Recurrent neural network regularization

Wojciech Zaremba WOJ.ZAREMBA@GMAIL.COM
Google & New York University
Ilya Sutskever ILYASU@GOOGLE.COM
Google
Oriol Vinyals VINYALS@GOOGLE.COM
Google

Abstract

We present a simple regularization technique for Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) units. The technique is based on dropout and gives a tremendous reduction in overfitting. We show that it is useful in a variety of sequence modeling problems that include language modeling, speech recognition, and machine translation.

1. Introduction

RNNs yield the state of the art performance on many sequence modeling tasks, including language modeling (Mikolov, 2012), speech recognition (Graves et al., 2013), and machine translation (Cho et al., 2014). However, there have been no good ways of regularizing them. As a result, practical applications often use RNNs that are too small, as large RNNs tend to overfit. To date, existing regularization methods give relatively small improvements on RNNs (Graves, 2013). Dropout is a highly effective way of regularizing feedforward neural networks (Srivastava, 2013) that had enjoyed considerable success. However, it is not clear how to use dropout on RNNs, because a naive application of dropout does not yield good results. In this work, we show how to correctly apply dropout to LSTMs, and demonstrate that this results in great reduction in overfitting.

2. Related work

Dropout (Srivastava, 2013) is a recently regularization method that has enjoyed a lot of success in applications of feed-forward neural networks. While there has been a lot of work extending Dropout (Wang & Manning, 2013; Wan et al., 2013), there has been relatively little research so far in applying dropout to RNNs. The only paper on the topic is (Bayer et al., 2013) which focuses on "marginalized dropout" (from (Wang & Manning, 2013)) which is a noiseless deterministic approximation to conventional dropout. (Bayer et al., 2013) claims that conventional dropout cannot be successfully used with RNNs because its interaction with recurrence causes large variance which hurts learning and causes poor convergence. In this work, we show that a specific application of dropout greatly reduces the overfitting of RNNs.

There has been extensive work on using RNNs for language modeling (Mikolov, 2012; Sutskever, 2013). Moreover, there have been a number of architectural variations on the RNN that are better suited for learning on data with long term dependencies (Hochreiter & Schmidhuber, 1997; Graves et al., 2009; Cho et al., 2014; Jaeger et al., 2007; Koutník et al., 2014). This work focuses on the LSTM which is the most widespread variant of the RNN, although it is likely that our findings are valid for other models.

In this paper, we focus on the following RNN tasks: language modeling, speech recognition, and machine translation. Language modeling is the first task where RNNs achieved substantial success (Mikolov et al., 2010; 2011; Pascanu et al., 2013). RNNs have also been successfully used for speech recognition (Robinson et al., 1996; Graves et al., 2013) and have been applied to machine translation, where they are typically used for language modeling, reranking, or phrase modeling (**?**Cho et al., 2014; Chow et al., 1987; Mikolov et al., 2013).

3. Regularizing RNN with LSTM cells

In this section we describe the deep LSTM 3.1. Next, we show how to regularize them 3.2, and we provide an intuition for why our regularization scheme works.

We use lowerscript to denote timesteps and upperscript to denote layer number. All our states are n-dimensional. Let

$h_k^l \in \mathbb{R}^n$ be a hidden state in layer l in step k. Moreover, let $T_{n,m} : \mathbb{R}^n \to \mathbb{R}^m$ be a linear transform and with a bias ($Wx + b$ for some W and b). Let $\odot$ be a element-wise multiplication and let h_k^0 be an input word vector. We use the activations h_k^L to predict y_k, since L is the number of layers in our deep LSTM.

3.1. Long-short term memory units

The RNNs dynamics can be described in terms of deterministic transitions from previous hidden states to the hidden states in the next step. The transition is a function

$$\text{RNN} : h_k^{l-1}, h_{k-1}^l \to h_k^l$$

For classical RNNs, this function is given by

$$h_k^l = f(T_{n,n}h_k^{l-1} + T_{n,n}h_{k-1}^l), \text{ where } f \in \{\text{sigm}, \tanh\}$$

The LSTM has relatively complicated dynamics that make it easy to "memorize" information for extended number of time steps. The "long term" memory is stored in a vector of *memory cells* $c_k^l \in \mathbb{R}^n$. Although there are many LSTM architectures that differ in their connectivity sturcutre and activation functions, all LSTM architectures have explicit memory cells that make it easy to store information for extended periods of time. The LSTM can decide to overwrite this information, retrieve it, or keep it for the next time step. The LSTM architecture used in our experiments is given by the following equations (Graves et al., 2013):

$$\text{LSTM} : h_k^{l-1}, h_{k-1}^l, c_{k-1}^l \to h_k^l, c_k^l$$
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} h_k^{l-1} \\ h_{k-1}^l \end{pmatrix}$$
$$c_k^l = f \odot c_{k-1}^l + i \odot g$$
$$h_k^l = o \odot \tanh(c_k^l)$$

In these equations, sigm and tanh are applied element-wise. Diagram 1 illustrates the LSTM equations.

