DROPOUT APPROACHES FOR LSTM BASED SPEECH RECOGNITION SYSTEMS

Jayadev Billa[*†]

Information Sciences Institute,
University of Southern California, Marina del Rey, CA 90292, USA
jbilla@isi.edu

ABSTRACT

In this paper we examine dropout approaches in a Long Short Term Memory (LSTM) based automatic speech recognition (ASR) system trained with the Connectionist Temporal Classification (CTC) loss function. In particular, using an Eesen based LSTM-CTC speech recognition system, we present dropout implementations that result in significant improvements in speech recognizer performance on Librispeech and GALE Arabic datasets, with 24.64% and 13.75% relative reduction in word error rates (WER) from their respective baselines.

Index Terms— LSTM, dropout, speech recognition

1. INTRODUCTION

Dropout is a very effective regularization technique in neural network training wherein a random subset of neural activations is set to zero, i.e., *masked*, at each training iteration. The general approach to apply dropout is well established for feedforward networks [1], however, application to recurrent neural networks has seen a number of variants [2, 3, 4, 5]. In speech recognition, dropout in feedforward networks has been explored extensively, e.g., [6, 7, 8, 9, 10]. That said, application of dropout in recurrent neural network based ASR systems had been limited, [3] reported results on Wall Street Journal (WSJ) corpus; only recently, in efforts parallel to and independent of our own, has dropout been applied in recurrent neural network based ASR systems across large vocabulary speech recognition problems [11]. In this paper we present our work on extending the simple single factor dropout mask formulation to the speech recognition task in an LSTM-CTC based system.

This paper starts with a brief overview of our baseline LSTM-CTC system. We follow in Section 3 with quick review of prior work, followed by our work and results on feedforward dropout and then our efforts with recurrent dropout. In Section 4 we explore dropout combination approaches and present results before we summarize and conclude this paper.

2. BASELINE LSTM-CTC SYSTEM

Our baseline system is based on the publicly available LSTM-CTC based Eesen Toolkit [12]. The LSTM model consists of layers of LSTM cells that receive input from the previous layer or model input, the feedforward connection, as well as the cell's immediate past or future. Each LSTM cell consists of three gates: input, forget, and output, which control input signal flow and memory retention. The vector formulas that describe the LSTM cell are

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{R}_i\mathbf{h}_{t-1} + \mathbf{P}_i\mathbf{c}_{t-1} + \mathbf{b}_i) \quad (1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{R}_f\mathbf{h}_{t-1} + \mathbf{P}_f\mathbf{c}_{t-1} + \mathbf{b}_f) \quad (2)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \phi(\mathbf{W}_c\mathbf{x}_t + \mathbf{R}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{R}_o\mathbf{h}_{t-1} + \mathbf{P}_o\mathbf{c}_t + \mathbf{b}_o) \quad (4)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \phi(\mathbf{c}_t) \quad (5)$$

where $\mathbf{x}_t$ is the input vector at time t, $\mathbf{W}$ are rectangular input weight matrices connecting inputs to the LSTM cell, $\mathbf{R}$ are square recurrent weight matrices connecting the previous memory cell state to the LSTM cell, $\mathbf{P}$ are diagonal peephole weight matrices and $\mathbf{b}$ are bias vectors. Functions σ and ϕ are the *logistic sigmoid* and $\tanh$ nonlinearities respectively. $\odot$ represents the point-wise multiplication operator.

The LSTM model in our experiments consists of 4 bidirectional stacked layers of 640 LSTM cells (320 in each direction). Details of our acoustic training, language modeling, and test sets on Librispeech is in [13]. The Arabic system uses GALE Phase 2 Broadcast Conversation corpora (audio: LDC2013S02, LDC2013S027; transcripts: LDC2013T17, LDC2013T04) consisting of ~250hrs of acoustic data to build a grapheme based system. We follow the Kaldi[14] GALE Arabic `s5b` recipe[1] for data preparation, grapheme set, dictionary and language model.The Librispeech system is phoneme based whereas the GALE Arabic system is a grapheme based system. Table 1 summarizes the baseline WER on these corpora.

