NLP with Seq2Seq model*

1st Zhang Qing 1st Mi Ao 1st Li Shichao

***Abstract*—This document is an instruction for the nueral translation model which is used to translate language from German into English. This model is based on Seq2Seq structure and attention mechanism. And it works in the Pytorch framework, Seq2seq strategy and attention mechanism is explained. Preprocessing for the dataset and training process are illustrated. Analysis of the result and potential influencing factors to the performence are also discussed.**

***Index Terms*—Pytorch, Seq2Seq,NLP,MT,bleu**

I. Introduction

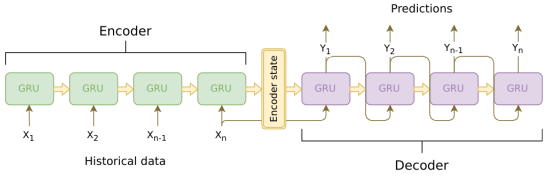
Machine translation is an important task in natural language processing (NLP) as it automates the translation process and reduces the reliance on human translators.

The idea of machine translation in early stage was very straightfoward, which is setting up a large number of translation rules and a large bilingual comparison table is constructed to translate the source language into the target language. Following that, statistical machine translation, a strategy based on probability, was developmented. It performences well but extremely complicated. At present, thanks to the proposed deep learning, Neural Machine Translation takes place gradually.

II. Aglorithm

A. Seq2seq

Seq2seq network structure is consisted of two substructure, encoder and decoder. Both encoder and decoder are implemented by RNN. Encoder transforms input word to a vector. Then decoder predicts the content we need according to this vector and the result of the last time step as input.

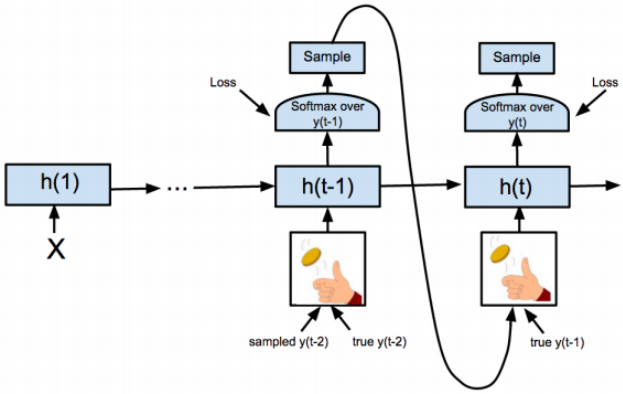


To be specific, the Encoder side first converts the word sequence of the source text into vectors through the embedding layer, and then vectors are entered into an RNN structure. In addition, RNN can also be multi-layered or bidirectional. After a series of computations of RNN, the input of the final hidden layer is regarded as a representation vector of the whole source text, which is called the context vector.

The situation of decoder side is more complicated. In the training process, the correct target text, the "standard answer", is used as inputs. But in predicting process, there are no so-call "standard answer". Each step predicts the next output word which is based on the current correct output word and the hidden state of the previous step.

During training process of the Decoder, the input of each time step is not exact the output of the previous time step. The true value is also probably selected as the input.

The above strategy is imported as scheduled sampling. The Decoder's prediction may be based on the input of previous step with probability p, and on the "standard answer" with probability $1-p$.



Typically, the input sequence of an Encoder needs to be added with a terminator $''\langle EOS\rangle''$, but the starting character $''\langle SOS\rangle''$ is not required.The Decoder input sequence needs to add a initiation and termination character during training. In prediction, the Decoder receives a initiation character $''\langle SOS\rangle''$, which is similar to a signal telling the Decoder to start working, and we can stop when it outputs a termination character $''\langle EOS\rangle''$(usually a maximum output length can be set, preventing a Decoder from never printing a terminator.)

