

The Experiment Report of Machine Learning

SCHOOL: SCHOOL OF SOFTWARE ENGINEERING

SUBJECT: SOFTWARE ENGINEERING

|  |  |
| --- | --- |
| *Author:*  San Zhang | *Supervisor:*  Mingkui Tan |
|  |  |
| *Student ID:*  202010100001 | *Grade:*  Undergraduate |

2020-10-24

# Chinese-English Translation Machine Based on Sequence to Sequence Network

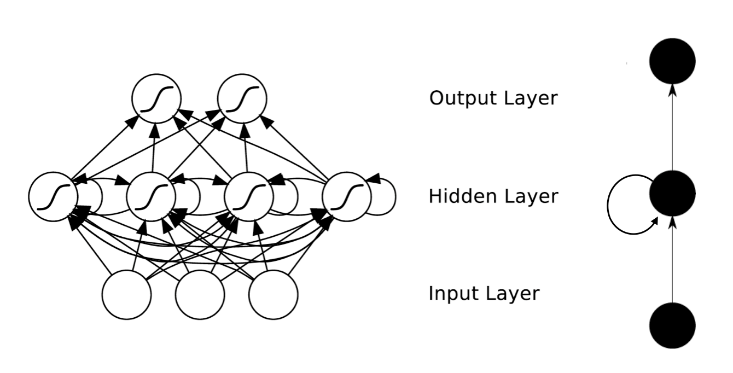
***Abstract—In this experiment, we try to build a classic Sequence-to-Sequence machine translation model with the application of attention mechanism, verifying model performance on simple and small-scale dataset.***

Figure. 1. The structure of RNN

## INTRODUCTION

Machine translation, is the process of using a computer to convert one natural language (source language) into another natural language (target language). It is a branch of computational linguistics, one of the ultimate goals of artificial intelligence, and has important scientific research value. At the same time, machine translation has important practical value.

Our experiment aims to translate Chinese sentences to English with a sequence to sequence network and attention. Since it is too complex and hard work for a personal computer to realize the commercial-grade machine translation with RNN, our experiment only trains on a small corpus based on a small -scale dataset and the evaluate it with BLEU. And continuously improve our model and increase the accuracy and quality of translation results by evaluating the merits and demerits of translated sentences,

## METHODS AND THEORY

1. Recurrent Neural Network (RNN)

A Recurrent Neural Network, or RNN, is a network that operates on a sequence and uses its own output as input for subsequent steps. Compared to traditional convolution Neural networks (CNNs), the current output of a sequence is related to the previous output in RNN. The specific form is that the network will remember the previous information and apply it to the calculation of the current output, that is, the nodes between the hidden layers are no longer unconnected but connected, and the input of the hidden layer not only includes the output of the input layer It also includes the output of the hidden layer at the previous moment. Since the translation belongs to a task of sequence modeling, where the inputs and output have a sequential dependence, our experiment is based on RNNs to gives the network a kind of 'memory' function to the front content like human cognition.

The RNN is composed of input layer, hidden layer, output layer. In addition, there is an arrow in the hidden layer to indicate the cyclic updating of data. This is the way to realize the time memory function.

We can model the recurrence formula by a sequence of vectors at evert time step:

And a RNN cell at time step t:

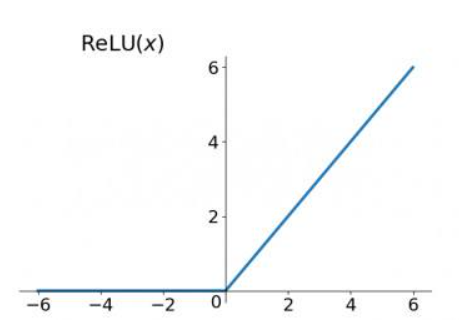
ReLU and SoftMax is the activation function we used in our experiment (other activation function can be used instead). is the hidden node and is the output at time step t.

Figure. 2. The ReLU function

And by forwarding the propagation of RNN cells we predict the output . In addition, Cross entropy function NLLLoss is used as the loss function:

And the total loss is:

The parameters in the model are adjusted by Backpropagation Through Time (BPTT), which pass the error value of the output in the reverse direction and update it with the gradient descent:

We find that the cumulative multiplication will lead to the cumulative multiplication of the derivative of the activation function, which may lead to the "gradient disappear" and "gradient explosion" phenomenon.

