Neural Machine Translation with attention mechanisms in encoding and decoding processes

Wenyue Liu
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599
wenyue@ad.unc.edu

Abstract

In this paper, we used Sequence-to-sequence (seq2seq) model (Sutskever et al., 2014) to build a French to English Neural Machine Translation (FENMT) model. To boost this performance of this NMT model, we compared and implemented both GRU, LSTM, bi-GRU, bi-LSTM, pre-trained embedding methods, encoder attention, decoder attention and both. Loss function decreased significantly by implement some of these technics.

1 Introduction

1.1 Introduction to Neural Machine Translation

Neural Machine Translation (NMT) is an approach to language translation through neural network method. Traditional machine translation methods use rule-based methods, dictionary-based methods, and statistical methods. For example, by using n-grams and single word corresponding translation methods, some simple traditional machine translation methods can provide a baseline translation performance.

With the development of neural networks, linguists and computer scientists introduced neural network into machine translation area. Google announced its translation services are using neural machine translation technology in preference to previous statistical machine translation technology. Google's model is consists of a deep LSTM network with 8 encoder and 8 decoder layers using attention and residual connections.[1]

Typically a RNN known as encoder is used by the neural network to encode a source sentence for the second RNN known as a decoder to predict words in target language. To deal with the problem of long-term dependencies, Long Short Term Memory networks (LSTM) and Gated Recurrent Unit (GRU) were typically used in NMT implementation rather than vanilla RNN model. A typical encoder-decoder NMT model showed in following figure.

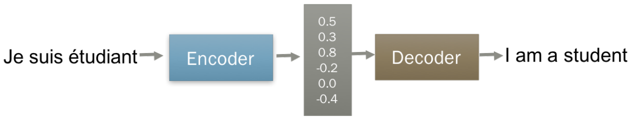

Figure 1: Encoder-Decoder model

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

1.2 Introduction to Attention Mechanism

The concept of ’attention’ has gained popularity recently in training neural networks, which allows models to learn alignments between different modalities.[2] To apply the attention mechanism onto NMT, Loung et.al [2] discussed two approaches of attention mechanisms which can be applied onto decoder layer. One is called ***global*** approach, which all source words are attended. The ***local*** one considers only a subset of source words. Typically, both mechanism works well in NMT tasks, but local approaches might boost the performance better.

In addition, both of these approaches applied onto the decoder layer, and feed the output of the encoder (not only the hidden state) to the decoder to build the attention layer. However, apply attention mechanism onto encoder is also an approach to boost NMT performance as discussed by Lin et. al.[3]

1.3 Introduction to this translation task

In this task, we use French sentences as source sentences and English sentences as target sentences. The data is downloaded from **tatoeba**. For simplicity, we used sentences with maximum length is 10 words, and we only use the sentence start with ’i am’, ’he is’, ’she is’, ’you are’, ’we are’, and ’they are’ for simplicity. In total, there are 10853 sentence pairs, with 2925 English words and 4489 French words. In the later part, some more complicated sentences will be used to show the performance of attention mechanisms.

2 Seq2Seq attempts

2.1 Basic Seq2Seq model

Seq2Seq model was first introduced by Sutskever et.al [4] Traditional Deep Neural Networks has a limitation on sequence data because DNNs require that the dimensionality of the inputs and outputs is known and fixed. However, the straightforward application of RNN architecture can solve general sequence to sequence problems. The idea is to use one RNN to read the input, and then use another RNN to extract information to a large fixed-dimensional vector representation, and then to use another RNN to extract the output sequence from that vector. The classical model of this Seq2Seq model is shown in figure 2.

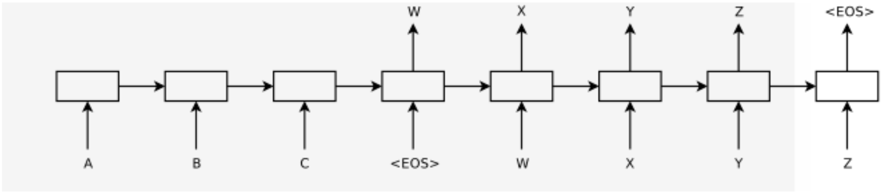

Figure 2: Basic Seq2Seq model. input sentence ’ABC’ and produce ’WXYZ’ as output sentence. When the model receive <EOS>, the second RNN use the ’fixed-dimensional’ vector and previous word to predict next word in output sentence.

The basic model consists encoding layer and decoding layer. The structure of these two layers is shown in figure 3 and 4. Both GRU and LSTM were used for this NMT task as encoder and decoder, and GRU shows better performance on this task. In addition, bi-directional GRU shows better performance than unidirectional one because it reads input from two ways, one from past to future and one from future to past. So it preserves information from both past and future, and this information is passed to the decoder. The following equation describe how does the decoder work, where x represents source sentence and y represents target sentence.

$$\text{log}p(y|x) = \sum_{j=1}^{m} \text{log}p(y_j|y < j, \mathbf{s})$$

In addition, according to training loss function, the model with Glove word embedding methods shows faster converge speed rather than the model with random embedding. And the French word embedding hurts the model performance as shown in figure 5.

