Comparison between Convolutional Sequence to Sequence Learning and Bidirectional Long Short-term Memory in English-Chinese Neural Machine Translation

|  |
| --- |
| Hong Xu |
| Georgetown University |
| hx67@georgetown.edu |

|  |  |
| --- | --- |
|  |  |

Abstract

Machine translation between Chinese and English is one of the most popular tasks in the research of Natural Language Processing. Currently, the most neural machine translation (NMT) models follow the Recurrent Neural Networks (RNN) framework and a big pain point for any RNN model training is that they are very time consuming. This project focus on comparing the performance of traditional Bidirectional RNN Models and the Convolutional Sequence to Sequence Learning (Fairseq). The result shows that the Fairseq model could yielding comparable results with much shorter training time.

Introduction

Machine translation is one of the important research directions in natural language processing. In recent years, neural machine translation methods have surpassed traditional statistical machine translation methods in translation performance of most of languages and have become the mainstream methods of machine translation. Among all the NMT models, sequence to sequence learning has been successful in many tasks, especially machine translation. The dominant approach to date encodes the input sequence with a series of bi-directional recurrent neural networks (RNN) and generates a variable length output with another set of decoder RNNs, both of which interface via a soft-attention mechanism (Bahdanau et al., 2014; Luong et al., 2015). In machine translation, this architecture has been demonstrated to outperform traditional phrase-based models by large margins (Sennrich et al., 2016b; Zhou et al., 2016; Wu et al., 2016).

Convolutional neural networks (CNN) are less common for sequence modeling in current researches, despite several advantages (Waibel et al., 1989; LeCun & Bengio, 1995). Compared to the recurrent layers, CNN create representations for fixed size contexts, however, the effective context size of the network can easily be made larger by stacking several layers on top of each other. This allows to precisely control the maximum length of dependencies to be modeled. Convolutional networks do not depend on the computations of the previous time step and therefore allow parallelization over every element in a sequence. This contrasts with RNNs which maintain a hidden state of the entire past that prevents parallel computation within a sequence.

Before 2017, many works have applied CNNs to sequence modeling such as Bradbury et al. (2016) who introduce recurrent pooling between a succession of convolutional layers or Kalchbrenner et al. (2016) who tackle neural translation without attention. However, none of these approaches has been improved the results on large benchmark datasets. Gated convolutions have been previously explored for machine translation by Meng et al. (2015) but their evaluation was restricted to a small dataset and the model was used in tandem with a traditional count-based model. Architectures which are partially convolutional have shown strong performance on larger tasks, but their decoder is still recurrent (Gehring et al., 2016).

After these attempts, Facebook AI Research introduced the first fully convolutional model (Gehring et al. 2017) for sequence to sequence learning that outperforms strong recurrent models on very large benchmark datasets at an order of magnitude faster speed. The model was tested with English-Romanian, English-German, and English-French translation tasks. In this paper, I use the dataset **WMT’17** English-Chinese to test its performance and compare it with the traditional model based on the Long Short-term Memory (LSTM) model.

Dataset

News Commentary V12, from WMT17, is used in this paper. News Commentary V12 contains more than 227K parallel sentence pairs for 12 languages. The paper focus on the English-Chinese translation task. In the dataset, 3000 pairs are randomly selected as the test set and the rest are used to train the model.

* 1. Data Cleaning

The data cleaning process includes the following process:

**Missing Values and Wrong Tokens:** I first removed all those sentences where both English and Chinese versions are both missing. Those sentences which are not either English or Chinses and the corresponding sentences are also removed from the dataset. In addition, all wrong symbols and tokens, for example HTML markups and non-breaking white space, are removed.

**Word Token:** I tokenize the sentences in the dataset, using Python package NLTK for English and Python package Jieba for Chinese word segmentation.

