Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

CSE 291 - Advanced Statistical Natural Language Processing
Nishant D. Gurnani

May 23, 2017

Outline

Outline

Motivation

Deep neural networks (DNNs) are powerful models that work well whenever large labeled training sets are available

Drawbacks:

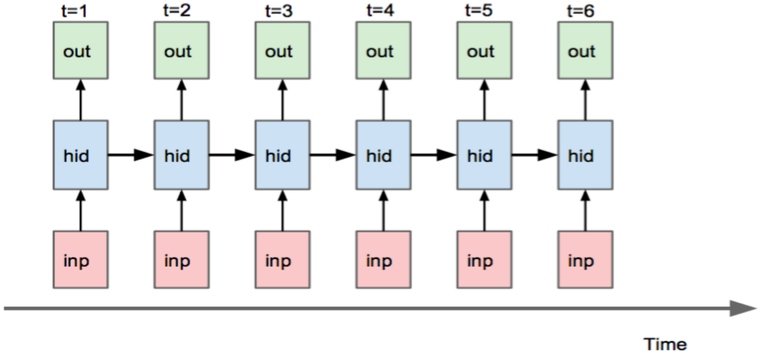
- Need inputs and outputs to be vectors of fixed dimensionality
- Consequently, cannot map sequences to sequences
- Significant limitation since many important problems (machine translation, image caption generation) are best expressed with sequences whose lengths are not known a-priori

Goal: a general sequence to sequence neural network

Recurrent Neural Networks

Given a sequence of inputs $(x_1, \ldots, x_T)$, a standard RNN computes a sequence of outputs $(y_1, \ldots, y_T)$ by iterating the following equation:

$$h_t = sigm(W^{hx} x_t + W^{hh} h_{t-1})$$
$$y_t = W^{yh} h_t$$



RNN Drawbacks

- Have a one-to-one correspondence between the inputs and outputs
- Have trouble learning "long-term dependencies"
  - vanishing gradient problem
  - exploding gradient problem
  - Hochreiter (1991); Bengio et. al (1994)

RNN Drawbacks

- Have a one-to-one correspondence between the inputs and outputs
- Have trouble learning "long-term dependencies"
  - vanishing gradient problem $\rightarrow$ **LSTM**
  - exploding gradient problem $\rightarrow$ **Gradient clipping**
  - Hochreiter (1991); Bengio et. al (1994)

Long Short-Term Memory (LSTM)

- Hochreiter and Schmidhuber (1997)
- An RNN architecture that is good at long-term dependencies
- Has almost no vanishing gradients

Key Insights:

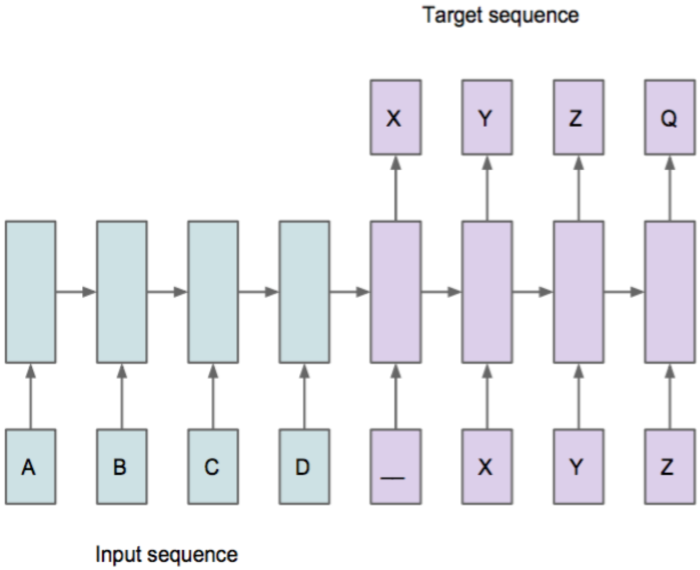
- RNNs overwrite the hidden state
- LSTMs add to the hidden state
  - compute a delta to the hidden state which we then add to it
  - addition has nice gradients
  - results in LSTM being good at noticing long-range correlations

