=\mathbf{w}^{(n)}}. \tag{7}$$

From now on, we drop $\mathbf{w}^{(n)}$ without loss of generality. Let

$$\beta_{\tau,m}(\tau,u) = \begin{cases} 1 & \text{if } u = 2m, 2m+1 \\ 0 & \text{otherwise} \end{cases}. \tag{8}$$

Following the standard CTC forward-backward equations in Graves et al. (2012),

$$p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau}) = \sum_{u=1}^{|\mathbf{z}'|} \alpha(t,u)\beta_{\tau,m}(t,u) \tag{9}$$

$$\frac{\partial p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau})}{\partial y_k^t} = \frac{1}{y_k^t} \sum_{u \in B(\mathbf{z},k)} \alpha(t,u)\beta_{\tau,m}(t,u). \tag{10}$$

From (9) and (10), $p(Z|\mathbf{x}_{1:\tau})$ and its derivative become

$$p(Z|\mathbf{x}_{1:\tau}) = \sum_{m=0}^{|\mathbf{z}|} p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau}) = \sum_{u=1}^{|\mathbf{z}'|} \alpha(t,u)\beta_{\tau}(t,u) \tag{11}$$

$$\frac{\partial p(Z|\mathbf{x}_{1:\tau})}{\partial y_k^t} = \sum_{m=0}^{|\mathbf{z}|} \frac{\partial p(\mathbf{z}_{1:m}|\mathbf{x}_{1:\tau})}{\partial y_k^t} = \frac{1}{y_k^t} \sum_{u \in B(\mathbf{z},k)} \alpha(t,u)\beta_{\tau}(t,u), \tag{12}$$

where the new backward variable for $p(Z|\mathbf{x}_{1:\tau})$ is

$$\beta_{\tau}(t,u) = \sum_{m=0}^{|\mathbf{z}|} \beta_{\tau,m}(t,u), \tag{13}$$

which results in the simple initialization as

$$\beta_{\tau}(\tau,u) = 1, \ \forall u. \tag{14}$$

Then, the error gradients become

$$\frac{\partial \mathcal{L}_{\tau}(\mathbf{x},\mathbf{z})}{\partial y_k^t} = -\frac{1}{p(Z|\mathbf{x}_{1:\tau})} \frac{1}{y_k^t} \sum_{u \in B(\mathbf{z},k)} \alpha(t,u)\beta_{\tau}(t,u) \tag{15}$$

$$\frac{\partial \mathcal{L}_{\tau}(\mathbf{x},\mathbf{z})}{\partial a_k^t} = y_k^t - \frac{1}{p(Z|\mathbf{x}_{1:\tau})} \sum_{u \in B(\mathbf{z},k)} \alpha(t,u)\beta_{\tau}(t,u), \tag{16}$$

where

$$p(Z|\mathbf{x}_{1:\tau}) = \sum_{u=1}^{|\mathbf{z}'|} \alpha(\tau,u). \tag{17}$$

B. Phoneme Recognition on TIMIT

B.1. TIMIT Corpus

The TIMIT corpus (Garofolo et al., 1993) contains American English recordings of 630 speakers from 8 major dialect regions in the United States. The training set contains about 3.1 hours of 3,696 utterances from 462 speakers after removing the SA recordings, in which only two sentences are spoken by multiple speakers. Figure 1 shows the histogram of the length of the training sequences, where the feature frames are extracted with the 10 ms period. The average length of the training sequences is 304 frames. We use the *core test* set with 192 utterances as the test set. The development set contains the remaining 1,152 utterances that are obtained by excluding the *core test* set from the *complete test* set. The corpus also includes the full phonetic transcriptions.

