Natural Language Understanding with Distributed Representations - Assignment 4
Due Nov 25 (11:59 PM), 2016

In this assignment, you will build a neural machine translator to translate the source sentence F (in our case, Japanese) to target sentence E (in our case, English) on a toy dataset [1] with different model architectures. You can use TF's library code for seq2seq or attention.

1 Data Preparation (2 pts)

Dataset can be downloaded from https://github.com/neubig/nmt-tips/tree/master/data. The original dataset is already split into train (10000 sentences), dev (500 sentences), and test (500 sentences), with parallel corpora of English and Japanese sentences. At this point you should be very familiar with data preparation, which is the same as all the assignments before.

2 Neural Machine Translation

2.1 Simple Encoder-Decoder Model (3 pts)

Neural machine translation is a kind of SMT that learns the probabilistic model $P(E|F)$ using neural networks. Encoder-decoder models[1] are the simplest version of NMT. The idea is relatively simple: we read in the words of a target sentence one-by-one using a recurrent neural network, then predict the words in the target sentence. For decoding, you can use greedy search instead of beam search for simplicity. Please read carefully Tensorflow's sequence to sequence tutorial[2] carefully before you start building your model.

Please plot perplexity on your train / dev set, and report BLEU score on test set.

2.2 Attention-based Neural Machine Translation (3 pts)

One important property of the simple encoder-decoder model for neural machine translation is that a whole source sentence is compressed into a single real-valued vector c. Unfortunately, considering the complexity of any natural language sentence, it is quite easy to guess that this mapping must be highly nonlinear and will require a huge encoder.

An attention-based encoder-decoder[2] model has a bidirectional recurrent neural network as an encoder and a decoder with the attention mechanism, which trains "attention" on different words instead of trying to encode all the information into one vector. You can find more details on Chapter 6 of Cho's lecture notes[3]. In terms of style of attention, you can implement any attention architecture of your choice.

In this section you will implement an attention-based encoder-decoder for machine translation. Please plot perplexity on your train / dev set, and report BLEU score on test set.

[1] https://github.com/neubig/nmt-tips
[2] https://www.tensorflow.org/versions/r0.11/tutorials/seq2seq/index.html
[3] https://arxiv.org/pdf/1511.07916v1.pdf

3 Writeup (2 pts)

The deadline is Friday, Nov 25th. Please submit your report and corresponding code to dl4nlp2016@gmail.com. Besides the reporting the results from sections above, the paper (< 3 pages) should also consist of:

- description of the architecture
- description of training procedure(what optimization method(Adam,SGD,Adadelta,etc.), learning rate, dropout, train/valid/test error, etc.)
- summary of your findings

References

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

[2] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. EMNLP, 2014.