**Casing:** I remove cases from English and converted all string to lower case.

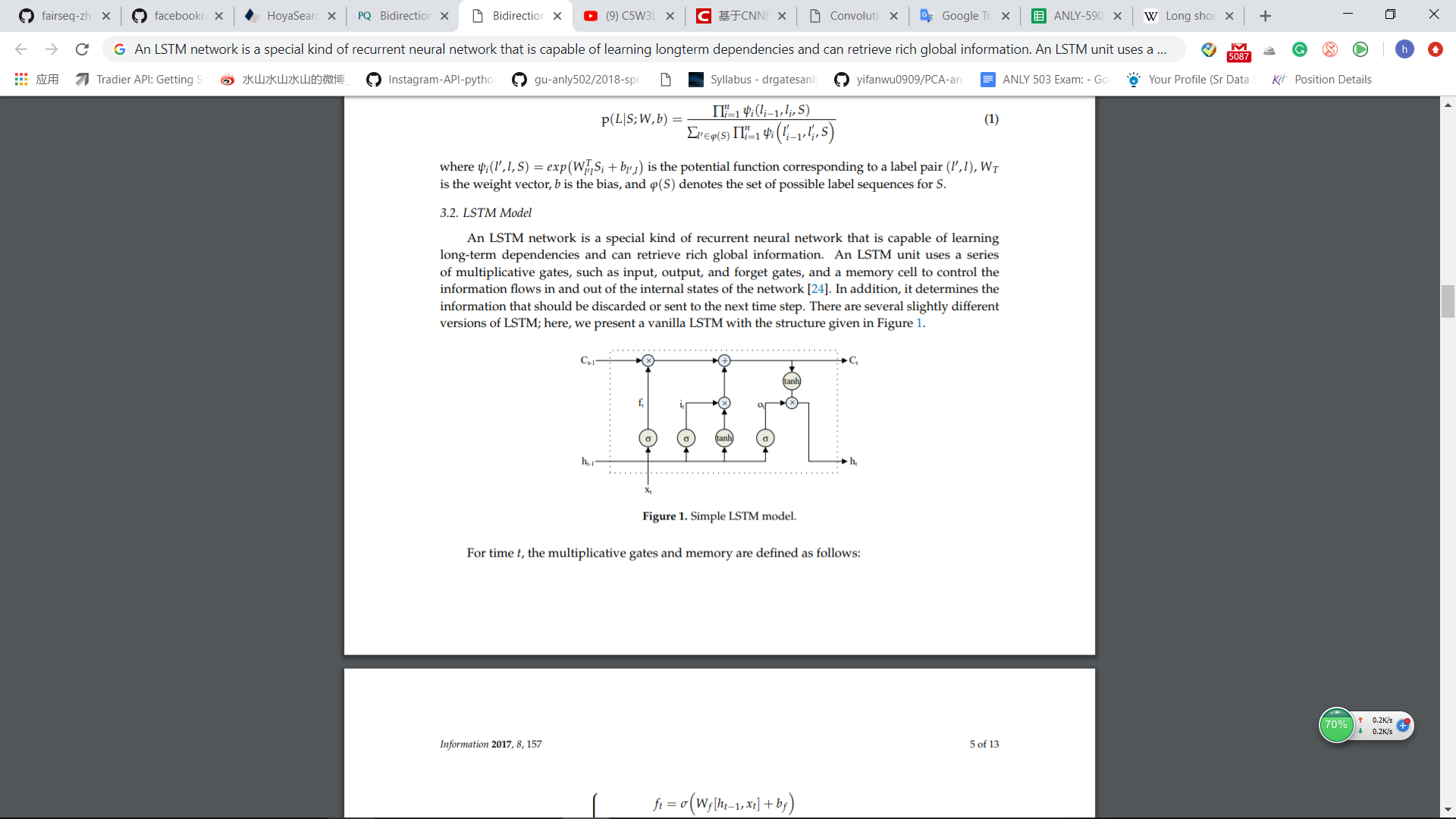
**Merge blank lines:** I note that dataset often has blank lines. In some cases, these problems are caused by formatting errors, but there are also some cases where a long English sentence is translated to several Chinese sentences. This appears as a sentence followed by blank line on the English corpus. To deal with this, we merge the Chinese sentences onto same line, and then remove the blank line from both English and Chinese corpuses.

Recurrent Neural Network

This paper focus on the comparison between the Bidirectional LSTM (Bi-LSTM) and CNN sequence to sequence learning. This section will introduce the Bi-LSTM briefly.