[*]This paper is based largely on research conducted by the author as an unaffiliated researcher.

[†]GALE Arabic results are based upon work supported by the United States Air Force Research Laboratory (AFRL) under contract FA8650-16-C-6697. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the United States Air Force.

[1]https://github.com/kaldi-asr/kaldi/tree/master/egs/gale_arabic/s5b

Table 1. Librispeech and GALE Arabic baseline.

System	%WER
Librispeech 100hr (dev-clean)	9.78
GALE Arabic 250hr (withheld test)	22.84

3. DROPOUT IN RECURRENT MODELS

Initial implementations of dropout for recurrent neural networks focused on application of dropout on the feedforward connections only [2, 15, 16]. Subsequently, Moon et. al. [3] introduced RNNDrop, in which dropout is applied to LSTM cell memory at a sequence level, i.e., the dropout mask is kept fixed across each training sequence, or in the case of speech, each utterance. Variational RNN [5] is similar, but dropout is applied to the LSTM cell output recurrent connection as well as the forward non-recurrent connections, also at a sequence level. An interesting variant is recurrent dropout without memory loss [4], where dropout is applied to the LSTM cell memory update, which prevents the LSTM cell memory from being reset as is the case with RNNDrop. More recently, [11] explores dropout in projected LSTM based ASR systems. Given the slightly different architecture of the projected LSTM cell, they investigate different dropout location approaches and further incorporate a schedule driven dropout rate, where the dropout factor changes over time, and show relative WER reductions, on the order of 3% to 7%, on a variety of data sets.

In our experiments, detailed below, we explore dropout on feedforward and recurrent connections separately as well as together. To maintain a common operating point in our experiments, we fix the dropout rate to 0.2 for forward and recurrent connections as applicable. It is likely that a more comprehensive exploration of dropout rates will provide additional performance improvement.

3.1. Dropout on feedforward connections

Dropout on feedforward connections in LSTM networks is closely aligned with the original formulation of dropout where the composite LSTM cell is the unit to be dropped. In [15] and [2], the argument is made to apply dropout only on the feedforward connections and not the recurrent connections so as to minimize impact on the sequence modeling capability of a recurrent network. In the implementation described in [2], dropout is applied every time step to the feedforward connections. However, given the within layer recurrence, application of dropout at every time step would be a noisy implementation of the classical dropout approach, i.e., each sampling of the networks would receive input from another sampling of the network via the recurrent connections. We argue that a more faithful implementation for recurrent networks would be to retain the dropout mask across a complete sequence/utterance for the forward non-recurrent connections to eliminate this cross-sampling noise. In this instance, each sequence is trained on a particular sampling of the network, rather than a multitude of sampled networks driving a noisy recurrence. We note that this is identical to the dropout implementation in [5] for forward connections. Figure 1 illustrates the forward dropout implementation in our experiments.

Table 2. Experiments with dropout on feedforward connections on Librispeech. Postfix step and seq indicate whether the dropout mask is sampled every time step or every sequence. RI refers to relative improvement over baseline.

Librispeech System	%WER	%RI
Librispeech baseline	9.78	-
Forward-step	9.51	2.76
Forward-seq	10.03	−2.56

Table 3. Experiments with dropout on feedforward connections with stacked/strided (SS) features.

Librispeech System	%WER	%RI
Librispeech w/ SS feats	10.72	−9.61
Forward-step	9.17	6.24
Forward-seq	8.63	11.76

To validate our thinking, we experimented with both time step and sequence forward dropout variants. Table 2 details our results. We expected a performance improvement with dropout mask sampled every time step and a larger improvement when the dropout mask is sampled every sequence. However, while we see an improvement with dropout mask sampling every time step, when the dropout mask is sampled every sequence, performance is worse than without dropout. Inspection of the training logs showed that this model started to overfit early, driving the lower performance.