Outline

Main Idea

- Neural networks are excellent at learning very complicated functions
- "Coerce" a neural network to read one sequence and produce another
- Learning should take care of the rest

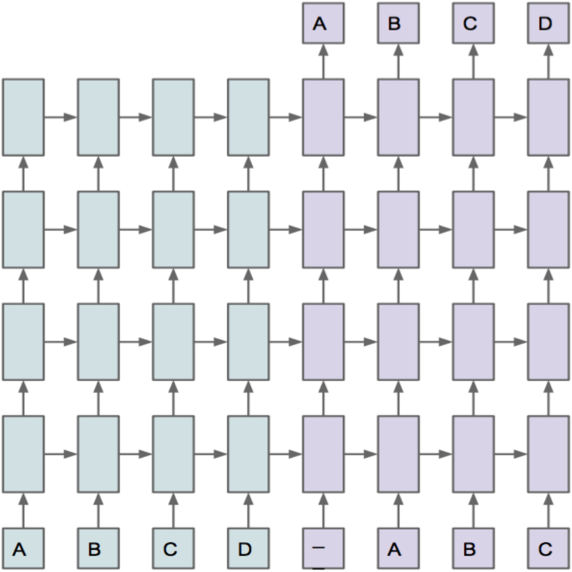
Model



LSTM hidden state

- The LSTM needs to read the entire input sequence, and then produce the target sequence “from memory”
- The input sequence is stored by a **single** LSTM hidden state
- So hidden state must be large

Deep model with large hidden state



Similar Work

- Kalchbrenner and Blunsom (2013) Recurrent Continuous Translation Models $\rightarrow$ convolutional encoder, recurrent decoder
- Cho et. al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation $\rightarrow$ recurrent encoder, recurrent decoder
- Bahdanau et. al (2014 arxiv version) Neural machine translation by jointly learning to align and translate $\rightarrow$ recurrent encoder, recurrent decoder + attention

Outline

Dataset

- WMT'14 English to French
- 12M sentences
- 348M French words
- 304M English words
- Train on 30% of training data which is a clean "selected" subset
- Choose this subset because of public availability of a tokenized training and test set together with 1000-best lists from the baseline SMT

Training

We define a distribution over output sequences given input sequences and maximize the log probability of a correct translation T given the source sentence S.

Training Objective:

$$\frac{1}{|S|} \sum_{(T,S)\in S} \log p(T|S)$$

Once training is complete, we produce translations by finding the most likely translation according to the LSTM:

$$\hat{T} = \operatorname*{argmax}_{T} p(T|S)$$

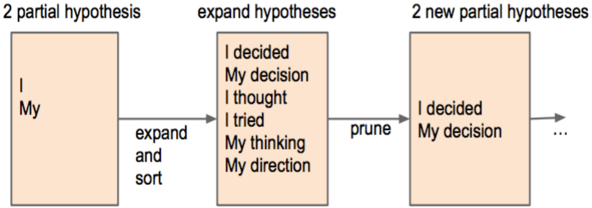
Searching for the most likely translation is done using a simple left-to-right beam search decoder