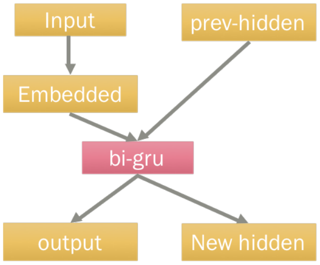

Figure 3: Structure of basic encoder

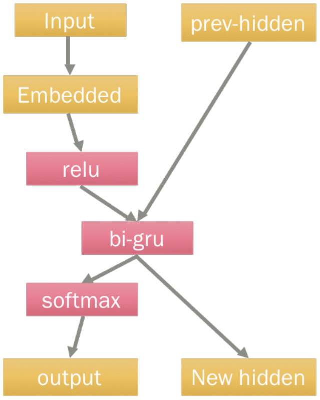

Figure 4: Structure of basic encoder

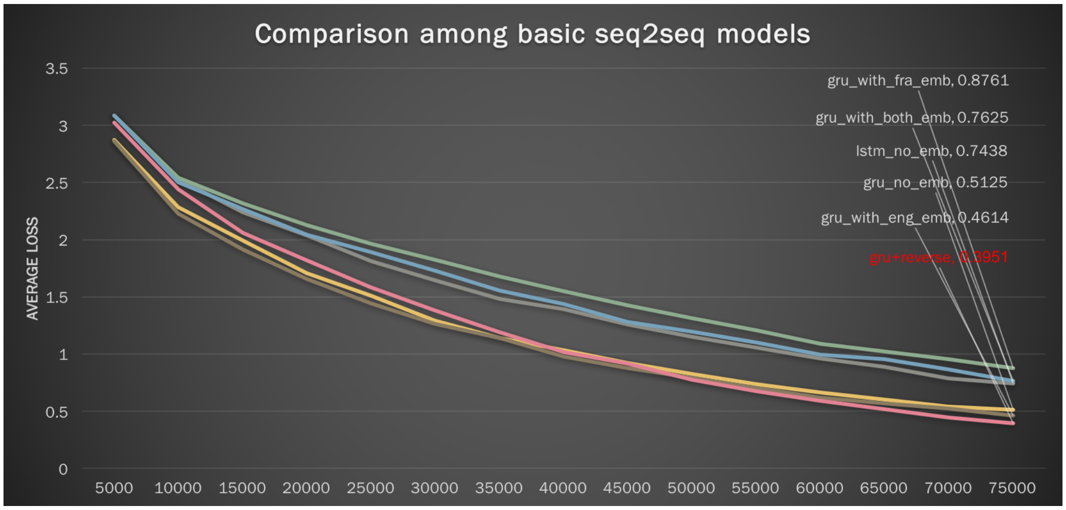

Figure 5: Comparison among basic Seq2Seq models. Bi-directional GRU with glove embedding show best performance.

2.2 Seq2Seq model with global attention mechanism in decoder

Attention mechanism was first applied to NMT task by Luong in 2015.[2] The attenion algorithm in decoder first takes the hidden state h_t from the encoder. And it takes the derived context vector c_t which captures relevant source side information to help the model to predict the current target word y_t.

In Luong's paper, two types of approach to derive context vector c_t was proposed, they are called global and local. In this section, we implement and explain the gloabl attention mechanism. The goal of global attention is by aligning target hidden state h_t and each source hidden state $\bar{h}_s$ to generate a variable length alignment vector a_t. By multipling c_t and h_t together and input this into the RNN model, the output variable can be insert into softmax for next word prediction. The detail structure of this global attention model is described in Figure 6.

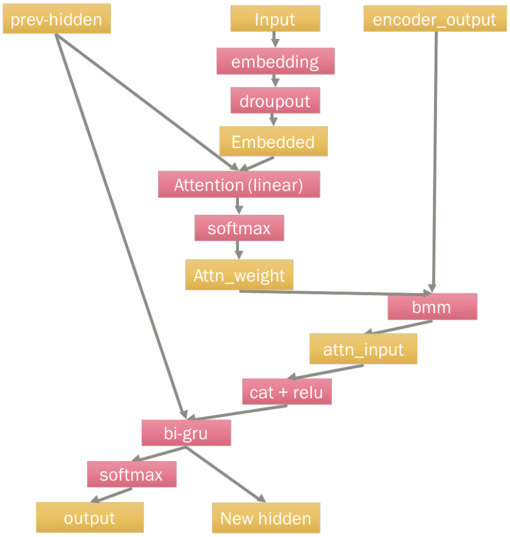

Figure 6: Structure of global attention decoder.