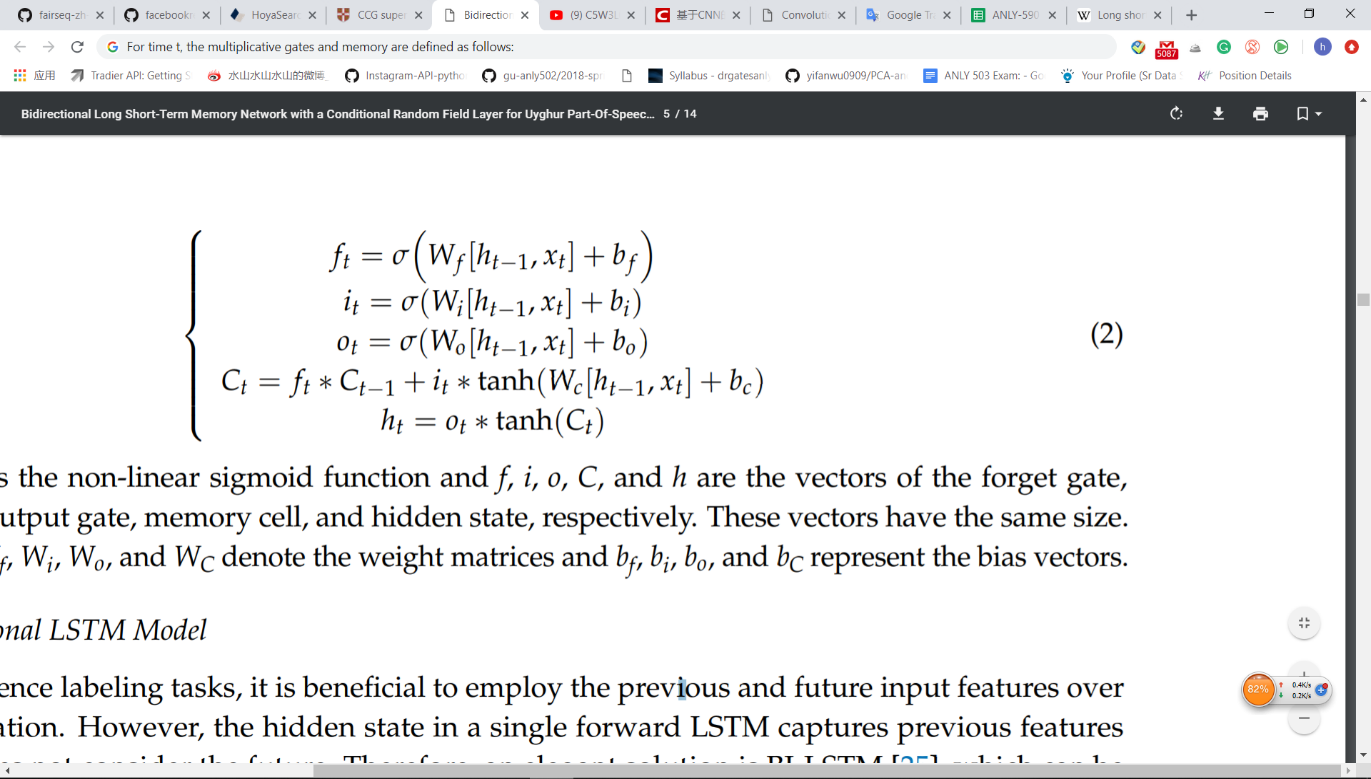
* 1. LSTM Model

An LSTM network is a special kind of recurrent neural network that is capable of learning the long-term dependencies and can retrieve rich global information. An LSTM unit uses a series of multiplicative gates, such as input, output, and forget gates, and a memory cell to control the information flows in and out of the internal states of the network. In addition, it determines the information that should be discarded or sent to the next time step. There are several slightly different versions of LSTM; here, we present a vanilla LSTM with the structure given in Figure 1:



**Figure 1:** Simple LSTM Model

For time *t*, the multiplicative gates and memory are defined as follows:



