Seq2seq模型——机器翻译

实验报告

姓名:王雨静

班级 ： 1期NLP训练营

日期:2021年3月9日

目录

一、案例简介

seq2seq是神经机器翻译的主流框架，如今的商用机器翻译系统大多都基于其构建，在本案例中，我们将使用由NIST提供的中英文本数据训练一个简单的中英翻译系统，在实践中学习seq2seq的具体细节，以及了解机器翻译的基本技术。

1.1 seq2seq模型

从根本上讲，机器翻译需要将输入序列（源语言中的单词）映射到输出序列（目标语言中的单词）。正如我们在课堂上讨论的那样，递归神经网络（RNN）可有效处理此类顺序数据。机器翻译中的一个重要难题是输入和输出序列之间没有一对一的对应关系。即，序列通常具有不同的长度，并且单词对应可以是不平凡的（例如，彼此直接翻译的单词可能不会以相同的顺序出现）。

为了解决这个问题，我们将使用一种更灵活的架构，称为seq2seq模型。该模型由编码器和解码器两部分组成，它们都是RNN。编码器将源语言中的单词序列作为输入，并输出RNN层的最终隐藏状态。解码器与之类似，除了它还具有一个附加的全连接层（带有softmax激活），用于定义翻译中下一个单词的概率分布。以此方式，解码器本质上用作目标语言的神经语言模型。关键区别在于，解码器将编码器的输出用作其初始隐藏状态，而不是零向量。

1.2 数据和代码

本案例使用了一个小规模的中英平行语料数据，并提供了一个简单的seq2seq模型实现，包括数据的预处理、模型的训练、以及简单的评测。

1.3 评分要求

分数由两部分组成，各占50%。第一部分得分为对于简单seq2seq模型的改进，并撰写实验报告，改进方式多样，下一小节会给出一些可能的改进方向。第二分部得分为测试数据的评测结果，我们将给出一个中文测试数据集（test.txt），其中每一行为一句中文文本，需要同学提交模型做出的对应翻译结果，助教将对于大家的提交结果统一机器评测，并给出分数。

1.4 改进方向

初级改进：

- 将RNN模型替换成GRU或者LSTM
- 使用双向的encoder获得更好的源语言表示
- 对于现有超参数进行调优，这里建议划分出一个开发集，在开发集上进行grid search，并且在报告中汇报开发集结果
- 引入更多的训练语料（如果尝试复杂模型，更多的训练数据将非常关键）

进阶改进：

- 使用注意力机制(注意力机制是一个很重要的NMT技术，建议大家优先进行这方面的尝试，具体有许多种变体，可以参考这个综述)
- 在Encoder部分，使用了字级别的中文输入，可以考虑加入分词的结果，并且将Encoder的词向量替换为预训练过的词向量，获得更好的性能

复杂改进：

- 使用beam search的技术来帮助更好的解码，对于beam-width进行调优
- 将RNN替换为Transformer模型，以及最新的改进变体

二、评价函数

2.1 BLEU指标

本实验使用bleu对翻译效果进行评价。
由于1-gram中仅考虑输出的词是否在参考答案中，只考虑了准确率却没有考虑长度、语义等，一些不好的翻译可能会有虚高的准确率。所以不能只考虑1-gram的bleu值。
在此实验中，将综合考虑1-gram、2-gram、3-gram、4-gram的值，并对其求加权和，且对过短翻译进行惩罚。

对模型的评价方法：
在验证集中取500个句子（处于运行速度考虑），分别求1-gram、2-gram、3-gram、4-gram、weighted-bleu，并求和。
（备注：结果有效范围0-500）（实际bleu值应求平均，即所示结果除以500）