Decoding

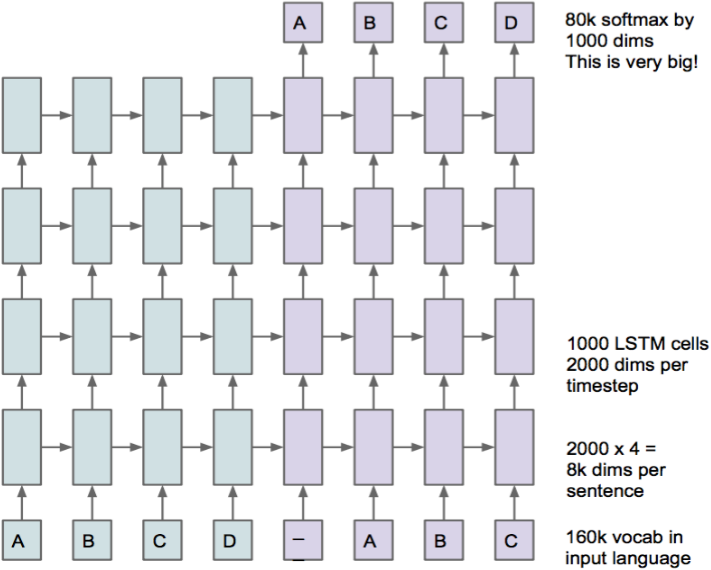
- Since there are exponentially many sentences, how do we find the sentence with the highest probability?
- Search problem: use simple greedy beam search

Decoding in a nutshell

- proceed left to right
- maintain N partial translations
- expand each translation with possible next words
- discard all but the top N new partial translations



Experimental Setup



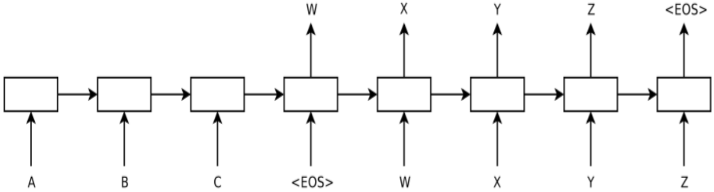
Learning Parameters

For a change the learning parameters are fairly simple and straightforward:

- batch size $= 128$
- learning rate $= 0.7/$batch size
- initialize uniform between -0.8 and 0.8
- norm of gradient is clipped to 5
- learning rate is halved every 0.5 epochs after 5 epochs
- no momentum

Reversing Source Sentences

- Authors find that LSTM learns much better when the source sentences are reverse
- Results in test BLEU scores of decoded translations increasing from 25.9 to 30.6
- Retroactively provide and explanation suggesting that doing so introduces many short term dependencies in the data that make the optimization problem much easier



Experiments

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**

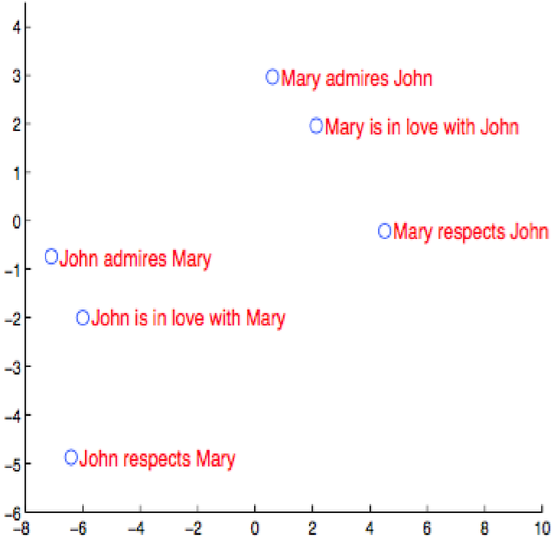
Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.

Experiments

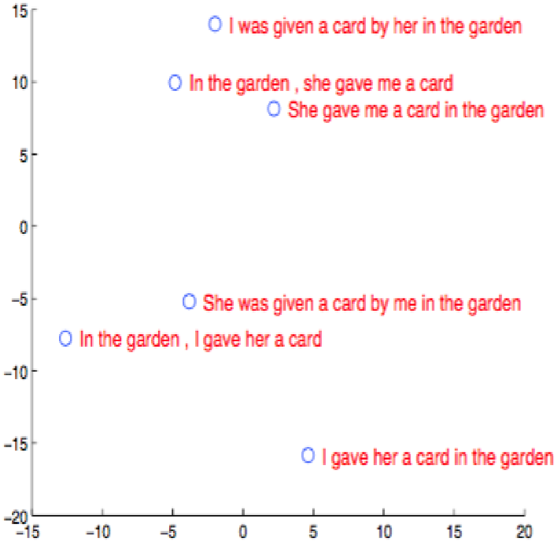
Method	test BLEU score (ntst14)
Baseline System [29]	33.30
Cho et al. [5]	34.54
State of the art [9]	**37.0**
Rescoring the baseline 1000-best with a single forward LSTM	35.61
Rescoring the baseline 1000-best with a single reversed LSTM	35.85
Rescoring the baseline 1000-best with an ensemble of 5 reversed 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).

