Summary 15: Sequence to Sequence Learning with Neural Networks

Susan Cherry

Deep Neural Networks are extremely powerful and result in excellent results on a number of challenging problems, but they have some drawbacks. DNNs can only be applied to problems whose inputs and outputs can be encoded with vectors of fixed dimensionality. Many important problems are represented best by sequences whose lengths are not known beforehand, so this is a serious drawback. This paper shows that a straightforward application of Long Short-Term Memory (LSTM) architecture can be used to read the input sequence on timestep at a time to obtain large fixed dimensional vector representation. Then another LSTM is used to extract the output sequence from that vector.

First the authors describe the model, which is the Recurrent Neural Network (RNN). RNN is a generalization of feedforward neural networks to sequences. The paper's model differs from the RNN model described in three important ways. First, they use two different LSTMs, which increases the number of parameters at negligible computational cost and makes it easy to train the LSTM on multiple language pairs simultaneously. Second, they find that deep LSTMs significantly outperform shallow, so use LSTM with four layers. Third, they find it extremely valuable to reverse the order of the words in a input sentence.

They apply their method to the WMT'14 English to French dataset. The core of their experiments involved training a large deep LSTM on many sentence pairs by maximizing the log probability of a correct translation given the source sentence. Once the training is complete, they produce translations by finding the most likely translation according to the LSTM by using abeam search decoder. As mentioned earlier, the LSTM learns much better when the source sentences are reversed, probably because this introduces many short term dependencies to the dataset. The authors find that the LSTM models are fairly easy to train and outline the complete training details in section 3.4. They parallelize the model using an 8-GPU machine.

The use the cased BLEU score to evaluate the quality of the translations. The best results are obtained with an ensemble of LSTMs that differ in random initializations and in random order of minibatches. To their surprise, the authors also find that LSTM performs well on long sentences. The authors conclude by describing related work and how they suspect that this work could be improved with the results of their paper.

Overall, the paper shows that a large deep LSTM with almost now assumptions about the problem structure can outperform a standard SMT-based system. The success of the LSTM means that it is likely to perform well on other sequence learning problems as long as there is sufficient training data. They also conclude that it is important to find a problem encoding that has a large number of short term dependencies (such as reversing the words in the sentences). Perhaps the most important takeaway of this paper is that a simple, relatively unoptimized approach can outperform an SMT system. Further work along these lines will likely lead to greater translation accuracy.