2.2 代码实现

```
import numpy as np
import pandas as pd

def  bleu1(r, c): #both are list(sequence) of words
  total = len(c)
  correct = 0

  if total == 0: return 0

  for word in c:
    if word in r: correct += 1
  return correct / total

def bleu2(r, c):
  r_list = []
  c_list = []
  for i in range(len(c) - 1 ):
    phrase = ' '.join((c[i], c[i+1]))
    c_list.append(phrase)
  for i in range(len(r) - 1 ):
    phrase = ' '.join((r[i], r[i+1]))
    r_list.append(phrase)
```

```
  total = len(c_list)
  correct = 0
  if total == 0: return 0

  for phrase in c_list:
    if phrase in r_list: correct += 1

  return correct / total

def bleu3(r, c):
  r_list = []
  c_list = []
  for i in range(len(r) - 2 ):
    phrase = ' '.join((r[i], r[i+1], r[i+2]))
    r_list.append(phrase)
  for i in range(len(c) - 2 ):
    phrase = ' '.join((c[i], c[i+1], c[i+2]))
    c_list.append(phrase)
  total = len(c_list)

  if total == 0: return 0

  correct = 0
  for phrase in c_list:
    if phrase in r_list: correct += 1

  return correct / total

def bleu4(r, c):
  r_list = []
  c_list = []
  for i in range(len(r) - 3 ):
    phrase = ' '.join((r[i], r[i+1], r[i+2], r[i+3]))
    r_list.append(phrase)
  for i in range(len(c) - 3 ):
    phrase = ' '.join((c[i], c[i+1], c[i+2], c[i+3]))
    c_list.append(phrase)

  total = len(c_list)

  if total == 0: return 0
```

```
  correct = 0
  for phrase in c_list:
    if phrase in r_list: correct += 1

  return correct / total

def BP(r, c):
  lc = len(c)
  lr = len(r)
  if lc == 0: return 0
  if lc > lr: return 1
  else: return np.exp(1 - lr/lc)

def w_bleu(r, c): #N = 3, wi = 1/4
  w_total = np.exp(1/4 * (np.log(bleu1(r, c))+np.log(bleu2(r,
c))+np.log(bleu3(r, c))+np.log(bleu4(r, c))))
  bp = BP(r, c)
  score = bp * w_total
  return score

def bleu_eva():
  bleu_1 = 0
  bleu_2 = 0
  bleu_3 = 0
  bleu_4 = 0

  bleu_w = 0

  n = len(pairs_dev)

  for i, pair in enumerate(pairs_dev[:500]):
    reference = pair[1].split(' ')
    candidate = evaluate(pair[0])
    bleu_1 += bleu1(reference, candidate)
    bleu_2 += bleu2(reference, candidate)
    bleu_3 += bleu3(reference, candidate)
    bleu_4 += bleu4(reference, candidate)
    bleu_w += w_bleu(reference, candidate)

  return bleu_1, bleu_2, bleu_3, bleu_4, bleu_w
```

