Sequence to Sequence Learning with Neural Networks

Ilya Sutskever, Oriol Vinyals, Quoc V. Le

Neural Information Processing Systems, 2014

Problems Addressed

1. The paper presents a general end-to-end approach to sequence learning that makes minimal assumptions about sequence structure. In doing so, it shows a straightforward application of LSTM architecture in solving general sequence-to-sequence problems.
2. RNNs can easily map sequences to sequences whenever the alignment between inputs and outputs is known ahead of time. This becomes complicated when the input and output sequences have different lengths with non-monotonic relationships.

Proposed Solution

1. The main idea is to obtain a large fixed dimensional vector representation of the input sequence by using one LSTM to read the input sequence one timestep at a time, and then using another LSTM to extract the output sequence from that vector.
2. The second LSTM in this case is essentially a LSTM-LM that is conditioned on the input sequence.
3. Therefore, the LSTM's goal is to estimate the conditional probability $p(y_1, ..., y_{T'}|x_1, ..., x_T)$, where $y_1, ..., y_{T'}$ and $x_1, ..., x_T$ are the output and input sequences respectively (lengths may differ).
4. The paper uses LSTM formulation from Graves (2014), with some changes.

$$p(y_1, ..., y_{T'}|x_1, ..., x_T) = \prod_{t=1}^{T'} p(y_t|v, y_1, ..., y_{t-1})$$

   - In the above equation, each $p(y_t|v, y_1, ..., y_{t-1})$ distribution is represented with a softmax over all the words in the vocabulary.
   - The model requires each sentence to end with a special <EOS> symbol, so as to enable the model to define distributions over all possible lengths.
   - Both the languages had a fixed vocabulary - 1,60,000 most frequent words for source language and 80,000 most frequent words for target language. Unknown words were replaced by a special "UNK" token.
   - The actual model is slightly different, as it uses two LSTMs - this **increases model parameters at negligible cost** and makes it natural to train the LSTM on multiple language pairs simultaneously.

Experiments

1. This method was applied to WMT'14 English to French Machine Translation task in two ways.
    - The model was used to directly translate the input sequence without using a reference SMT (Statistical Machine Translation) system.
    - Also used LSTM to rescore the 1000-best lists from baseline system - done by computing log probability of every hypothesis with the LSTM and taking an even average with the baseline score and LSTM score.
2. Cased BLEU score was used to evaluate the quality of translations. Scores were computed using using multi-bleu.pl on the tokenized predictions and ground truth. (Evaluating the best WMT'14 system (whose predictions can be downloaded from statmt.org\matrix this way gives it a score of 37.0, which is greater than the 35.8 reported by statmt.org\matrix)
3. The models were trained on a subset of 12M sentences consisting of 348M French words and 304M English words. This translation task and test set was chosen because:
    - Tokenized training and test set is publicly available.
    - 1000-best lists from baseline SMT are also available.
4. The training aim was to maximize the log probability of correct translation T given source sentence S. Hence, the training objective was:

$$\frac{1}{|S|} \sum_{(T,S)\in S} \log p(T|S), \ S \textit{ is the training set.}$$