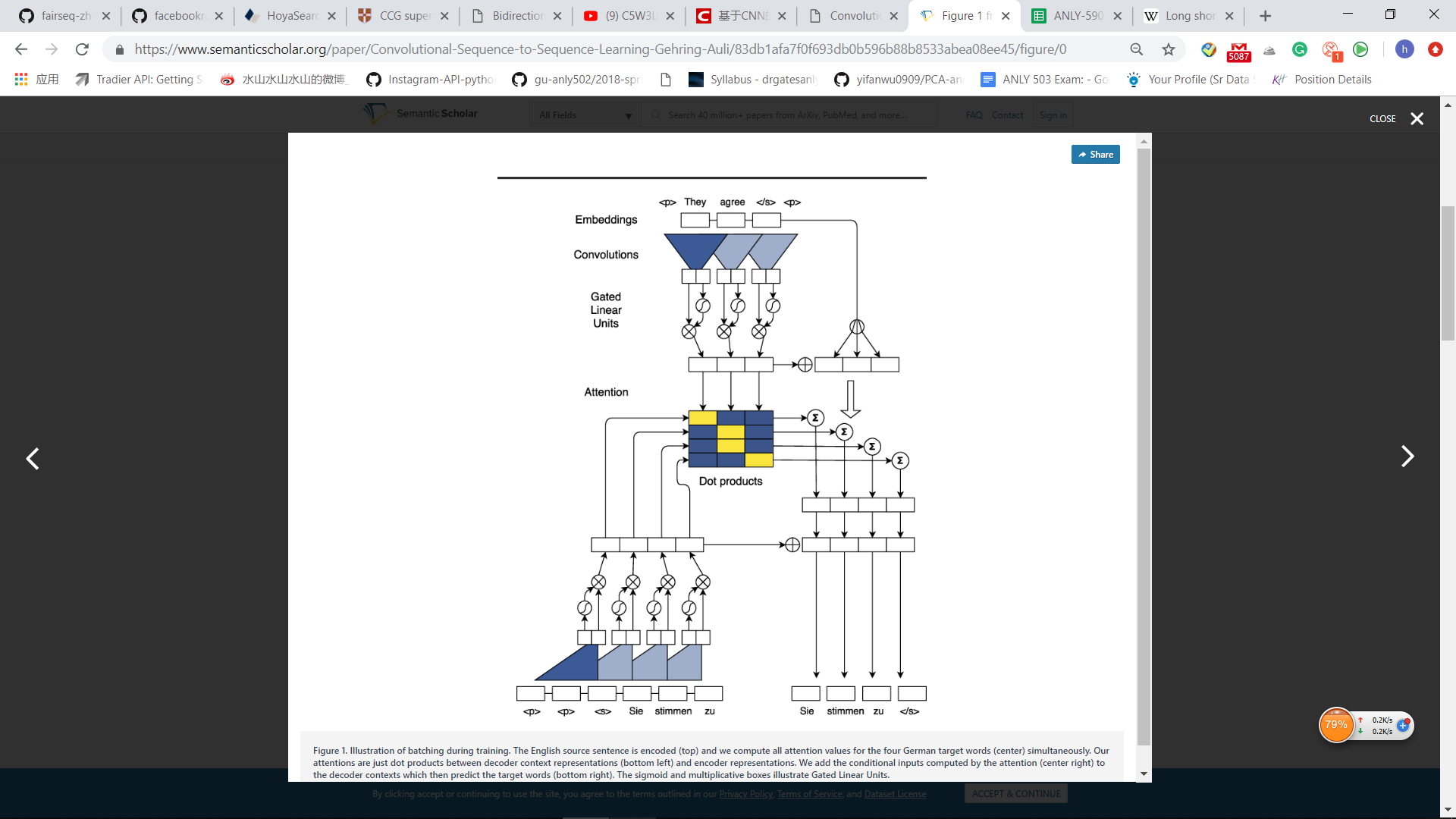
where is the non-linear sigmoid function and , and are the vectors of the forget gate, input gate, output gate, memory cell, and hidden state, respectively. These vectors have the same size. Moreover, , and denote the weight matrices and , and represent the bias vectors.