Since the derivative range of sigmoid function is (0,0.25), and that of tanh function is (0,1), whose maximum derivative is no more than 1, the result of multiplication is smaller and smaller if sigmoid function is taken as the activation function in the process of the above formula multiplication. With the development of time series, the cumulative multiplication of decimals will cause the gradient to become smaller and smaller until it is close to 0, which is the phenomenon of "gradient disappearing". So is the tanh function. That’s why we choose ReLU as activation function before, whose derivative is always 1.

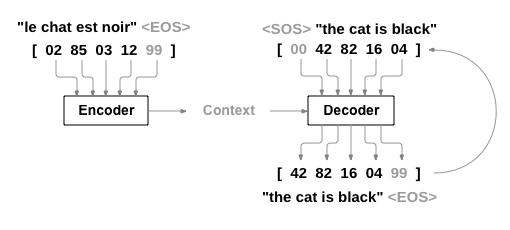
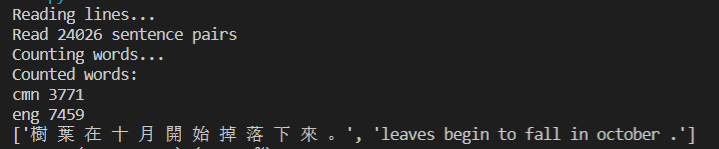
1. Encoder-Decoder Architecture

Figure. 3. Encoder-Decoder Architecture

A [Sequence to Sequence network](https://arxiv.org/abs/1409.3215), or seq2seq network, or [Encoder Decoder network](https://arxiv.org/pdf/1406.1078v3.pdf), is a model consisting of two RNNs called the encoder and decoder. The encoder expresses the source language into a high-dimensional vector (context vector) after a series of neural network transformations. And the decoder is responsible for re-decoding (translating) this high dimensional vector into the target language.

In the training process, the decoder will get an input data and a hidden layer state data. The initial input data is a placeholder < SOS > marking the beginning of a sentence, and the initial hidden layer state data is the context vector, that is, the last hidden layer state data from the encoder.

1. Attention

The long-term dependencies in traditional encoder-decoder, which compress all the necessary information of a source sentence into a static context carrying the burden of encoding the entire sentence, can make the translation effect deteriorate rapidly as the length of text increases.

Attention, using direct connection to the encoder to focus on a particular part of the source sequence, allows the decoder network to “focus” on a different part of the encoder’s outputs for every step of the decoder’s own outputs.

First, we calculate a set of *attention weights*. First, compute the alignment score between and:

And then compute the attention weight for :

These will be multiplied by the encoder output vectors to create a weighted combination. The result should contain information about that specific part of the input sequence, and thus help the decoder choose the right output words.

## EXPERIMENT

### Dataset

There are a total of 23,610 translation data pairs, and each pair of translation data is on the same line: English on the left, Chinese in the middle, and other attributes information on the right. The separator is \t.

### Implementation

1. Download [Chinese-English translation dataset](http://www.manythings.org/anki/cmn-eng.zip)，unzip as ./data/eng-cmn.txt.

2. Read the dataset by row and remove the attribute information (only use top-2 split for each line) when constructing the training data pair. And some necessary preprocessing is done like turning a Unicode string into plain ASCII, lowering, trimming, and removing non-letter characters

3. Split words from the training sentences and construct a comparison table of Chinese and English words in the dataset, during which we build a Lang class to fully handle operations related to text data.

At this point, the data preparation work has been completed, we have obtained two corpora (3711 words in Chinese Corpus and 7459 words in English one) and have the ability to obtain the Chinese sentences we are interested in in the training data and their corresponding English translations (or vice versa), as the following picture shows:

Figure. 4. The result of preprocessing

4. Build a machine translation model：

* Build the encoder（Encoder）.
* Build a decoder based on the attention mechanism（Attention Decoder）.

1. Define loss function and train machine translation model. The cross-entropy function is chosen as loss function, 75000 iterations are conducted and gradient descent optimizer is adopted.