5. After training, the most likely translation is found according to:

$$\hat{T} = \arg\max_{T} p(T|S)$$

6. Most likely translation was found using left to right **beam search decoder**.
    - It maintains B partial hypotheses (prefix of some translation).
    - At each time step, each partial hypothesis is extended by every possible word in vocabulary, and B most likely hypotheses are selected once again.
    - As soon as "<EOS>" is obtained in any hypothesis, it is removed from beam and added to set of complete hypotheses.
    - This performs well even with beam size 1, although 2 provides most benefits.
7. **Training Details**:
    - The LSTM uses 4 layers, and the order of words in the input sequence is reversed. Each layer has 1000 cells and 1000 dimensional word embeddings.
    - The model uses a naive softmax over the 80,000 words at each output.
    - The resulting LSTM has 384M parameters - 64M are pure recurrent connections (32M for encoder and 32M for decoder).
    - All LSTM parameters initialized uniform distribution between -0.08 and 0.08.
    - The model used SGD **without** momentum, with a fixed learning rate of 0.7. After 5 epochs, the learning rate is halved every half epoch. The model was trained for 7.5 epochs.
    - Used batches of 128 sequences for gradient. Since sentences can have different lengths, ensured that all sentences in a minibatch are of similar lengths, leading to a **2x** speedup.
    - Exploding gradients were avoided by enforcing a constraint on norm of gradient and scaling it when it exceeded the threshold. For each training batch, computed s = ||g||, where g is gradient by 128. If s > 5, set g = $\frac{5g}{s}$.

- The model was parallelized on an 8 GPU machine, with each layer of LSTM run on one GPU, while the remaining 4 GPUs were used to parallelize softmax, so each GPU was responsible for multiplying by 1000×20000 matrix. This achieved 6300 words per second (both English and French) with a minibatch size of 128, and took 10 days to train.

Results

1. Deep LSTMs outperformed shallow LSTMs, with each layer reducing perplexity by 10%.
2. Reversing source sentences reduced LSTM's test set perplexity from 5.8 to 4.7, and test BLEU scores improved from 25.9 to 30.6. LSTMs trained on reversed sentences also did better on longer sentences, suggesting better memory utilization (Better reasoning found in "Visualizing and Understanding Recurrent Networks").
3. Best results were obtained with an **ensemble** of LSTMs that differed in their random initializations and in their random order of minibatches.

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.

Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
Best WMT'14 result [9]	**37.0**
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed LSTMs	**36.5**
Oracle Rescoring of the Baseline 1000-best lists	∼45

Table 2: Methods that use neural networks together with an SMT system on the WMT'14 English to French test set (ntst14).

Claims

1. Even though decoded translations do not outperform best WMT'14 system, it is the first time that a pure neural translation system outperforms a phrase-based SMT baseline on a larger scale MT task, despite the inability to handle words outside vocabulary.
2. The LSTM did well on long sentences too.
3. Vector representations are sensitive to word ordering, but insensitive to replacement of active voice with passive voice. (2D Projections obtained using PCA)

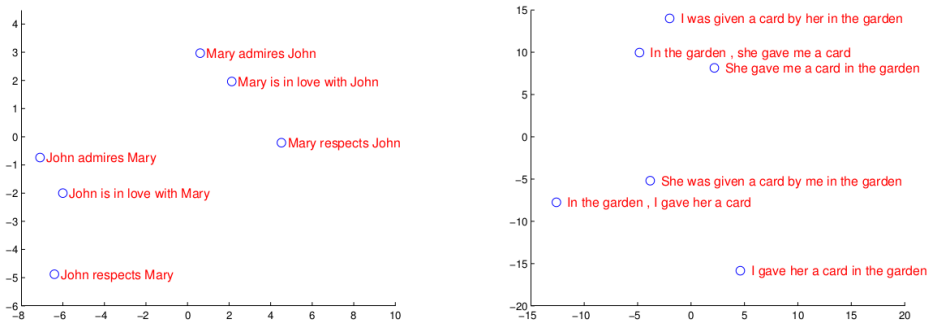

Figure 2: The figure shows a 2-dimensional PCA projection of the LSTM hidden states that are obtained after processing the phrases in the figures. The phrases are clustered by meaning, which in these examples is primarily a function of word order, which would be difficult to capture with a bag-of-words model. Notice that both clusters have similar internal structure.