三、模型改进与对比评价

3.1 原模型效果

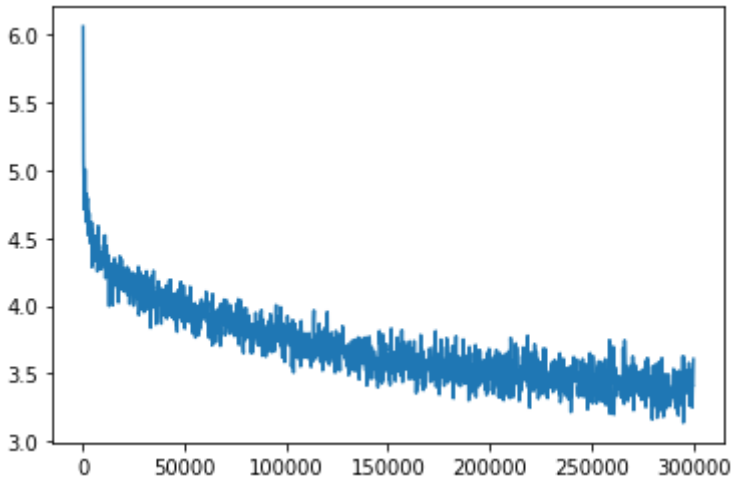

Fig1. 原模型效果

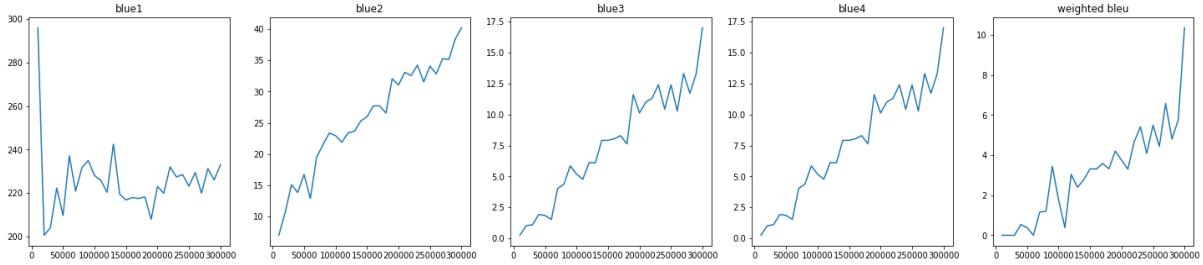

Fig2. 原模型bleu值

略做尝试，原模型跑5万个epoch后，翻译结果较短且有末尾有很多'.'，可见模型甚至不能很好地判断一句话的结束。因此增加训练次数。
原模型在耗时90分钟，运行了30万个epoch后，loss下降到了约3.5左右，2-gram及以上的bleu值虽然在持续上升，但是依旧很低，约为10/500=0.02左右，由此可见模型的效果并不好。

3.1.1 翻译例子

```
> 我以为这是真的。
= i thought it was true .
< i am very happy to . <EOS>

> 我想去商场。
= i d like to go to the mall .
< i want to eat something . <EOS>

> 他是个天才。
= he is a genius .
```

```
< he is my father . <EOS>
```

3.2 加入分词

3.2.1 分词工具包

本实验使用的是THULAC（THU Lexical Analyzer for Chinese）由清华学自然语言处理与社会人文计算实验室研制推出的一套中文词法分析工具包，具有中文分词和词性标注功能。

代码：

```
# 中文分词工具
!pip install thulac
import thulac
```

例子：

```
#加入中文分词
thu1 = thulac.thulac(seg_only=True, deli=' ')
thu1.cut('我爱北京天安门。', text=True).split(' ')
```

```
Model loaded succeed
['我', '爱', '北京', '天安门', '。']
```