2.3 Seq2Seq model with self attention mechanism in encoder

Attention mechanism can also be used onto the encoder side. Lin et. al [3] proposed a new method called self-attentive sentence embedding this year. In previous attention mechanism, decoder uses encoder's output, previous target word and previous hidden state to do the prediction. However, for the attention mechanism in encoder, there is no 'encoder output'. The alternative way Lin proposed is to use the intermediate output from bi-gru. The difference is in decoder attention, attention weight is calculated, and then input into RNN model. However, in self attentive method, attention weight calcuated after RNN model. In general, this enables attention to be used in those cases when there are no extra inputs. Structure of self attention encoder is shown in Figure 7.

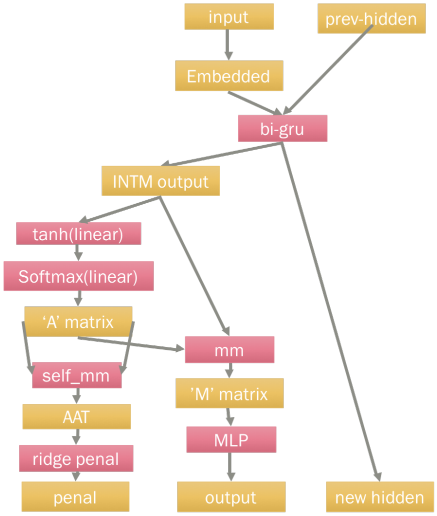

Figure 7: Structure of self-attention encoder.

In addition, this vanila model can suffer from problems if the attention mechanism always provides similar summartion weights for all the hops (attention units). Therefore, the penalized item which is modified from ridge regression model was applied to the model and cross entropy loss after 75,000 iteration has siginificant reduction from 0.959 to 0.4023. The penalized item is calculated in the model and added to loss function directly.

2.4 Seq2Seq model with attention mechanism in both encoder and decoer

To enhance the preformance on both encoder and decoder, we combined self-attention encoder and global attention decoder together. To train this model with GLOVE embedding, the loss function score has strong fluctuation. This problem might be related to the data set. The training data set is too small, and contains sentence start with 'i am', 'he is', and 'she is' etc. The trainign loss without GLOVE embedding shows smoth loss decreasing curve.

2.5 Comparison of models attempts in this project

By comparing the loss score after 75000 iterations (around 7 epochs), Seq2Seq model with both attention mechanism shows the best performance.

Model	Loss	Model	Loss
gru-noemd-256dim	0.5912	bigru+global-attn	0.3802
gru-noemd	0.5125	bigru+selfattn-10hops-nopenl	0.9595
lstm-noemd	0.7438	bigru+selfattn-10hops-penl	0.4023
gru-glove	0.4614	bigru+selfattn-7hops-penl	0.3869
bigru-glove	0.3951	bigru+selfattn-7hops-penl+global-attn	0.3552

Table 1: Comparison of loss score after 75000 iterations. Note: all but the first has 300 dimensions; bigru: bi-directional gru; selfattn: self-attentive mechanism; penl: penalization; global-attn: decoder global attention mechanism

To furtuer test the performance of this model on more complicated data, we applied both this model on to another subset of tatoeba data set. I selected pairs of sentences with length from 8 to 20 and without requirement of sentence start words. However, the bigru+selfteen-10hops-penl+global-attn model is not stable, when I applied it onto biger data, the loss function jump up to high value after 3 epoches. The reasoning is the shallow RNN structure works well on simple data which start with specific words. By applying three layers RNN structure, loss function shrinks well.

3 Summary

In this project, we implemented the basic Sequence to Sequence model; tried LSTM, GRU and bi-directional GRU as encoder and decoder; applied En and Frword embedding methods; implemented global attention model to decoder and self attentive model to the encoder. In addition, we combined all of these models together to achieve the best performance.

For the smaller data set, dual attention model with one GRU layer shows the best performance after 75000 iterations. But the improvement of attention mechanism is not very notable. However, by applying this neural network structure to some more complicated data set, the performance drop down significantly, and the loss function is hard to converge. It is because of the shallow structure. Therefore, we applied three layers structure onto the more complicated data set, and the model back to normal, with about three times longer training time. The estimated training time for 20000 complicated sentence pairs are about 5 hours.

By using the bidirectional GRU model with global attention mechanism, the model gets BLEU score 15.02.

4 Future plan

This model might be improved by using following methods:

- Train on some more complicated data such as WMT translation data set.
- For the decoder attention model, I used the concat method to calculate the alignment score. However, there are other methods such as 'dot' and 'general'.
- Figure out the reason why using French embedding hurts the performance.

References

[1] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation arXiv:1609.08144

[2] Minh-Thang Luong, Hieu Pham, Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. arXiv: 1508.04025

[3] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, Yoshua Bengio. A Structured Self-attentive Sentence Embedding. arXiv:1703.03130

[4] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Sequence to Sequence Learning with Neural Networks. arXiv: 1409.3215