# Recurrent neural network regularization

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

## Abstract

We present a simple regularization technique of recurrent neural networks (RNNs) with long short term memory (LSTM) units. This technique is based on dropout and gives tremendous performance boost. We show that it is beneficial in variety sequence modelling problems like language modeling, speech recognition, and machine translation.

## 1. Introduction

Recurrent neural networks yields the state-of-the-art performance on many sequence modelling tasks like language modelling, and speech recognition. Moreover, recent results in machine translation (Cho et al., 2014) shows their potential use in this field as well.

Model averaging gives significant improvement in various natural language processing tasks (Mikolov, 2012). This can be explained in two ways. Various models capture various aspects of language modeling, or models were not properly regularized. We show that with a single model regularized RNN, we are able to achieve better results than state-of-the-art from averaging multiple models.

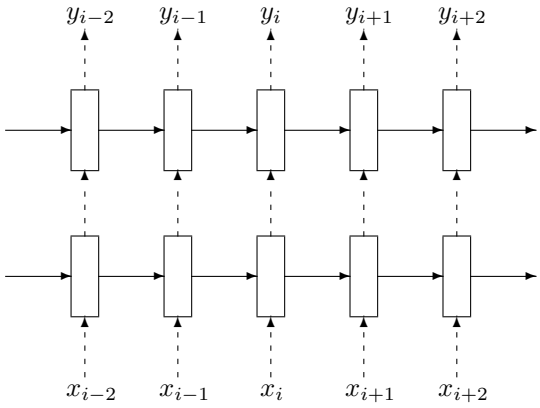


*Figure 1.* Regularized multilayer RNN. Dashed arrows indicate connections with applied dropout, while solid lines indicate connections where dropout is not applied.

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} L \begin{pmatrix} \mathbf{D}(h_k^{l-1}) \\ h_{k-1}^l \end{pmatrix}$$
$$c_k^l = f \odot c_{k-1}^l + i \odot g$$
$$h_k^l = o \odot \tanh(c_k)$$

*Figure 2.* To describe dynamics of multilayer LSTM with dropout, we use multi-indexing. Lower indices correspond to dynamics over time, and upper indices correspond to dynamics over layers. For simplicity, we denote by $L$ a linear transform with bias ($Wx + b$), and by $D$ a dropout layer. $\odot$ is a element-wise multiplication. $h_k^0$ is a input word-vector, and $h_k^L$ is used to predict $y_k$ ($L$ is a number of layers).

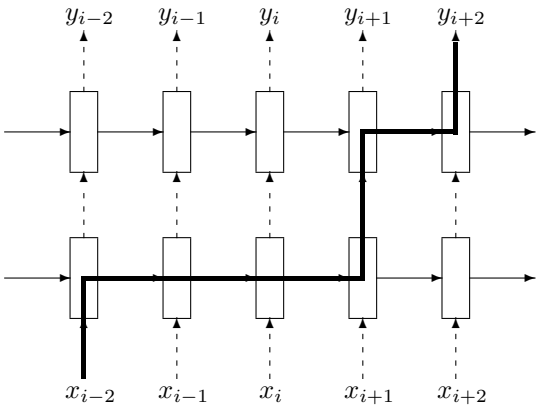


*Figure 3.* Thick line indicates an exemplary information flow in RNN. Information flow line is crossed $L$ times, where $L$ is depth of network.

| Model | Validation set | Test set |
|---|---|---|
| A single model | | |
| Previous state-of-the-art [1] | | 107.5 |
| Regularized RNN | **87** | **83** |
| Model averaging | | |
| Previous state-of-the-art [2] | 83.5[3] | 89.4 |
| 2 regularized RNNs | | |
| 5 regularized RNNs | | |
| 10 regularized RNNs | | |

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

## 2. Related work

## 3. Long-short term memory units

## 4. Regularization with dropout

## 5. Intuition

## 6. Experiments

### 6.1. Language modeling

### 6.2. Machine translation

### 6.3. Speech recognition

## 7. Discussion

## References

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.

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

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

[1] (Pascanu et al., 2013)

[2] (Mikolov, 2012)

[3] Weight of individual models are tuned to minimize this score. This few parameters are fit on this validation set