3.2.2 模型效果对比

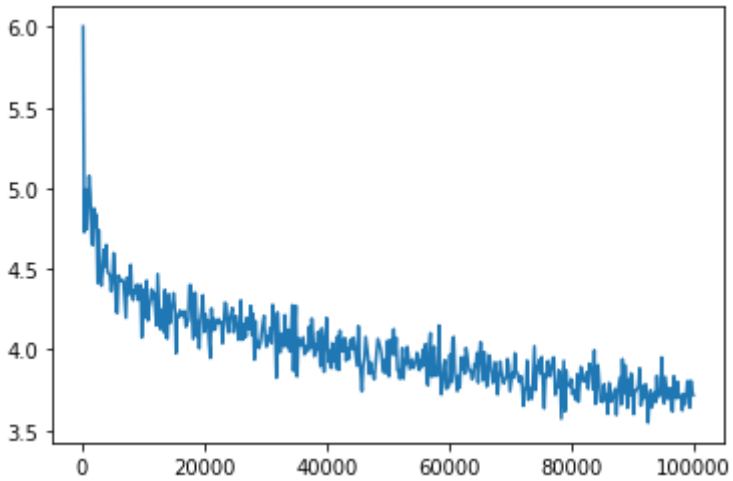

Fig3. 分词后loss图

由于时间有限，训练了10万个epoch。

和Fig1对比可以发现，分词后loss下降可能略快一些，但大致相似。

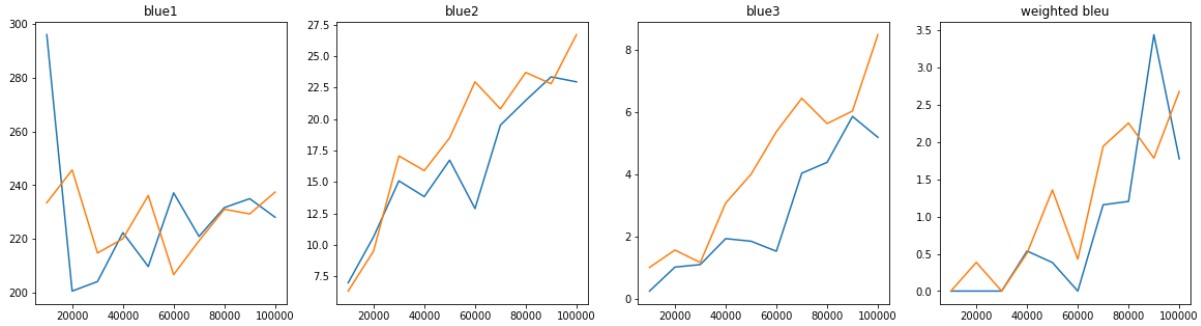

Fig4. 分词前后bleu值对比

图中蓝色代表原模型，橙色代表分词后的模型。从图中可以发现，分词后模型效果可能略有提升，但是总体效果仍然较差。

3.2.3 翻译例子

（总觉得这个模型有点阴阳怪气，/苦涩.jpg）

```
> 我以为这是真的。
= i thought it was true .
< i m very tired . <EOS>

> 他为啥生气？
= why is he angry ?
< what did he say ? <EOS>

> 你有道理。
= you re right .
< you re really idiot . <EOS>

> 我想去商场。
= i d like to go to the mall .
< i want to eat something . <EOS>

> 這個計劃徹底的失敗了。
= the project was a complete failure .
< this is is t to . . . <EOS>
```

3.3 使用GRU模型

3.3.1 GRU模型

GRU（Gate Recurrent Unit）是循环神经网络（Recurrent Neural Network，RNN）的一种。和LSTM（Long-Short Term Memory）一样，GRU能解决长期记忆和反向传播中的梯度等问题。和LSTM相比，GRU模型的效果类似，但是训练效率更高。所以本实验中选用了GRU模型。