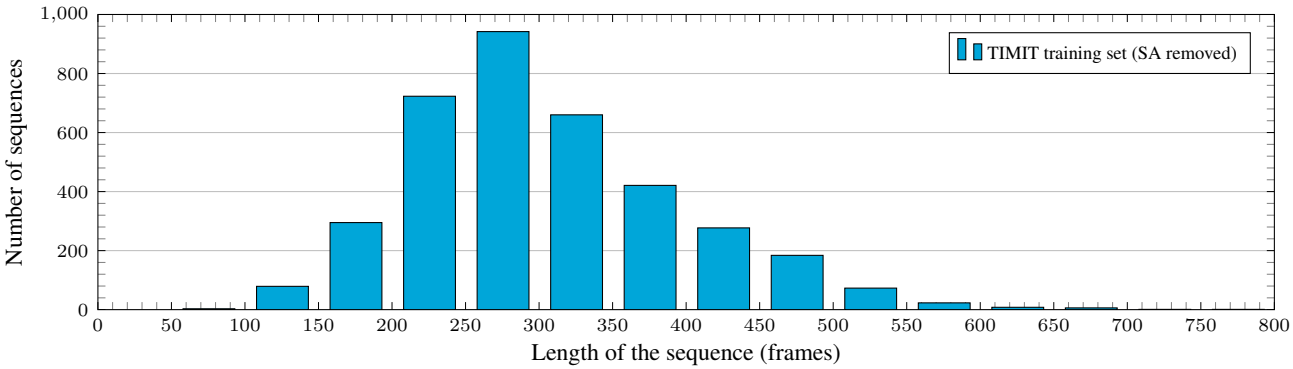

Figure 1. Histogram of the length of the sequences in the TIMIT training set (SA removed), where the feature frames are extracted with the period of 10 ms.

B.2. Network Structure

The network structure is a deep unidirectional LSTM RNN with 3 LSTM hidden layers, where each LSTM layer has 512 cells. The input is the same log Mel-frequency filterbank feature as in the WSJ experiments. The training procedure is also similar. The original TIMIT transcriptions are based on 61 phonetic labels. Accordingly, the RNN output is a 62-dimensional vector that consists of the probabilities of the original 61 phonemes and the extra CTC label. However, after decoding, they are mapped to 39 phonemes for evaluation as in Lee & Hon (1989).

B.3. Training Procedure

For the experiments, the continuous CTC($2h'$; h') algorithm is employed so that the resulting RNN can run continuously on a infinitely long stream of the input speech. The networks are pre-trained with ADADELTA (Zeiler, 2012), where the local learning rates are adaptively adjusted using the statistics of the recent gradient values. Before the online CTC training with the unroll amount greater than of equal to 512, the pre-training is performed for the 8 M (8×2^{20}) training frames with the unroll amount of 2,048, the learning rate of 10^{-5}, the Nesterov momentum of 0.9, and the RMS decay rate of 0.99 for ADADELTA. On the other hand, we pre-trained the network with 12 M frames for the subsequent CTC training with less than 512 unroll steps. Unlike in the WSJ experiments, it is observed that applying the standard SGD method at the beginning often fails to initiate the training. We consider this is because the gradient computed by the SGD method is initially not noisy enough to help the parameters escape from the initial saddle point.

After the pre-training, the standard SGD is applied with the Nesterov momentum of 0.9. The training starts with the learning rate of 10^{-4}. The intermediate evaluations are performed at every 2 M (2×2^{20}) training frames on the development set with the best path decoding. If the phoneme error rate (PER) fails to improve during 6 consecutive evaluations, the learning rate decreases by the factor of 2 and the parameters are restored to those of the second best network. The training finishes when the learning rate becomes less than 10^{-6}.

The network is regularized with dropout (Hinton et al., 2012) in both the pre-training and the main training stages following the approach in Zaremba et al. (2014), that is, dropout is only applied on the non-recurrent connections. The dropout rate is fixed to 0.5 throughout the experiments.

B.4. Evaluation

The networks are evaluated on the very long test stream that is obtained by concatenating the entire test sequences. For the evaluation, the network output is decoded by the CTC beam search. The experiments are repeated 4 times and the mean and standard deviation estimates of PERs are reported based on the reduced 39-phoneme set.