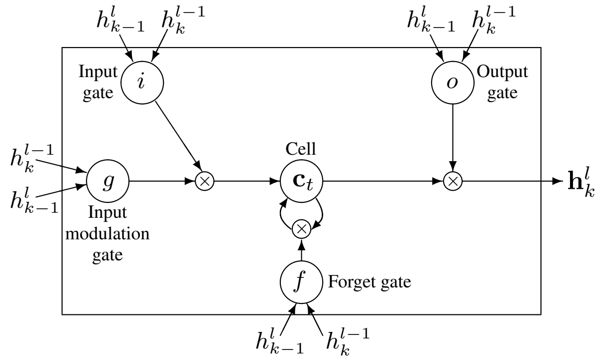

Figure 1. Graphical representation of LSTM memory cells used in this paper (there are minor differences in compare to (Graves, 2013).

3.2. Regularization with Dropout

The main contribution of this paper is the discovery that carefully placed dropout tremendously improves the generalization ability of LSTMs. To be effective, dropout must be placed on the non-recurrent connections 2. The following equation describes it more precisely, where $\mathbf{D}$ is the dropout operator that sets a random subset of its argument to zero:

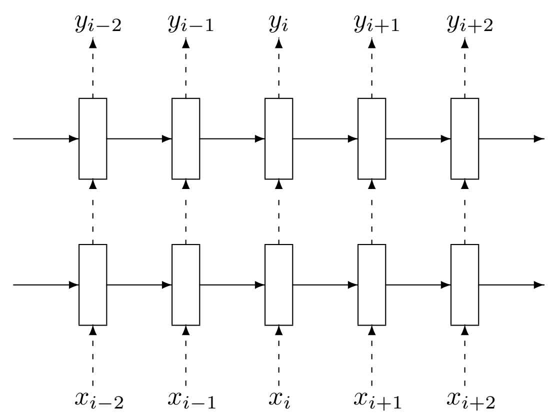

Figure 2. Regularized multilayer RNN. Dashed arrows indicate connections with applied dropout, while solid lines indicate connections where dropout is not applied.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} \mathbf{D}(h_k^{l-1}) \\ h_{k-1}^l \end{pmatrix}$$
$$c_k^l = f \odot c_{k-1}^l + i \odot g$$
$$h_k^l = o \odot \tanh(c_k^l)$$

The dropout operator corrupts the information carried by the units, which forces them to perform their intermediate computations in a more robust manner. At the same time, we do not want to erase all of the information conveyed by the units. We would especially like the units to remember

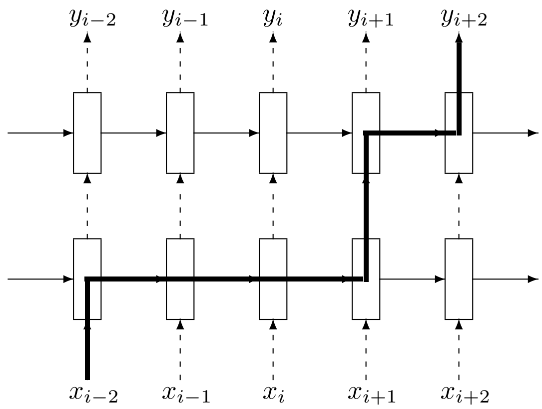

Figure 3. Thick line indicates an exemplary information flow in RNN. Information flow line is crossed $L + 1$ times, where L is depth of network.

Model	Validation set	Test set
A single model		
(Pascanu et al., 2013)		107.5
(Cheng et al.)		100.0
Non-regularized LSTM	120.7	114.5
Medium Regularized LSTM	86.2	82.7
Large Regularized LSTM	82.2	**78.4**
Model averaging		
(Mikolov, 2012)	83.5	89.4
(Cheng et al.)		80.6
2 non-regularized LSTMs	100.4	96.1
5 non-regularized LSTMs	87.9	84.1
10 non-regularized LSTMs	83.5	80.0
2 medium regularized LSTMs	80.6	77.0
5 medium regularized LSTMs	76.7	73.3
10 medium regularized LSTMs	75.2	72.0
2 large regularized LSTMs	76.9	73.6
10 large regularized LSTMs	72.8	69.5
38 large regularized LSTMs	71.9	**68.7**
Model averaging with dynamic RNNs		
(Mikolov & Zweig, 2012)		72.9