B. The loss function of Seq2seq

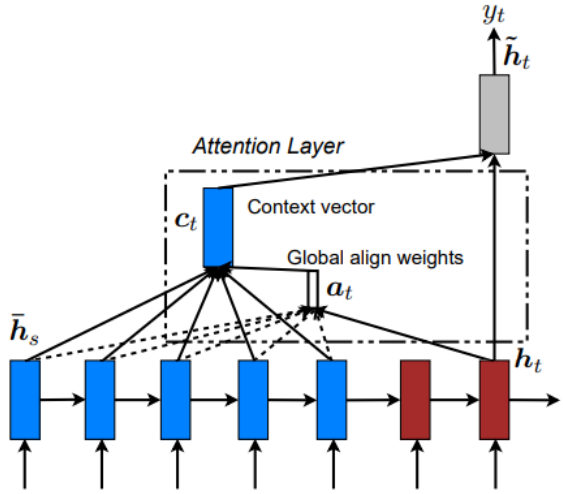
After generating the hidden state, the Decoder is to map this hidden state to corresponds word by a projection layer. And the projection layer is usually a softmax model which outputs an V-dimensional vector with the same dimension as vocabulary. Each of their components of vector represents the probability of a word.

Cross entropy is chosen as the loss function. Therefore, we need to find out the one-dimensional probability $\hat{p}$ of the correct prediction of the corresponding word in the V-dimension vector, then the loss of this step is $-log(\hat{p})$, and the sum of the losses of each step is obtained, namely, the overall loss function: $J = -log(p\hat{y_1}) - log(p\hat{y_2}) - ... - log(p\hat{y_n}) - log(p(EOS)) = -\frac{1}{T}\sum_i^T -log(p\hat{y_i})$ And T is total steps of Decoder.

C. Attention

The ordinary decoder mechanism simply converts the entire sentence into a vector, which does not conform to the real situation where human translation always has a process of "word-for-word translation". Meanwhile ordinary decoder mechanism lacks the learning of details. Those shortcomings have become the translation effect cannot be improved.

So we improve the decoder mechanism by making sure that information of the entire input sequence is encoded by the Encoder into a fixed-length vector which is similar to "lossy compression". This vector cannot fully express the information of the entire input sequence. In addition, as the input length increases, this fixed-length vector will gradually lose more information.



So we define that: at each time step t, the hidden state of the Decoder at the current time step is h_t. Its hidden state is h_s. While the hidden state of output by the entire Encoder is $\overline{h_s}$, the weight value a_t, and its context vector is ct.

The value of attention can be quantified by the following way: value is calculated in the following way:

$$score(h_t, \overline{h_s}) = F^{HLLC} = \begin{cases} h_t^T \overline{h_s} & dot \\ h_t^T W_a \overline{h_s} & general \\ v_a^T tanh(W_a[h_t; \overline{h_s}] & concat \end{cases}$$

By softmax normalization, we get

$$a_t(s) = align(h_t, \overline{h_s}) = \frac{exp(score(h_t, \overline{h_s})}{\sum_{s'} exp(score(h_t, \overline{h_s})}$$

According weight, The vector is averaged for each time step of the hidden state output by the Encoder according to the weight.

$$c_t = \sum_s (s) \cdot \overline{h_s}$$

It can be spliced with the hidden state of the Decoder at the current time step to calculate an attention hidden state.

$$\overline{h_t} = tanh(W_c[c_t; t])$$

Than we can Predict the output result based on this hidden state of attention.

$$y = softmax(W_s \overline{h_t})$$

```
a person stands on a structure on the lake .
a person structure the structure the of the . <EOS>

several young adults performing in martial arts clothing .
a young boy martial on the the a a a . <EOS>

three men are setting up a sledge on a snowy plain .
three dogs stand a a on a sitting on a . <EOS>

a dog is running on the beach .
a couple the beach . <EOS>

a little boy is holding a green and yellow sponge while standing in front of a stove .
a little boy is holding in of front of . <EOS>

a little girl playing in water on a frog sculpture .
a little girl playing in water on a stage on a stage . <EOS>

a smiling boy runs through the grass .
a smiling boy through the grass . <EOS>

three children  baby in the center  sit in an outdoor swing chair .
three children guys outside outside outside outside outside . <EOS>

a young boy in a swimming suit sits in water .
a young boy in a red suit a water . <EOS>

a man in a black suit and briefcase hurrying somewhere  while talking on his cellphone .
a man in a shirt is his by . <EOS>
```

III. Preprocessing and Training

The given data sets separates original language and target language into different files. To training more convenient, we joint them to one file. In the new data set, the sentence of original language and its target language are placed in the same line and divided by t character.