3.3.2 模型效果对比

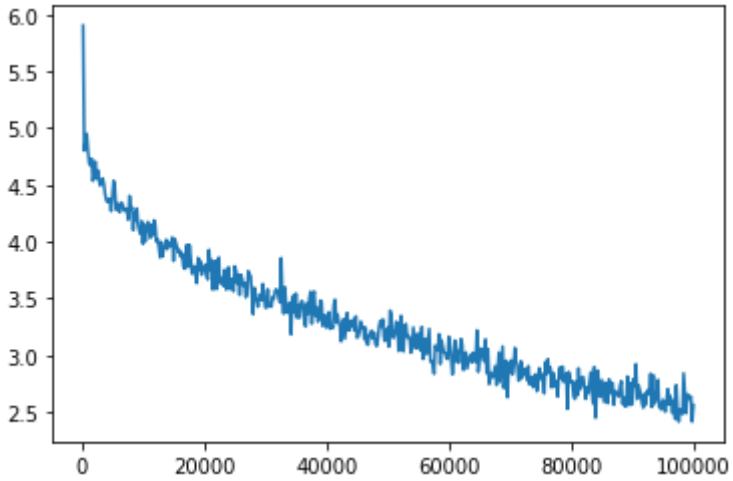

Fig5. GRU模型loss

和Fig3对比可以发现，GRU模型在仅仅10万个epoch就将loss降到了2.5左右，且还在持续下降。

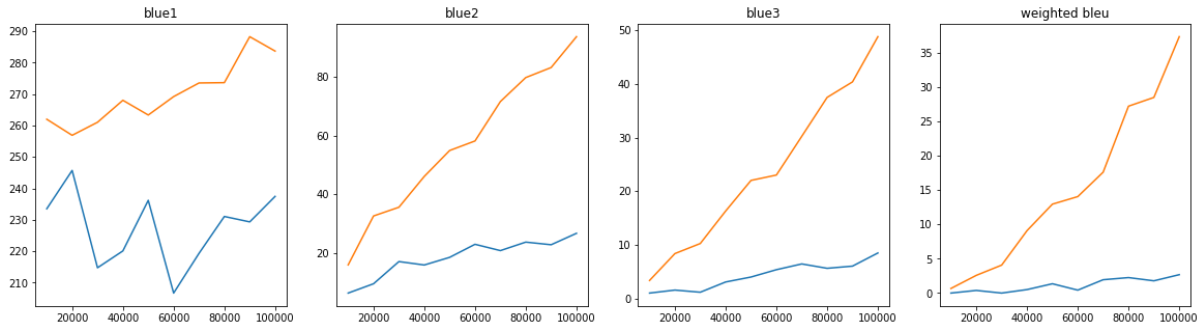

Fig6. GRUbleu值对比

图中蓝线代表的是原模型，橙线代表的是GRU模型。

从图中可以发现，将RNN层替换成GRU后，不论是1-gram还是2-gram，3-gram，GRU模型的bleu值都远大于原模型，由此可见模型效果大幅度提升了。而且bleu值还在持续上升，说明模型还有训练空间，可以训练更多组数。

3.3.3 翻译例子

```
> 我以为这是真的。
= i thought it was true .
< i think it s true . <EOS>

> 他为啥生气？
= why is he angry ?
< does he angry ? <EOS>

> 你有道理。
= you re right .
< you re right . <EOS>

> 我想去商场。
= i d like to go to the mall .
< i want to go to america . <EOS>
```

虽然还是有很多错误，但是翻译效果比之前的大幅度提升了。

3.4 增大MAX_LENGTH

相比起RNN，GRU能够更好地处理更长的序列，所以本实验尝试了增加MAX_LENGTH（句子最大长度）,将其从10增加到20进行训练。

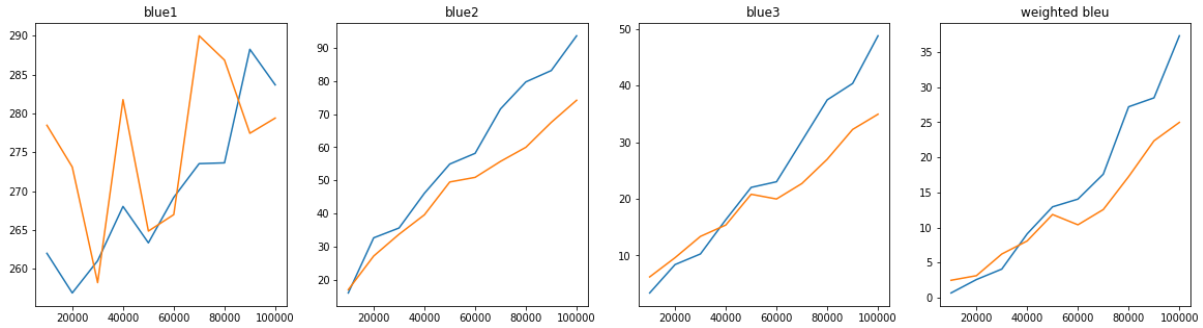

Fig7. MAX_LENGTH bleu值对比

图中蓝色的是max_length = 10，橙色的代表max_length = 20。

从图中可以发现，max_length较低的模型的bleu值略高一些。但这其实可能并不能说明模型效果更好，因为句子的筛选（filter）是在划分数据集前进行的，两者的开发集的数据是不一样的，所以这样的评价对max_length = 20的模型来说并不公平（由于其开发集的句子可能更长）。

但是，图中两者的多gram的bleu值都在持续提升，更长的句子长度也没有使模型效果下降很多，这反应了GRU有处理长句子的潜力。而实际生活中的句子，大多比较长，所以翻译长句子的能力是很重要的。