TABLE I

MODEL PARAMETERS

|  |  |
| --- | --- |
| Max length for training sentence | MAX\_LENGTH = 50 |
| Iteration numbers | n\_iters=75000 |
| Size of hidden layers | hidden\_size=256 |
| Learning rate | 0.01 |

6. Evaluate the training model using BLEU score.

7.Visualize the test results and organize experiment results.

As we can see from Figure 5, the average loss keep falling, and decrease to about 2.4277 after 75000 iterations.

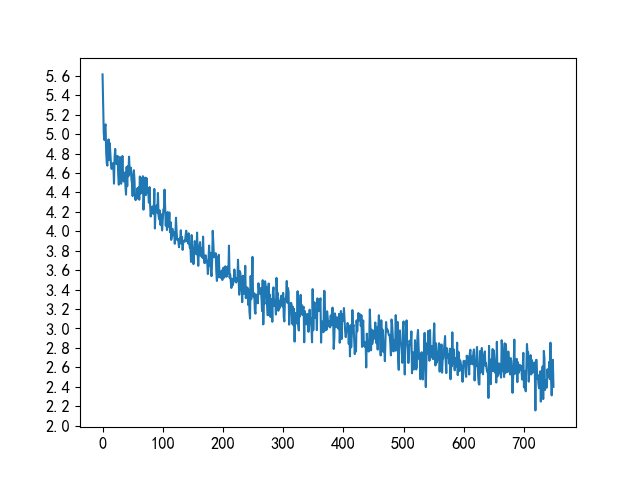


Figure.5. Loss of training process

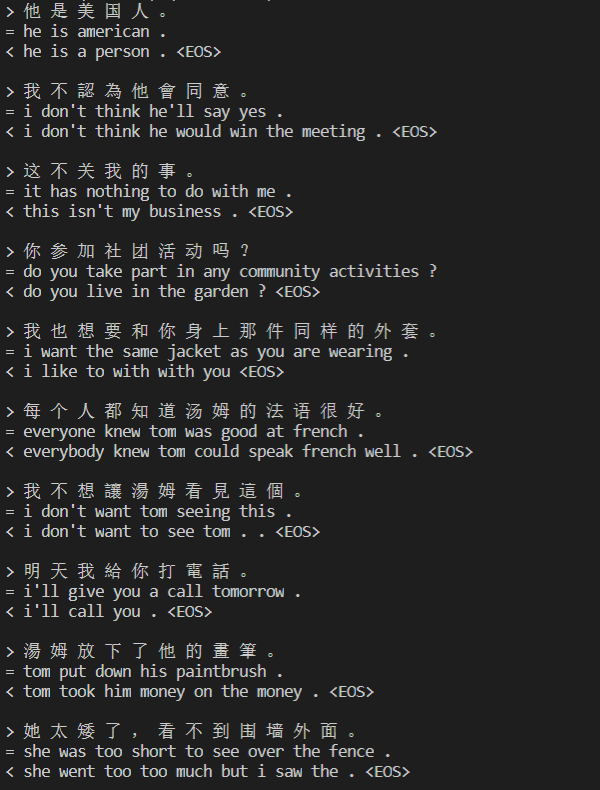
After that, we randomly choose some translation pairs to observe the prediction of the network：

Figure.6. The output of translation

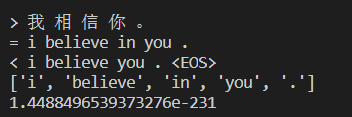
Use machine translation metrics such as BLEU to evaluate the trained model.

Figure.6. The output of translation with BLEU value

 A useful property of the attention mechanism is its highly interpretable outputs. Because it is used to weight specific encoder outputs of the input sequence, we can imagine looking where the network is focused most at each time step.

### Figure.7. Visualizing Attention

## CONCLUSION

1. Summary

Machine translation itself is very challenging and uncertain work with the difficulty of translation selection and order as well as data sparsity. As we can see from the results above, the result of translation is not satisfactory. However，compared with statistical machine translation, neural network translation is relatively simple in terms of model.

1. Gain and inspiration

In order to improve the quality of translation, we can adopt transformer in the model. In addition，the BLEU value has obvious disadvantages. The score of BLEU value is affected by reference. The more diverse the reference, the greater the possibility of matching. However, in out experiment the reference is single，which made the score not informative. Instead, using automatic evaluation and manual evaluation methods to measure will be more persuasive and accurate.