In other experiments detailed in [13], we observed a slight but better relative WER improvement with 9-fold max perturbation, our data augmentation protocol, using stacked and strided features. We proceeded to investigate if dropout would also show a similar magnified effect when coupled with stacked and strided features. For the stacked and strided feature we use the current frame and ± 1 feature frames, for a composite stacked feature of 3 frames, and a stride of 3, for an effective 30ms frame rate vs. the original 10ms frame rate. We applied the same forward dropout variants as in Table 2 with stacked and strided features; Table 3 details these results. Per time step dropout mask sampling with stack and stride features yields a 14.5% relative WER improvement over the corresponding baseline, and per sequence dropout mask sampling yields 19.5% relative WER improvement. To be conservative, compared to the base feature system with 9.78% WER, we still see a relative WER improvement of 6.2% and 11.8% respectively for the time step and sequence variants.

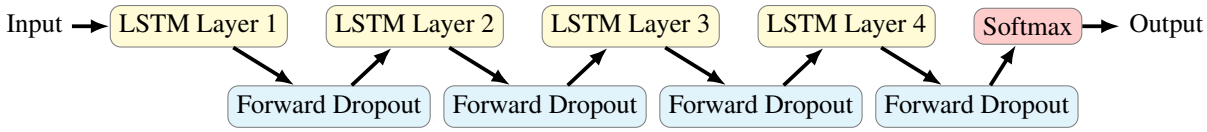

Fig. 1. Forward dropout as implemented in our experiments. The dropout mask can be sampled either every time step or every sequence. In the latter, the dropout mask is fixed for all time steps in any specific sequence.

During training we found that networks trained on stacked and strided features were able to train for many more epochs without overfitting when coupled with dropout. Given this strong outperformance we shifted to using stacked and strided features for all subsequent experiments.

3.2. Dropout on recurrent connections

In exploring dropout for recurrent connections, we considered two approaches to recurrent connection dropout, RNNDrop [3] and recurrent dropout without memory loss [4]. In RNNDrop, dropout is applied to memory cell content, in particular, Equation 3 which describes the memory cell, changes as below:

$$\begin{aligned}\mathbf{c}_t = \mathbb{m}_t \odot \big(&\mathbf{f}_t \odot \mathbf{c}_{t-1} \\ &+ \mathbf{i}_t \odot \phi(\mathbf{W}_c\mathbf{x}_t + \mathbf{R}_c\mathbf{h}_{t-1} + \mathbf{b}_c)\big)\end{aligned} \tag{6}$$

where $\mathbb{m}_t$ represents the dropout mask at time t.

In the case of recurrent dropout without memory loss, dropout is only applied to the incremental memory cell update, and Equation 3 changes as below:

$$\begin{aligned}\mathbf{c}_t =&\mathbf{f}_t \odot \mathbf{c}_{t-1} \\ &+ \mathbb{m}_t \odot \mathbf{i}_t \odot \phi(\mathbf{W}_c\mathbf{x}_t + \mathbf{R}_c\mathbf{h}_{t-1} + \mathbf{b}_c)\end{aligned} \tag{7}$$

where again $\mathbb{m}_t$ represents the dropout mask at time t.

We expect recurrent dropout without memory loss to show better performance vis-à-vis RNNDrop since the cell memory is not being continually reset as is the case with RNNDrop.

Table 4 details our results, where we have abbreviated recurrent dropout without memory loss as no memory loss (NML) dropout. We exclude the RNNDrop-seq model since it suffered from an exploding memory cell value problem, an issue predicted in [4]. In line with our expectations, NML dropout worked better than RNNDrop, with the per sequence dropout mask variant slightly edging out the per time step dropout mask variant, inline with trends reported in [4] albeit on different tasks. For the NML dropout variants, we see relative WER reductions over 20% compared to the corresponding baseline, or ~13% relative WER reduction compared to the prior best system without dropout WER of 9.78%.

