Research on Neural Machine Translation Model

Mengyao Chen, Yong Li, Runqi Li
Institute of Information, Beijing University of Technology, No100 Pingleyuan, Chaoyang District, Beijing, China
chentuta123@163.com

Abstract. In neural machine translation (NMT), cyclic neural networks, especially long-term and short-term memory networks and gated recurrent neural networks, have been regarded as the latest methods for sequence modeling and transduction problems for a long time, such as language modeling and machine translation. When the cyclic neural network is running, the sequence information is processed one by one, strictly following the order from left to right or from right to left, processing one word at a time, and parallel operation cannot be realized, resulting in slow running speed. With the rapid development of neural machine translation (NMT) network architecture, cyclic neural network has been effectively replaced by convolution network and self-attention. Convolution neural network has replaced the divine circulation neural network due to its parallel computation of convolution. The Transformer model replaces the long-term and short-term memory network with a complete self-attention structure, and abandons the traditional encoder and decoder model which must combine the inherent mode of convolutional neural network or circular neural network and only uses the self-attention mechanism. Although the biggest innovation of Transformer architecture is to use full self-attention, there are several other factors, such as multi-head attention and residual connection. The model flexibly combine several common building blocks in the Transformer architecture with the cyclic neural network. By borrowing the framework of the Transformer architecture without using full self-attention, experiments show that the cyclic model can be very close to the performance of the Transformer Our model achieved 26.7 BLEU in the WMT 2014 English to German translation task and 37.8 BLEU in the WMT 2014 English to French translation task. Using these two scores alone is very close to the score of the Transformer architecture using full attention, so even if the cyclic neural network is used instead of full self-attention, it can perform well on the data set.

1. Introduction
Machine translation is that a computer can automatically translate one human language into another human language, thus realizing automatic conversion between different languages [1]. Traditional machine translation methods include rule-based machine translation, instance-based machine translation and statistics-based machine translation. With the rapid development of artificial intelligence, depth-based learning has also been widely used in the field of natural language processing. The emergence of neural machine translation (NMT) has significantly improved the quality of machine translation, thus becoming the mainstream machine translation method in the current industry.

The mainstream model framework of neural machine translation is encoder - decoder model. The model uses two different neural networks as encoder and decoder respectively. The encoder is used to
read the source sentence, encode it into a vector of fixed dimension, and then the decoder reads the vector and generates the corresponding target language sequence.

The neural networks used in the current mainstream encoder and decoder models are cyclic neural networks, their variants and convolutional neural networks. However, when the input sentence is long, the vector with fixed dimensions is difficult to store enough information, so the academic circle introduces an attention mechanism to solve this problem. The attention mechanism allows the decoder to look up words or fragments of the input sentence at any time, so it is not necessary for the intermediate vector to store all information.

Then Ashish Vaswani proposed the Transformer model, which uses self-Attention, i.e. the model is completely dependent on the attention mechanism. Through the Attention layer, global connections can be captured one step at a time without obtaining step by step, thus leading to significant acceleration of training time and improvement of quality. Transformer includes other differences besides self-attention, including layer standardization of the whole model, multi-source attention mechanism, multi-head attention mechanism, residual connection and the use of feed-forward layer.

This paper propose to introduce the multi-head attention, residual connection and feed-forward network into the cyclic neural network in the Transformer architecture, and the cyclic neural network can also be very close to the Transformer model in performance.

2. Related Work
In 2014, Ilya Sutskever et al. proposed an end-to-end sequence learning method that uses a multi-layer LSTM to map an input sequence to a vector of fixed dimensions and then another depth LSTM to decode the target sequence of the vector [2]. This can be understood as

In 2017, Jonas Gehring et al. proposed a fully convoluted sequence-to-sequence model that replaces the cyclic neural network originally used in encoders and decoders, while equipped with gated linear elements and residuals in the model. Connected and introduced the attention mechanism [3]. Compared with RNN, CNN framework accelerates the operation speed by multi-level convolution operation on the source and target sentence sequences, and better handles and extracts the complex relationship information in the sentence structure. The sequence-dependent structure of RNN itself is quite unfriendly for massively parallel computing, because the calculation of T time depends on the hidden layer calculation result at time T-1, and the calculation of T-1 time depends on the hidden layer calculation at time T-2. As a result, so-called sequence dependencies are formed, which makes it difficult for RNNs to have efficient parallel computing capabilities.