IV. Result

[H]

we implement two seq2seq models, one of them is consist of multiple encoder,decoder layers,and bidirection which is sentences split by char. the other one is consist of one layer while its sentences is split by words.epoch23min, 500epoch Because the first bidirection multi-layer model requires 23 minutes per epoch and whole model requires nearly 500 epochs to converge, we show the results of the second single-layer model here. We will hand in both code.

V. Influence Elements

A. infulence element analysis

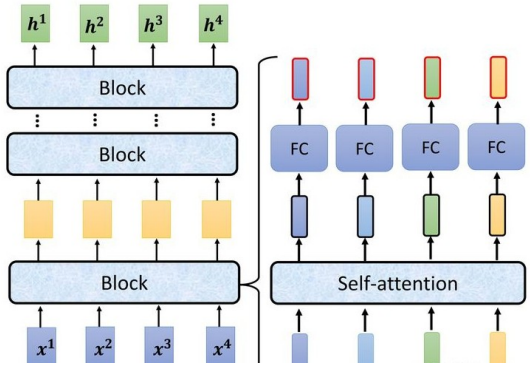
- **learning rate**
  It determines whether the neural network can converge to the global minimum. Choosing a higher learning rate may lead to a undesirable consequence for your loss process, so we may never reach the global minimum because the global minimum is likely to be skipped. Choosing a small learning rate may help the neural network converge to the global minimum, but it will take a lot of time, because you only make very few adjustments to the weights of the network. So you have to spend more time training the neural network. A smaller learning rate is also easier to make the neural network fall into a local minimum, that is, the neural network will converge to the local minimum, and because the learning rate is relatively small, it cannot jump out of the local minimum.

  In this project, we run the model with $0.0001, 0.001, 0.01, 0.1$ learning rate, and if the learning bigger than 0.001, the loss does not converge.

- **epoch**
  It costs 15min-30min to run a epoch, the size of epoch determine if the model can converge and the total running time. We use 10, 50,100,200 as parameter to

train the model. So we find the model start to converge after 170 epochs with 0.001 learning rate.

- **layer**
  We will use RNN to encode the input. As shown in the figure below, with the increase of N_{Layer} parameter, There will be layers superimposed in coding layers, which is similar to each block in the figure below. Each block has a input consisted of a row of vectors, and its output is a row of vectors.



  Increasing the number of layers will increase the training time and will need to increase epoch. If the layer is more than 6, it will lead to over fitting. After this test, we think 2 layers are enough for out model.
- **data size**
  After the train, we find the relationship between data size and train time can be described as the time cost function as follows

$$t = e^{\alpha m}$$

  α is a parameter
  m is the sample size (you can think it is the number of rows in the train data set). Then there will be some trade off. We can using small data set to reduce the train time with less accuracy. Or we can use big data set to ensure the precise with more time.
  According to the time cost function, we use the half size of the data size to train out model.
- **data format**
  As we all know, a nice format of data set can help us to train a nice model. A useful method is helpful to control the length of each sentence. the data set Separated to long sentence set and short sentence set behave better.
- **bidirectional**
  bidirectional indicates that the model learns information from left to right and from right to left. When certain words in a sentence are covered up or hidden, model can read the entire sentence from left to right and from right to left according to your language knowledge. Train the model indirect can increase the accuracy of translation.