In the RNNDrop-step model, with a dropout rate of 0.2, we effectively reset ~99%[2] of all LSTM cells within 20 time steps, i.e., each LSTM cell retains at most 600ms of memory with our 30ms frame rate. An interesting corollary of this observation is that we can consider refactoring a bidirectional LSTM with RNNDrop system as an unidirectional LSTM system with suitably delayed output, with similar or near similar performance and lower latency.

[2] $1 - 0.8^{20}$

Table 4. Experiments with dropout on recurrent connections.

Librispeech System	%WER	%RI
Librispeech w/ SS feats	10.72	-9.61
RNNDrop-step	9.07	7.26
NML-step	8.55	12.58
NML-seq	8.45	13.60

Table 5. Experiments with naïve combination.

Librispeech System	%WER	%RI
RNNDrop-step + Forward-step	8.60	12.07
RNNDrop-step + Forward-seq	8.85	9.51
NML-step + Forward-step	8.08	17.38
NML-step + Forward-seq	7.76	20.65
NML-seq + Forward-step	7.72	21.06
NML-seq + Forward-seq	7.97	18.51

4. COMBINING FEEDFORWARD AND RECURRENT DROPOUT

4.1. Naïve dropout combination

The simplest approach to combine dropout is to apply both forward and recurrent dropout concurrently during network training; we refer to this combined dropout approach as *naïve* dropout combination. Indeed this is the approach that has been traditionally taken while combining dropout [5, 4].

To be exhaustive we ran experiments with all combinations of the three recurrent dropout variants and the two forward dropout variants, albeit with the same fixed dropout rate of 0.2. Table 5 summarizes these models and their corresponding WERs. The RNNDrop combination variants, while showing better WER performance than the RNNDrop alone system, are worse, or at best similar, in performance to the forward dropout alone models. On the other hand, the NML combination variants all show performance improvements over the NML or forward dropout alone models.

Table 6. Comparison to hybrid DNN systems.

System	%WER
Librispeech 100hr best (this paper)	7.37
Kaldi (p-norm DNN, LDA+MLLT+SAT, 100hrs training)†	7.91
Kaldi (p-norm DNN, LDA+MLLT+SAT, 460hrs training)†	7.16
GALE Arabic 250hr best (this paper)	19.70
Kaldi (TDNN chain model, MMI, 415hrs training)‡	20.26
Kaldi (TDNN/LSTM chain model, MMI, 415hrs training)‡	17.64

† See Kaldi GitHub repo under `egs/librispeech/s5/RESULTS` (retrieved 10/26/2017).
‡ See Kaldi GitHub repo under `egs/gale_arabic/s5b/RESULTS` (retrieved 10/26/2017).

4.2. Stochastic and cascade dropout combination

During our experimentation, it was clear that different dropout approaches had very different training profiles. One conjecture is that the different dropout approaches impact regularization in different ways. If so, we can direct this regularization more deliberately than a naïve dropout combination.

One combination approach is to apply forward or recurrent dropout singly rather than concurrently. The exact approach we implemented is that for each minibatch, we pick from an equiprobable Bernoulli distribution to decide between forward or recurrent dropout, and then apply the appropriate dropout for that minibatch. Note that there is nothing special in our choice of distribution or decision choice, an equally valid implementation could bias towards a particular dropout or introduce an additional decision choice to pick between forward and/or recurrent dropout types. We refer to this general approach of distribution based choice to determine dropout combination as *stochastic* dropout combination.

Another combination approach is to train the model with one type, combination or parameterization of dropout, and then switch to a different type of dropout, combination or parameterization, at an opportune time. Given that we cascade different dropout combinations during training we refer to this general approach as *cascade* dropout combination. The dropout schedule approach described in Cheng et. al. [11] is a specific example of cascade dropout combination. We should note that cascade and stochastic dropout combination are orthogonal approaches and can be applied at the same time.