Attention mechanisms have become an integral part of the compelling sequence modeling and transformation models of various tasks, allowing modelling of dependencies regardless of their distance in the input or output sequence [6, 12]. In a few cases [13], this attention mechanism is used in conjunction with a circular network.

In the same year, Ashish Vaswani et al. proposed a network structure called Transformer in the paper "Attention is all you need", which completely relies on the attention mechanism and completely abandons the recursive structure [4], in addition to multiple Attention focused network. Similar to CNN, Transformer sets the maximum length of the input. If the sentence is not so long, it is padded with Padding. However, due to the fixed context length of the Transformer model, the model cannot capture any long-term dependencies that exceed the predefined context length. But so far, Transformer is still one of the best performing models on multiple data sets.

3. Model

3.1. Transformer
Any text or material outside the aforementioned margins will not be printed.

The sequence model of most neural networks has an encoder-decoder structure [2][5][6]. The encoder maps the input sequence of the symbolic representation \((x_1, \ldots, x_n)\) to the sequence of consecutive representations \(z = (z_1, \ldots, z_n)\). Given \(z\), the decoder then generates the output
The sequence \((y_1, \ldots, y_m)\) symbol is one element at a time. In each step, the model is autoregressive [7], consuming the previously generated symbols as additional inputs when generating the next one. Transformer follows this overall architecture, using a stacking layer of self-focused and fully point-by-point encoders and decoders. As shown in Figure 1.

The encoder is made up of the same 6 modules, each of which has two sublayers. The first sub-level is the Multi-Head self-attention mechanism, where self-attention indicates that both the input and output sequences are the same. The second sub-level uses a fully connected network, the main role is to pay attention to the characteristics of the sub-level. In addition, each sub-level adds a residual connection and hierarchical normalization.

The decoder is also composed of the same six modules. Each decoder module has three sub-layers. Each sub-layer also adds residual connection and hierarchical normalization. The first and third sublayers are identical to the encoder's Multi-Head Self-Attention Layer and the Full Connection Layer, respectively, and the Multi-Head Attention mechanism used by the second sub-layer uses the encoder's output as Key and Value. The output of the first sublayer of the decoding module is used as the Query.

Figure 1: The Transformer model architecture

3.2. The RNN Model

Cyclic neural network is mainly used to process sequence data, especially for variable length sequence data [29]. Most of the neural machine translation uses cyclic neural network to realize the original encoder decoder for the cyclic neural network, and improve Long-term and short-term memory networks, gated cyclic neural networks, for sequence modeling. Cyclic neural network models typically take into account the calculation of the symbol positions along the input and output sequences. Aligning the positions with the steps in the calculation time, they generate a series of hidden states \(h_t\) as a function of the inputs of the previously hidden states \(h_{t-1}\) and position \(t\).

The input \(x\) is mapped to the output \(O\) by the cyclic neural network, \(y\) is the target sequence to be reached by the model (usually given by the training corpus), \(L\) is the loss function, \(U\) is the weight
matrix input to the hidden layer, and W is the hidden layer to the hidden The weight matrix of the layer, V is the weight matrix of the hidden layer to the output, and the time series t ranges from [1, T]. The entire network is updated as follows:

\[ \alpha_t = Wh_{t-1} + Ux_t + b \]  \hspace{1cm} (1)
\[ h_t = \tanh(\alpha_t) \]  \hspace{1cm} (2)
\[ O_t = Vh_t + c \]  \hspace{1cm} (3)
\[ y_t = \text{softmax}(O_t) \]  \hspace{1cm} (4)

After the cyclic neural network is unfolded, it can be trained by backpropagation algorithm, which is called time back propagation. In practical applications, gradient disappearance problem will occur [9]. The long-and-short-time memory neural network [10] is a deformed structure of the cyclic neural network, which adopts a more efficient mechanism of forgetting and updating, has similar structures and advantages as the cyclic neural network, and has better performance. The threshold loop unit [11] merges the input gate and the forgetting gate of the long and short memory loop unit into an update gate, and introduces a reset gate. The update gate is used to control the current state and the forgotten history information and the accepted new information are reset. How much information in the gate control candidate state is obtained from the historical information. This structure is a simplification of the long and short memory neural network, the effect is similar to the latter, and the calculation amount is reduced.