B. Result with different parameters

Translation result with 1 layer, 0.0001 learning rate, 20 epoch.

```
YM`2"My2"rpMR=`;"MMq2y"My2"yMGg"M`#"LM=y
YM`2"GMy2"MMyg2q"MMq"M`#"M""Mpp"BMqg=`
YM`2"My2"rMyg2y"My2"LM=yq"My2"LMq"My2"L"M
YM`2"My2"My2"My2"LMMy=`;"MMq2"MMq"MMq'
YM`2"GMy2"MM`""MM`""MMq2y"MMq2y"MMq2y'
YMä2yMp"GMyy=`;"My2"GMyy=`;"M"G'ä2y"GMyG!
YM`2"My2"rMyg2y"MMq2y"MMq2y"MMq2""MMq"
YM`2"My2"LM`"My2"My2"My2"M`#"MMq2y"MMq2
YM`2"My2"rpMR=`;"MMq2y"MMq2y"MMq"M`#"MI
YM`2"r2yL'y"=L"GMy=`;"M`"My2"LM`#=`;"M`"My2"M
Y'ä2`"My2"MM`g=`;"M""My2"MMq2""MMq"M""M"
YM`G2";yMLL2#"äM`"=L"GMyG!=`;"M`"M"L2MG!"M
YM`2"My2"LMM`=`;"M""M"2""MMyg"M`#"M""M"2'
YM`2"rMyg2y""My2"MM`"M`#"MMq2y"MMq"M""M
YM`2"r2Mrp2"My2"My2"My2"M`#"SpMGg"GMy""M
YM`2"My2"MMpg=`;"M""M"2""Mpp"BMq2y"MMpg
YMä2"GMyG2y"MMq2y""My2"My2"My2"My2"My2"
YM`2"My2"LMyg2y"My2"LM`#=`;"M`"M"G`q"M`""M
YM`2"GMy"GMy"My2"LM`""MMq2y""`""M""M"2""
YM`2"MMy2"MMq2y"MMq2y"MMq"M""M"
YM`"r2Mrp2"My2"LMM`=`;"M""M"2""MMq"M""M"'
```

Translation results with 2 layer, 0.0001 learning rate, 20 epoch.

```
A woman is sitting a a a a a a a a a a a ae a ae ae a
A man with a white a red a a a ae ae ae ae ae ae ae
A woman in a sting a a a a a a e sitting a standing a stand
A woman in a white a white a white shild ng a a in in a blu
A man in a back a a a a a a a a ae ing a ae ae ae ae ae
A citting a standing a a on a e on a e .
A people and in a back a a a a a a a a a white a a while
A man is sitting a a a a a a a ae in a standing a a e in a e
A woman with a black a a a a a a a a ae a ae ae ae a a
A ban is and in a ring a a a a aea aeaeaeaeae.
A man in a back and a a a a a a a aea aeaeaeae ae seat
A young girl in a red in a red in a street in a back a ae a e si
A standing with a blae in a a a a a a a a e a e ae ae ae
A little girl in a blae a ing a black a a a ae a ae ae ae
A people with a blue a a a ae a ae ae ae ae ae ae ae
A woman iith a blue with a blue a a ae a ae ae ae ae ae
A man in a blay in a r a a a a a aea aeaeaeae.
A little bitting a black a a a a a a a a a a a a a a a ae
A man is a red a ae a a e street in the sare a a ae ae ae t
A man is sitting a blae a a a ae ae ae ae ae ae ae
Two men in a red with a red in a red a a a a a a a a aea
A coot on a man in a d and and and and and and and and an
Two children a d and a while a a a a a a ae of a a e a e a e
```

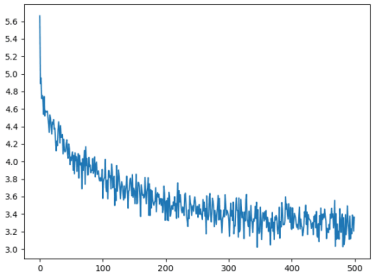
Translation results with 2 layer, 0.0001 learning rate, 50 epoch.

```
Eine Frau schaut sich ein rotes Halstuch von einer der Straße
A woman sit in a red shirt an a red shirt an a street an a stre

Ein Mann mit weißer Schürze hält ein Huhn über einen groß
A man with white shirt is shirt is a gray a a large a

Eine Frau in Rot sitzt am Rand eines steilen Felsens.<eos>
A woman in a sitting a a a a a a a a a a e a a e .

Eine Frau steht in einem weißen Raum, der mit Zahlen verse
A woman standing a white a white a white and with a wate

Ein Mann zeigt einer Person in einem Auto verschiedene Sch
A man is sitting a people are in a sand with a sare a are par

Ein Pitcher steht auf dem Hügel.<eos>
A person standing on the sareet.

Eine Person in einer roten Jacke hat ein Bier in der Hand und
A person in a red jacket and back and a man in a man in a r

Ein Wanderer steigt mit einem Spazierstock einen Weg mit S
A woman standing a a a a a a a a a a a a a a a a a a
```