* 1. Bidirectional LSTM Model

In neural machine translation tasks, it is beneficial to employ the previous and future input features over a given duration. However, the hidden state in a single forward LSTM captures previous features only and does not consider the future. Therefore, an elegant solution is Bi-LSTM, which can be regarded as a stack of two LSTM layers. The previous features are extracted by a forward LSTM layer, and the future features are captured by a backward LSTM layer. In this way, we can effectively utilize the previous and future features.

1. CNN Sequence to Sequence Learning

This section is going to introduce the CNN Architecture based on the Facebook AI Research report (Gehring et al. 2017). This approach relies in a fully CNN architecture. This substitutes the RNNs in computing the intermediate encoder states and decoder states .



**Figure 2:** CNN Sequence to Sequence Learning Architecture

* 1. Position Embeddings

First, the model embeds input elements in distributional space as , where is a column in an embedding matrix . They also equip our model with a sense of order by embedding the absolute position of input elements where . Both are combined to obtain input element representations . They proceed similarly for output elements that were already generated by the decoder network to yield output element representations that are being fed back into the decoder network . Position embeddings are useful in our architecture since they give our model a sense of which portion of the sequence in the input or output it is currently dealing with.

* 1. Convolutional Block Structure

The block structure of this convolutional neural network is a simple one. It computes intermediate states based on a fixed number of input elements. Each block contains a one-dimensional convolution followed by a non-linearity. The non-linearities chosen in this research were the so-called Gated Linear Units (GLUs). They implement a simple gating mechanism over the output of the convolution

* 1. Multi-step Attention

They introduce a separate attention mechanism for each decoder layer. To compute the attention, we combine the current decoder state with an embedding of the previous target element :

For decoder layer the attention of state and source element is computed as a dot-product between the decoder state summary and each output of the last encoder block :

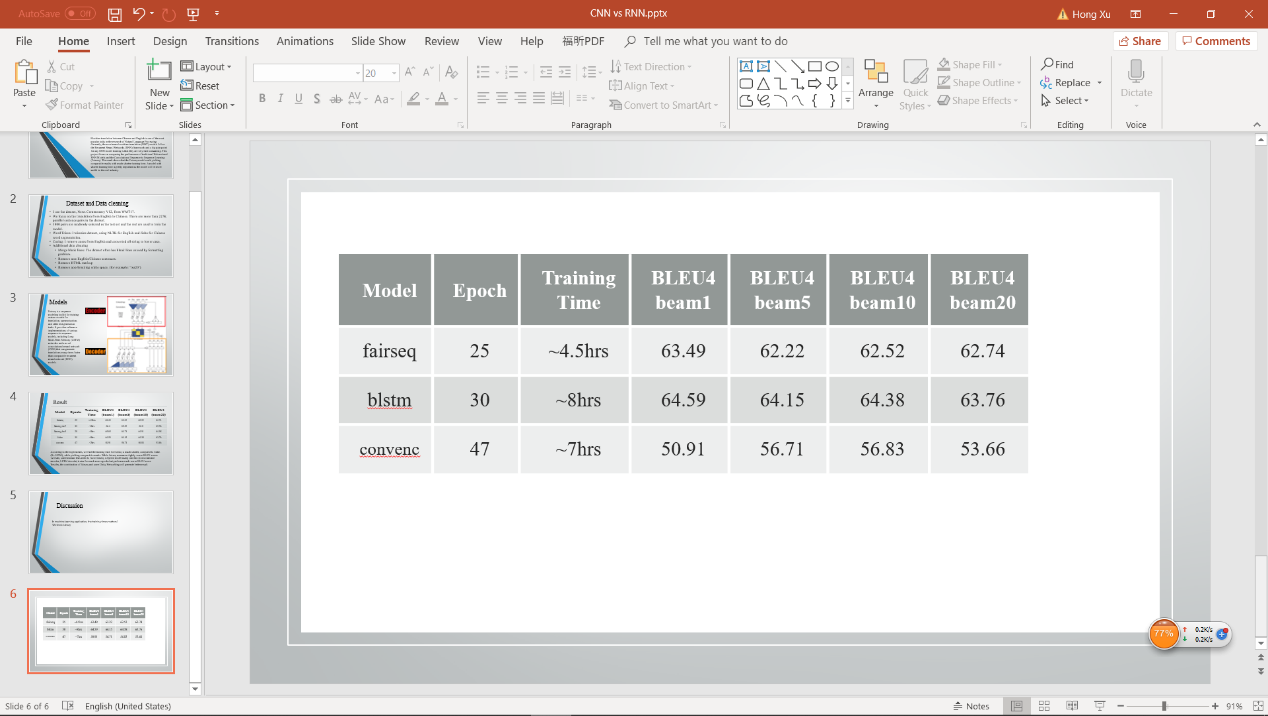
The conditional input to the current decoder layer is a weighted sum of the encoder outputs as well as the input element embeddings (Figure 2, center right):