3.2.1 Embedding
Each word in the input layer is represented by a real vector, which is called "word embedding." Word embedding can be understood as embedding a vocabulary into a real space of a fixed dimension. Converting a word number into a word vector reduces the input dimension, reduces the parameters and statistics of the cyclic neural network, and the word vector converts the sparse number into a vector of chips, so that the word vector can contain more information.

3.2.2 Encoder and Decoder
The encoder consists of three parts, input x, hidden state h, output y. The encoder reads the input \( x = (x_1, \ldots, x_T) \) and encodes it as a hidden state \( h = (h_1, \ldots, h_T) \) when using a cyclic neural network:

\[ h_t = f(x_t, h_{t-1}) \]  \hspace{1cm} (5)
\[ c = q([h_1, \ldots, h_T]) \]  \hspace{1cm} (6)

\( c \) is the source language sentence representation, \( f \) and \( q \) are nonlinear functions.

The task of the decoder is to obtain the maximum probability of the next word in the wood. The target language term \( y_t \) is generated in a given language representation \( c \) and the precursor output \( \{y_1, \ldots, y_{t-1}\} \), as defined below:

\[ P(y) = \prod_{t=1}^{T} p(y_t|y_1, \ldots, y_{t-1}, c) \]  \hspace{1cm} (7)

The Encoder-Decoder model is a general framework that can be implemented by different neural networks, such as long and short memory neural networks, gated cyclic neural networks.

3.3. RNN To Transformers
The difference between Transformer and RNN includes multiple source attention layers, multi-head attention, layer normalization and the residual upscaling feed-forward layers. The modules in our Transformer model introduce a cyclic neural network to improve the circulating neural network.

The encoder consists of a layer of bi-RNN and a stack of six identical layers. Of the six identical layers, each layer is divided into two sub-layers, the first sub-layer is the RNN network, and the second sub-layer is a simple, fully connected feedforward network. The paper use residual connections around both sublayers and then normalize the layers.
3.3.1 RNN
Here, the cyclic neural network can use LSTM [6] or GRU [15]. In addition, Bi-RNN can be used. Bi-RNN runs one RNN in the forward direction and another RNN in the other direction, and connects the two results. This article uses Bi-RNN. One of the benefits of using an RNN network is the ability to learn the sequence of sequences so that positional embedding can be used instead.

3.3.2 Feed-Forward Networks
Each of our encoders and decoders contains a fully connected feedforward network that is applied to each location separately and identically. This includes two linear transformations with RELU activation in between. Although linear transformations are identical at different locations, they use different parameters between layers. Another way to describe this situation is to understand it as two convolutions with a kernel size of one. The dimensions of the input and output are $\sqrt{d_{model}=512}$.

3.3.3 Attention
Inputs consist of the query and key of dimension $d_k$ and the value of dimension $d_v$. The model use all the keys to calculate the point product of the query, divide each term by $\sqrt{d_k}$, and apply the Softmax function to get the weight of these values. In practical applications, The model compute the attention function of a set of queries at the same time and package them into a matrix Q. The keys and values are also packed into matrices K and V. The calculation of the output matrix is:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V$$  \hspace{1cm} (8)

3.3.4 Multi-Head Attention
Multi-Head Attention allows the linear projection of queries, keys and values, h times to the dk, dk and dv dimensions, respectively. Then, on each of these projected versions of the query, keys, and values, The model execute the attention function in parallel to generate the dv-dimensional output value. They are connected in series and are projected again to get the final value, which is calculated as h=8.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h) W^O$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

3.3.5 Layer normalization and residual connection
Layer normalization [16] is normalized using mean and standard deviation. Calculated as follows:

$$\text{Layer}(x_t) = \frac{d}{\theta_t} (x_t - \theta_t) + b$$  \hspace{1cm} (9)

$$\theta_t = \frac{1}{d} \sum_{j=1}^{d} x_{t,j}$$  \hspace{1cm} (10)

$$\theta_t = \frac{1}{d} \sum_{j=1}^{d} (x_{t,j} - \theta_t)$$  \hspace{1cm} (11)

The residual layer adds the output of any layer 1 to the current hidden state, which is defined as:

$$\text{res}(x_t, l) = x_t + l(x_t)$$  \hspace{1cm} (12)