Table 1. Word-level perplexity on Penn-tree-bank dataset.

events that occurred many timeseps in the past. Figure 3 shows a possible flow of information from an event that occurred at x_{i-2} to the prediction in the step $i + 2$. We can see that the information is corrupted by dropout only $L + 1$ times, and it is independent of how far in past event occurred. A naive application of dropout would perturb the recurrent connections or the recurrent hidden state, which would greatly reduce the LSTM's memory capacity. By not using dropout on the recurrent connections, the LSTM is able to get most of the benefit of dropout without sacrifising its valuable ability to learn and store information for long periods of time.

4. Experiments

We present here results in there domains: language modeling 4.1, speech recognition 4.2, and machine translation 4.3.

4.1. Language modeling

We have conducted word-level prediction experiments on Penn tree bank (PTB) dataset (Marcus et al., 1993). This dataset consists of 929k training words, 73k validation words, and 82k test words. It has 10k words in vocabulary. We have trained two various size regularized LSTMs model. Both models are a two-layer LSTM unrolled for 35 steps. We set mini-batch to 20.

Medium size model has 650 units per layer whose parameters are initialized uniformly in $[-0.05, 0.05]$. We apply 50% dropout on the non-recurrent connections. We train for 39 epochs, starting with learning rate of 1, and after 6 epochs we decrease it by a factor of 1.2 in every epoch. We clip the norm of the gradients (normalized by minibatch size) at 5.

Large size model has 1500 units per layer whose parameters are initialized uniformly in $[-0.04, 0.04]$. We apply 65% dropout on the non-recurrent connections. We train for 55 epochs, starting with learning rate of 1, and after 14 epochs we decrease it by a factor of 1.15 in every epoch. We clip the norm of the gradients (normalized by minibatch size) at 10.

Table 1 compares various models with our models, and fig-

the meaning of life is that only if an end would be of the whole supplier. widespread rules are regarded as the companies of refuses to deliver. in balance of the nation 's information and loan growth associated with the carrier thrifts are in the process of slowing the seed and commercial paper.

the meaning of life is nearly in the first several months before the government was addressing such a move as president and chief executive of the nation past from a national commitment to curb grounds. meanwhile the government invests overcapacity that criticism and in the outer reversal of small-town america.

Figure 4. Some interesting samples drawn from large regularized model conditioned on "The meaning of life is". We have removed "unk", "N", "$" from possible outcomes.

Model	Training set	Validation set
Non-regularized LSTM	71.6	68.9
Regularized LSTM	69.4	**70.5**

Table 2. Frame-level accuracy on Icelandic Speech Dataset. **size of dataset?**

Model	Test perplexity	Test BELU score
Non-regularized LSTM	5.8	25.9
Regularized LSTM	5.0	29.03
LIUM system		33.30

Table 3. Results on the English to French translation task.

ure 4 shows samples draw from a single large size regularized model.

4.2. Speech recognition

Deep Neural Networks have been used for acoustic modeling for more than half a century (the reader is encouraged to read (Bourlard & Morgan, 1993) for a good review). Acoustic modeling is a key component in mapping acoustic signals to sequences of words, as it models $p(s_t|X)$ where s_t is the phonetic state at time t, and X the acoustic observation. Recent work has shown that LSTMs are very powerful models for acoustic modeling (Sak et al., 2014), and much smaller networks (in number of parameters) are able to overfit the data more easily. The primary metric to measure acoustic models is frame accuracy, which is measured on each s_t prediction for all t. Generally, this metric correlates with the actual metric of interest, the Word Error Rate (WER). However, since computing WER involves using a language model and tuning the decoding parameters for every change in the acoustic model, we decided to report frame accuracy. Table 2 clearly shows the positive effect of dropout by improving frame accuracy. Not surprisingly, the training frame accuracy drops due to the noise added during training, but as is often the case with dropout, this yields models that generalize better to unseen data. Note that the testing set is easier than the training set, given its higher accuracy. We report the performance of an LSTM on an internal Google Icelandic Speech dataset, which is relatively small, so overfitting is a greater concern.