The RNNs are unrolled 64, 128, 256, 512, 1,024, and 2,048 times. As shown in Table 1, the various unroll amounts make little difference to the final PERs on the test set. When the RNN is unrolled only 128 times, which is less than the average length of training sequences, the best PER of $20.73 \pm 0.40\%$ is obtained. On the other hand, the training with the unroll amount of 2,048 results in slightly degraded performance since it becomes harder for RNNs to catch the dependencies

Table 1. Comparison of CTC-TR coverages and PERs on the test set after CTC($2h'$; h') training with the varying amounts of unrolling.

# Streams	**CTC-TR coverage (%)**		**PER (%)**			
× # Unroll	Average	Maximum	Mean	± Stdev.	Min.	Max.
8 × 2,048	100.0	100.0	21.14	± 0.29	20.91	21.57
16 × 1,024	99.80	100.0	20.82	± 0.17	20.66	21.03
32 × 512	89.48	99.60	21.18	± 0.40	20.60	21.48
64 × 256	60.69	79.37	20.77	± 0.24	20.47	20.97
128 × 128	31.53	42.02	**20.73**	± 0.40	**20.39**	21.25
256 × 64	15.77	21.03	21.00	± 0.16	20.78	21.15

Table 2. Comparison of the proposed online CTC algorithm and the other models in the literature in terms of PER on the test set.

Model	**Network (# param)**	**Bi-**	**Test sequence**	**PER (%)**
Proposed online CTC	LSTM (5.5 M)	**No**	**Almost infinite stream** [a]	20.73
Attention-based model [b]	Conv.[c]+GRU[d]	**Yes**	**Long sequences** [e]	About 20
			Utterance-wise	17.6
RNN transducer [f]	LSTM (4.3 M)	Yes	Utterance-wise	17.7
Sequence-wise CTC [f]	LSTM (3.8 M)	Yes	Utterance-wise	18.4
		No		19.6

[a] Generated by concatenating all of the 192 test utterances
[b] Chorowski et al. (2015)
[c] Convolutional features
[d] Gated recurrent unit (Cho et al., 2014)
[e] Generated by concatenating 11 utterances
[f] Graves et al. (2013)

between the input and output sequences due to the noisy input frames from the consecutive sequences.

The performance of the proposed online CTC algorithm is compared with the other models in Table 2. The other models employ early stopping to prevent overfitting and add weight noise while training for regularization. The bidirectional attention-based model in Chorowski et al. (2015) shows 17.6% PER with utterance-wise decoding. However, the PER increases to about 20% with the long test sequences that are generated by concatenating 11 utterances. On the other hand, our CTC(128; 64)-trained unidirectional RNNs show 20.73±0.40% PER with a very long test stream that is made by concatenating the entire 192 test utterances. Note that, unlike the CTC-trained unidirectional RNNs, the bidirectional models require unrolling in test time and have to listen the entire speech before generating outputs. Therefore, the proposed unidirectional RNN models are more suitable for realtime low-latency speech recognition systems without sacrificing much performance.

References

Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chorowski, Jan, Bahdanau, Dzmitry, Serdyuk, Dmitriy, Cho, Kyunghyun, and Bengio, Yoshua. Attention-based models for speech recognition. *arXiv preprint arXiv:1506.07503*, 2015.

Garofolo, John S, Lamel, Lori F, Fisher, William M, Fiscus, Jonathon G, and Pallett, David S. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, 93:27403, 1993.

Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6645–6649. IEEE, 2013.

Graves, Alex et al. *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer, 2012.

Hinton, Geoffrey E, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Lee, Kai-Fu and Hon, Hsiao-Wuen. Speaker-independent phone recognition using hidden Markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(11):1641–1648, 1989.

Zaremba, Wojciech, Sutskever, Ilya, and Vinyals, Oriol. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Zeiler, Matthew D. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.