Given our limited compute resources, we were unable to systematically explore the many permutations of dropout combination in a reasonable time frame. Table 7 summarizes our experiments with stochastic dropout combination on Librispeech and GALE Arabic. We find that while not all stochastic dropout combination results show better performance over naïve dropout combination, in the case of sequence based NML/Forward dropout combination we see an additional 6.6% relative reduction in WER over naïve dropout combination. For GALE Arabic, a grapheme based ASR system, we see a large 13.75% improvement in WER with our best stochastic combination approach. Table 8 illustrates one example of cascade dropout combination where we see a 4.5% relative improvement in WER over the individual systems alone.

Table 7. Experiments with stochastic combination.

Librispeech System	%WER	%RI
NML-step + Forward-step	8.76	10.43
NML-step + Forward-seq	8.02	18.00
NML-seq + Forward-step	7.86	19.63
NML-seq + Forward-seq	7.44	23.93
GALE Arabic baseline	22.84	-
+ SS feats + NML-seq + Forward-seq	19.70	13.75

Table 8. Experiments with cascade combination.

Librispeech System	%WER	%RI
NML-seq + Forward-step (1)	7.72	21.06
NML-seq + Forward-seq (2)	7.97	18.51
Cascade (1) $\rightsquigarrow$ (2)	7.37	24.64

Note that with these results we have demonstrated, for the first time, LSTM-CTC performance to be equivalent or better than similarly trained hybrid DNN systems on smaller (100-300hr) corpora. Table 6 compares the best systems in this paper to equivalent systems built with the Kaldi toolkit [14]. We note that for both Librispeech and GALE Arabic, our best systems with dropout are comparable or very close in WER performance to systems trained with much more data, 460hr vs 100hr on Librispeech, 415hr vs 250hr on GALE Arabic.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we have presented dropout approaches that significantly improve LSTM-CTC ASR system performance, across languages (English vs. Arabic) and system type (phoneme vs. grapheme). Dropout implementations and Librispeech recipes have been merged into the public Eesen GitHub repository[3].

[3] `https://github.com/srvk/eesen`

6. REFERENCES

[1] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[2] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014.

[3] Taesup Moon, Heeyoul Choi, Hoshik Lee, and Inchul Song, "RNNDROP: A novel dropout for RNNs in ASR," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015*, pp. 65–70.

[4] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth, "Recurrent dropout without memory loss," in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pp. 1757–1766.

[5] Yarin Gal and Zoubin Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, December 5-10, 2016, Barcelona, Spain*, pp. 1019–1027.

[6] George E. Dahl, Tara N. Sainath, and Geoffrey E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013, pp. 8609–8613.

[7] Jie Li, Xiaorui Wang, and Bo Xu, "Understanding the dropout strategy and analyzing its effectiveness on LVCSR," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013, pp. 7614–7618.

[8] Shiliang Zhang, Yebo Bao, Pan Zhou, Hui Jiang, and Li-Rong Dai, "Improving deep neural networks for LVCSR using dropout and shrinking structure," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*. 2014, pp. 6849–6853, IEEE.

[9] Steven J. Rennie, Vaibhava Goel, and Samuel Thomas, "Annealed dropout training of deep networks," in *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*. 2014, pp. 159–164, IEEE.

[10] Steven J. Rennie, Pierre L. Dognin, Xiaodong Cui, and Vaibhava Goel, "Annealed dropout trained maxout networks for improved LVCSR," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*. 2015, pp. 5181–5185, IEEE.

[11] Gaofeng Cheng, Vijayaditya Peddinti, Daniel Povey, Vimal Manohar, Sanjeev Khudanpur, and Yonghong Yan, "An exploration of dropout with LSTMs," in *INTERSPEECH 2017, Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*.

[12] Yajie Miao, Mohammad Gowayyed, and Florian 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 2015, Scottsdale, AZ, USA, December 13-17, 2015*, pp. 167–174.

[13] Jayadev Billa, "Improving LSTM-CTC based ASR performance in domains with limited training data," *CoRR*, vol. abs/1707.00722, 2017.

[14] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[15] Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, Crete, Greece, September 1-4, 2014*, pp. 285–290.

[16] Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567.