* 1. More

In the report (Gehring et al. 2017) from Facebook AI Research, the writers also mention the Normalization Strategy and Initialization in the CNN Sequence to Sequence architecture. Please read the original paper for details.

1. Experiment Results

To test the models with the task of English-Chinese machine translation, I used the Bi-LSTM, CNN Sequence to Sequence Learning (fairseq, which represents the Facebook AI Research Sequence), and a hybrid model using the Convolutional encoder and LSTM decoder. The result includes the bilingual evaluation understudy (BLUE) score, which is a commonly used metric to evaluate translation performance, and the training time. It should be noted that I also tried different size of beam search in the test. The experiment results are as follow:



**Table 1.** Experiment Result

According to the experiments, we find the training time for CNN Sequence to Sequence model is much shorter, compared to the Bi-LSTM, while yielding comparable results. While CNN Sequence to Sequence model measures slightly worse BLEU scores than the Bi-LSTM, some manual tests seem to favor CNN Sequence to Sequence model. The hybrid model trains for much more epochs but performs much worse BLEU score. In conclusion, we find that the CNN Sequence to Sequence Learning from Facebook AI Research perform well on the English-Chinese machine translation task and its faster speed make the model more useful in the real-world application.

References

Bradbury, James, Merity, Stephen, Xiong, Caiming, and Socher, Richard. 2016. Quasi-Recurrent Neural Networks. arXiv preprint arXiv:1611.01576

Cho, Kyunghyun, Van Merrienboer, Bart, Gulcehre, ¨ Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proc. of EMNLP,

Gehring, Jonas, Auli, Michael, Grangier, David, and Dauphin, Yann N. 2016. A Convolutional Encoder Model for Neural Machine Translation. arXiv preprint arXiv:1611.02344

Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. 2017. Convolutional Sequence to Sequence Learning

Kalchbrenner, Nal, Espeholt, Lasse, Simonyan, Karen, van den Oord, Aaron, Graves, Alex, and Kavukcuoglu, Koray. 2016. Neural Machine Translation in Linear Time. arXiv

LeCun, Yann and Bengio, Yoshua. 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):1995

Luong, Minh-Thang, Pham, Hieu, and Manning, Christopher D. 2015. Effective approaches to attention-based neural machine translation. In Proc. of EMNLP

Meng, Fandong, Lu, Zhengdong, Wang, Mingxuan, Li, Hang, Jiang, Wenbin, and Liu, Qun. 2015. Encoding Source Language with Convolutional Neural Network for Machine Translation. In Proc. of ACL

Sennrich, Rico, Haddow, Barry, and Birch, Alexandra. 2016 Neural Machine Translation of Rare Words with Subword Units. In Proc. of ACL

Waibel, Alex, Hanazawa, Toshiyuki, Hinton, Geoffrey, Shikano, Kiyohiro, and Lang, Kevin J. 1989. Phoneme Recognition using Time-delay Neural Networks. IEEE transactions on acoustics, speech, and signal processing, 37 (3):328–339

Wu, Yonghui, Schuster, Mike, Chen, Zhifeng, Le, Quoc V, Norouzi, Mohammad, Macherey, Wolfgang, Krikun, Maxim, Cao, Yuan, Gao, Qin, Macherey, Klaus, et al. 2016. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv preprint arXiv:1609.08144

Zhou, Jie, Cao, Ying, Wang, Xuguang, Li, Peng, and Xu, Wei. 2016. Deep Recurrent Models with Fast-Forward Connections for Neural Machine Translation. arXiv preprint arXiv:1606.04199