4.3. Machine translation

We report the performance of the LSTM on a machine translation task. We formulate the translation task as a language modelling task, where the LSTM is trained to assign high probability to the correct translation given a source sentence. Thus, the LSTM is trained on sequences of the form `(source sentence, target sentence)` (**?**Cho et al., 2014). We compute the translations using a simple left-to-right decoder; see (**?**) for more details. We ran this experiment on the WMT'14 English to French dataset, on the "selected" subset from (Schwenk, 2014) which has 340M French words and 304M English words. Our LSTM that has 4 hidden layers, where both the layers and the word embeddings are 1000-dimensional, and where the English vocabulary has 160,000 words and the French vocabulary has 80,000 words, as in (**?**). We found the optimal dropout probability to be 0.2. Table 3 shows the comparative performance of both LSTMs. While this particular LSTM does not beat the standard SMT system (Schwenk et al., 2011), this result clearly shows that dropout greatly improves the performance of the LSTM.

5. Discussion

We presented a simple application of dropout to LSTMs that resulted in large performance boosts on many separate domains. Our results make dropout useful for RNNs, and our results suggest that this type of dropout could improve performance on a wide variety of applications.

References

Bayer, Justin, Osendorfer, Christian, Chen, Nutan, Urban, Sebastian, and van der Smagt, Patrick. On fast dropout and its applicability to recurrent networks. *arXiv preprint arXiv:1311.0701*, 2013.

Bourlard, H. and Morgan, N. *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1993.

Cheng, Wei-Chen, Kok, Stanley, Pham, Hoai Vu, Chieu, Hai Leong, and Chai, Kian Ming A. Language modeling with sum-product networks.

Cho, Kyunghyun, van Merrienboer, Bart, Gulcehre, Caglar, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chow, Y, Dunham, M, Kimball, O, Krasner, M, Kubala, G, Makhoul, J, Price, P, Roucos, S, and Schwartz, R. Byblos: The bbn continuous speech recognition system. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, volume 12, pp. 89–92. IEEE, 1987.

Graves, Alex. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Graves, Alex, Liwicki, Marcus, Fernández, Santiago, Bertolami, Roman, Bunke, Horst, and Schmidhuber, Jürgen. A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):855–868, 2009.

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.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Jaeger, Herbert, Lukoševičius, Mantas, Popovici, Dan, and Siewert, Udo. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3):335–352, 2007.

Koutník, Jan, Greff, Klaus, Gomez, Faustino, and Schmidhuber, Jürgen. A clockwork rnn. *arXiv preprint arXiv:1402.3511*, 2014.

Marcus, Mitchell P, Marcinkiewicz, Mary Ann, and Santorini, Beatrice. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Mikolov, Tomáš. *Statistical language models based on neural networks*. PhD thesis, Ph. D. thesis, Brno University of Technology, 2012.

Mikolov, Tomas and Zweig, Geoffrey. Context dependent recurrent neural network language model. In *SLT*, pp. 234–239, 2012.

Mikolov, Tomas, Karafiát, Martin, Burget, Lukas, Cernockỳ, Jan, and Khudanpur, Sanjeev. Recurrent neural network based language model. In *INTERSPEECH*, pp. 1045–1048, 2010.

Mikolov, Tomas, Deoras, Anoop, Povey, Daniel, Burget, Lukas, and Cernocky, Jan. Strategies for training large scale neural network language models. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pp. 196–201. IEEE, 2011.

Mikolov, Tomas, Le, Quoc V, and Sutskever, Ilya. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

Pascanu, Razvan, Gulcehre, Caglar, Cho, Kyunghyun, and Bengio, Yoshua. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

Robinson, Tony, Hochberg, Mike, and Renals, Steve. The use of recurrent neural networks in continuous speech recognition. In *Automatic speech and speaker recognition*, pp. 233–258. Springer, 1996.

Sak, H., Vinyals, O., Heigold, G., Senior, A., McDermott, E., Monga, R., and Mao, M. Sequence discriminative distributed training of long short-term memory recurrent neural networks. In *Interspeech*, 2014.

Schwenk, Holger. University le mans, 2014. URL `http://www-lium.univ-lemans.fr/~schwenk/cslm_joint_paper/`.

Schwenk, Holger, Lambert, Patrik, Barrault, Loïc, Servan, Christophe, Afli, Haithem, Abdul-Rauf, Sadaf, and Shah, Kashif. Lium's smt machine translation systems for wmt 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 464–469. Association for Computational Linguistics, 2011.

Srivastava, Nitish. *Improving neural networks with dropout*. PhD thesis, University of Toronto, 2013.

Sutskever, Ilya. *Training Recurrent Neural Networks*. PhD thesis, University of Toronto, 2013.

Wan, Li, Zeiler, Matthew, Zhang, Sixin, Cun, Yann L, and Fergus, Rob. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1058–1066, 2013.

Wang, Sida and Manning, Christopher. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 118–126, 2013.