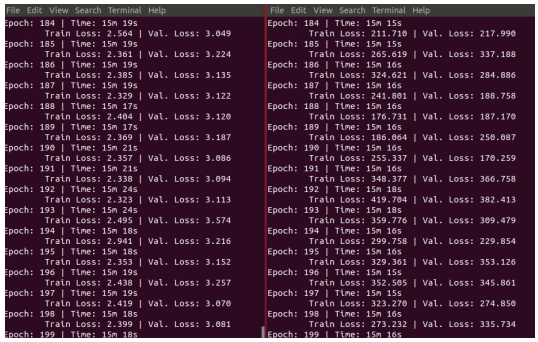
Train time with 2 layer, 0.0001 learning rate, 200 epoch, 10 data size.

```
Epoch: 01 | Time: 0m 2s
        Train Loss: 3.602 | Val. Loss: 3.509
Epoch: 03 | Time: 0m 2s
        Train Loss: 3.421 | Val. Loss: 3.231
Epoch: 05 | Time: 0m 2s
        Train Loss: 3.182 | Val. Loss: 3.047
Epoch: 07 | Time: 0m 2s
        Train Loss: 3.033 | Val. Loss: 3.090
Epoch: 09 | Time: 0m 2s
        Train Loss: 2.986 | Val. Loss: 3.104
Epoch: 11 | Time: 0m 2s
        Train Loss: 2.942 | Val. Loss: 2.975
Epoch: 13 | Time: 0m 3s
        Train Loss: 2.911 | Val. Loss: 2.943
Epoch: 15 | Time: 0m 3s
        Train Loss: 2.903 | Val. Loss: 2.952
Epoch: 17 | Time: 0m 3s
        Train Loss: 2.878 | Val. Loss: 2.979
Epoch: 19 | Time: 0m 3s
        Train Loss: 2.881 | Val. Loss: 2.945
Epoch: 21 | Time: 0m 3s
        Train Loss: 2.831 | Val. Loss: 2.890
Epoch: 23 | Time: 0m 2s
        Train Loss: 2.806 | Val. Loss: 2.879
Epoch: 25 | Time: 0m 3s
        Train Loss: 2.808 | Val. Loss: 2.871
Epoch: 27 | Time: 0m 3s
        Train Loss: 2.820 | Val. Loss: 2.860
Epoch: 29 | Time: 0m 2s
```

Loss change with 2 layer, 0.0001 learning rate, 50 epoch.



left side is the change of loss with 2 layer, 0.001 learning rate, 250 epoch. Right side is the change of loss with 2 layer, 0.01 learning rate, 250 epoch.



VI. ENVIRONMENT

We use ubuntu 18.04 with pytorch to run the code. To get such a virtual environment, first we should get ubuntu+pytorch [1],then we download cudnn with the documents provided by nvidia. [2]
The computer for run this model is Dell Precision T5820 with NVIDIA Quadro P2000.

VII. ASSIGN OF LABOUR

TABLE I
TABLE TYPE STYLES

member	***assigment***
Zhang Qing	
Mi Ao	
Li Shichao	

REFERENCES

[1] R. C. 12138, "Deep learning environment configuration ubuntu+pytorch," 2019. https://zhuanlan.zhihu.com/p/78075698/.

[2] NVIDIA, "nvidia cudnn documentation," 2021. https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html#install-windows.

[3] "Pytorch." https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[5] https://www.cnblogs.com/dogecheng/p/12864713.html, title = PyTorch implements Seq2Seq machine translation, author=The boy and the dog, year=2020,.