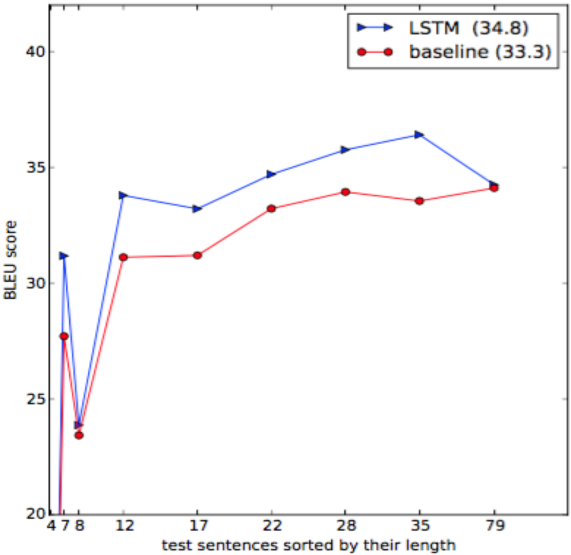
2-dimensional PCA projection



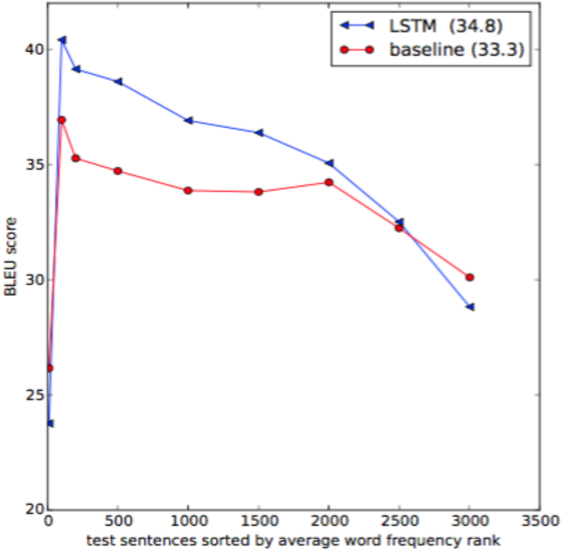
2-dimensional PCA projection



Performance as a function of sentence length



Performance on sentences with progressively more rare words



Related Work

- Bahdanau et al., ICLR 2015 Neural Machine Translation by Jointly Learning to Align and Translate
- Lee, et al., TACL 2017 Fully Character-Level Neural Machine Translation without Explicit Segmentation
- Wu et al., arxiv 2016 Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Related Work

- Ranzanto et. al, 2015 Sequence Level Training with Recurrent Neural Network
- Luong et. al, ICLR 2016 Multi-task Sequence to Sequence Learning
- Wiseman and Rush, EMNLP 2016 Sequence to Sequence Learning as Beam Search Optimization

Discussion FAQ

Q: What happens when the encoder and decoder models have different numbers of hidden layers? Is there a constraint that they need to have the same number?

Q: Why didn't the authors try deep bidirectional LSTMs?

Q: Does reversing the order of words in source sentences have any linguistic rationale?

Q: For long sentences and bigger depth, doesn't the model complexity increase beyond the expressive capability of the model?

Q: Can the learned sentence representations from a language pair (Ex. English to French) be used to train a LSTM decoder for another target language (Ex. English to Spanish)?