Sequence to Sequence Learning with Neural Networks

1 Citation

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.

`http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf`

2 Abstract

We translate from English to French with an encoder-decoder network. Our encoder is a stack of LSTMs that produces a fixed size output. Our decoder takes the this as input and produces a variable length output. A key insight we make is that the input sentence should be fed in reverse order. We get a BLEU score of 34.8 compared to 33.3 from a classic phrase based Statistical Machine Translation (SMT) system. Using our network to rank the SMT phrase pairs gets a BLEU score of 36.5 (near state of the art).

3 Introduction

A key limitation of neural networks is they require a fixed-size input. An encoder-decoder network overcomes this. We focus on the WMT'14 English to French translation task. We penalize our system when it encounters words outside its vocabulary. Reversing input sentences introduces short term dependencies to improve the model.

4 The Model

We stack LSTMs to map the reversed input sentence into a fixed size vector. We then use another stack of LSTMs to map the fixed sized vector into an output sequence. Sentences are terminated with an end-of-sentence (EOS) word. We use four-layer LSTMS. The final LSTM feeds into a softmax over the vocabulary.

To understand why reversing the input sentence is important, suppose that we want to map the sequence a, b, c to α, β, γ, Notice that in the sequence $a, b, c, \alpha, \beta, \gamma$ that a and α are very far apart. This means the output LSTMs will need a good memory to remember what started the sentence. On the other hand, if we feed in a reversed sentence, then the sequence $c, b, a, \alpha, \beta, \gamma$ has a and α right near each other and also has b and β reasonably close as well. This encourages better gradient flow.

5 Experiments

We train on 12M sentences (348M French words, 304M English words). We use 160K frequent words from the source language and 80K frequent words from the target language. Our training objective is,

for training set D,

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

At test time, we seek $\hat{T} = \text{argmax}_T p(T|S)$. We find this with beam search (beam size 2).

Reversing the input sentence boosts the BLEU score by about 4.

LSTMs have 1000 cells. Word embeddings have size 1000. LSTM parameters are initialized from a uniform distribution between $[-0.08, 0.08]$. We train for 7.5 epochs using SGD (learning rate 0.7) with momentum. After 5 epochs, we halved the learning rate every 0.5 epochs. Our batch size is 128. We do gradient clipping by computing $s = ||g||$ where g is the gradient divided by 128. If $s > 5$, we set the gradient to $\frac{5g}{s}$. We made sure sentences in a minibatch were about the same length to avoid making short sentences wait for long sentences to finish processing.

See the abstract of this summary for the metrics. The LSTM does well even if the sentence is long. We project our fixed-size vectors into 2D space with PCA and find that the vector captures word order and insensitive to passive vs. active voice.

6 Related Work

Most neural network translation work so far just rescores the phrase pairs of an SMT. Some people use Convolutional Neural Networks (CNNs) to turn sentences into vectors, but this loses word order. Others use attention mechanisms.

7 Conclusion

Our encoder-decoder LSTM-based network beats the SMT. We reverse the source sentence for big gains. We get good performance on long sentences while other researchers previous work struggled with them.