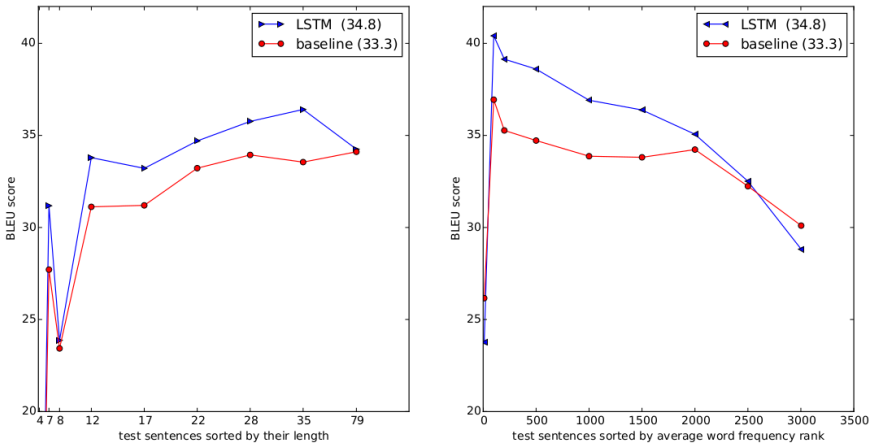

Figure 3: The left plot shows the performance of our system as a function of sentence length, where the x-axis corresponds to the test sentences sorted by their length and is marked by the actual sequence lengths. There is no degradation on sentences with less than 35 words, there is only a minor degradation on the longest sentences. The right plot shows the LSTM's performance on sentences with progressively more rare words, where the x-axis corresponds to the test sentences sorted by their "average word frequency rank".

Bibliography

1. M. Auli, M. Galley, C. Quirk, and G. Zweig. Joint language and translation modeling with recurrent neural networks. In EMNLP, 2013.

2. D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

3. Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. In Journal of Machine Learning Research, pages 1137–1155, 2003.

4. Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.

5. K. Cho, B.Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Arxiv preprint arXiv:1406.1078, 2014.

6. D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In CVPR, 2012.

7. G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing - Special Issue on Deep Learning for Speech and Language Processing, 2012.

8. J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In ACL, 2014.

9. Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh's phrase-based machine translation systems for wmt-14. In WMT, 2014.

10. A. Graves. Generating sequences with recurrent neural networks. In Arxiv preprint arXiv:1308.0850, 2013.

11. A. Graves, S. Fern´andez, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In ICML, 2006.

12. K. M. Hermann and P. Blunsom. Multilingual distributed representations without word alignment. In ICLR, 2014.

13. G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 2012.

14. S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Master's thesis, Institut fur Informatik, Technische Universitat, Munchen, 1991.

15. S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

16. S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 1997.

17. S. Hochreiter and J. Schmidhuber. LSTM can solve hard long time lag problems. 1997.

18. N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In EMNLP, 2013.

19. A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

20. Q.V. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, and A.Y. Ng. Building high-level features using large scale unsupervised learning. In ICML, 2012.

21. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998.

22. T. Mikolov. Statistical Language Models based on Neural Networks. PhD thesis, Brno University of Technology, 2012.

23. T. Mikolov, M. Karafi´at, L. Burget, J. Cernock'y, and S. Khudanpur. Recurrent neural network based language model. In INTERSPEECH, pages 1045–1048, 2010.

24. K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. BLEU: a method for automatic evaluation of machine translation. In ACL, 2002.

25. R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063, 2012.

26. J. Pouget-Abadie, D. Bahdanau, B. van Merrienboer, K. Cho, and Y. Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. arXiv preprint arXiv:1409.1257, 2014.

27. A. Razborov. On small depth threshold circuits. In Proc. 3rd Scandinavian Workshop on Algorithm Theory, 1992.

28. D. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. Nature, 323(6088):533–536, 1986.

29. H. Schwenk. University le mans. http://www-lium.univ-lemans.fr/~schwenk/cslm_ joint _paper/, 2014. [Online; accessed 03-September-2014].

30. M. Sundermeyer, R. Schluter, and H. Ney. LSTM neural networks for language modeling. In INTERSPEECH, 2010.

31. P. Werbos. Backpropagation through time: what it does and how to do it. Proceedings of IEEE, 1990.