The decoder consists of six identical layer stacks, one layer of attention and one layer of feedforward network. In addition to the two sublayers in each encoder layer, the decoder uses two additional sublayers that perform multi-head attention on the output of the encoder stack. Similar to the encoder, The model use a residual connection around each sublayer and then normalize the layer.
4. EXPERIMENTS

4.1. Environment and Data
The experimental hardware environment is Ubuntu16.04+cuda8.0+GTX1080. This paper used TensorFlow as a deep learning framework. Two datasets are used in this article. WMT'14 English-German dataset, in this dataset we use 4.5M sentence pairs for training. The other dataset is WMT'14 English-French, The model use the complete training set of 36M sentence pairs, delete sentences with more than 175 words and pairs with source/target length ratios over 1.5, so we will have sentences of 35.5M in size. Pair training.

4.2. Evaluation
The method of automatic evaluation of quality in the field of machine translation is BLEU [14]. BLUE is a text evaluation algorithm, which is used to evaluate the correspondence between machine translation and professional human translation. The core idea is that the closer the machine translation is to the professional human translation. The better the quality, the score obtained by the BLUE algorithm can be used as one of the indicators of machine translation quality. Co-occurrence frequency. BLEU is essentially the calculation of the co-occurrence frequency of two sentences, but the use of some techniques in the calculation process, the value of the pursuit of calculation can measure the degree of agreement between the two sentences, the formula is as follows:

$$\text{BLEU} = \text{BP} \times \exp \left( \sum_{n=1}^{N} w_n \log p_n \right)$$  \hspace{1cm} (13)

4.3. Result
This paper use the BLEU score to assess translation quality. The results are shown in Table 1. It can be seen that in the WMT 2014 English-German translation task, our best result was obtained by Transformer. The score reached RNN to Transformer. The translation did not exceed Transformer, but the results were quite close, only the difference. 0.9 BLEU, and significantly better than RNN-based
translation. In the WMT 2014 English-French translation task, our RNN to Transformer model achieved a BLUE score of 37.8, which is far superior to the pure RNN translation. The score difference with the Transformer model is 0.3 BLEU, and the effect is very close. The results show that even if you do not use full self-attention, and use the cyclic neural network, it can perform well according to the corresponding improvement.

Table 1: The Experiments result

| MODEL                 | EN - DE | EN - FR |
|-----------------------|---------|---------|
| RNN                   | 25.16   | 35.1    |
| TRANSFORMER           | 27.6    | 38.1    |
| RNN TO TRANSFORMER    | 26.7    | 37.8    |

5. Conclusion
In this article, we propose to improve the location information by modeling the multi-head attention, residual connection, feedforward network and cyclic neural network in the Transformer architecture by parasitizing the RNN into the Transformer model. The problem. The model have achieved good results on WMT 2014 English to German and English to French machine translation tasks. The results show that even if we do not use fully self-focused models, The model can perform well and make some construction of the model. If we adjust, we believe that This model will get better results.

It should be noted that although the introduction of RNN into the Transformer model has improved the translation effect of the original model, it also brings some problems of the RNN itself, such as parallel computing, which is also needed for our future work.

Acknowledgments
we express our sincere gratitude to Teacher Li Yong for his help in the process of writing the thesis.

References
[1] Zhang Yankun, Liu Yanghua, et al. Experimental study on flexural bearing capacity of profiled steel sheet-light-weight aggregate concrete floor slabs[J]. Industrial Construction, 2008, 38(8):83-85(in Chinese).
[2] LIU Q. Syntax-based Statistical Machine Translation Models and Approaches[J]. Journal of Chinese Information Processing, 2011,25(6):63-7.
[3] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In Proceedings of NIPS 2014. (Citation: 5,452).
[4] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. In Proceedings of ICML 2017. (Citation: 453).
[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In Proceedings of NIPS 2017. (Citation: 1,057).
[6] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, abs/1406.1078, 2014.
[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.
[8] Alex Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.
[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
[10] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Transactions on Neural Networks, 2002, 5(2):157-166.

[11] Hochreiter S, Schmidhuber, Jürgen. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[12] Chung J, Gulcehre C, Cho K H, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. Eprint Arxiv, 2014.

[13] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In International Conference on Learning Representations, 2017.

[14] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention based neural machine translation. arXiv preprint arXiv:1508.04025, 2015.

[15] Papineni, K, Roukos, S, Ward, T and Zhu, W. J. BLEU: a method for automatic evaluation of machine translation. (2002).

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation.

[17] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450.