3.5 增大训练次数

基于前面的改进（包括分词、GRU、max_length = 20），增大训练次数至30万个epoch耗时128分钟。

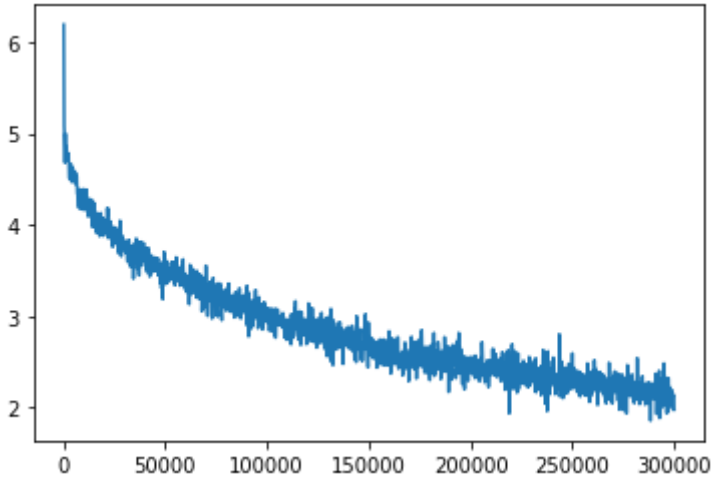

Fig8. GRU继续训练loss

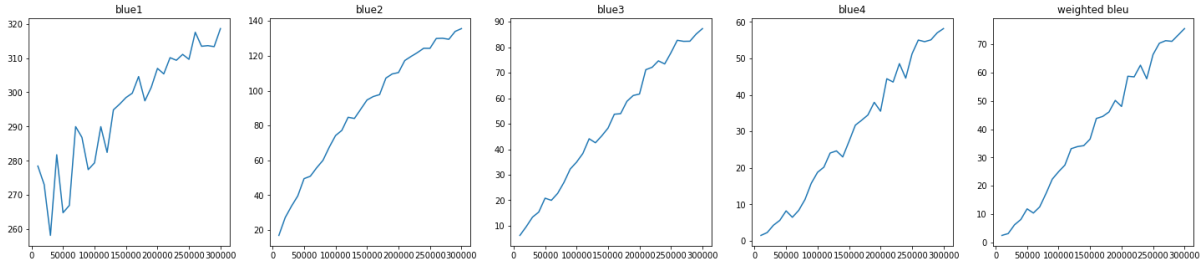

Fig9. GRU继续训练bleu值

从图中可以发现，loss开在持续下降，bleu值也在持续提升，模型效果还有从继续训练中获得提升的空间。

3.5.1 翻译例子

```
> 我留在希尔顿宾馆。
= i m staying at the hilton hotel .
< i am staying in the office . <EOS>


> 我明天回來。
= i ll be back tomorrow .
< i will be back tomorrow . <EOS>

> 我住在東京。
= i live in tokyo .
< i live in tokyo . <EOS>
```

3.6 注意力机制

3.6.1 参考链接和代码链接

参考链接 :https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

代码实现链接：

https://colab.research.google.com/drive/1VAFwcVAt_XW438oSHwDKj6hiLOFHCiOx?usp=sharing

（但是后来由于Google colab GPU使用上限，转移到百度AI平台进行训练和测试）

（最终代码上百度AI平台上，实验一版本一）

3.6.2 模型效果

耗时3小时，训练了30万个iteration，最终结果如下：

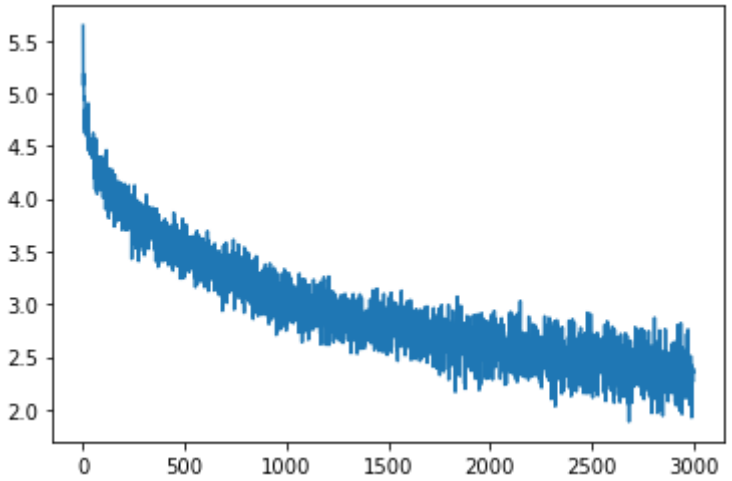

Fig10. 注意力模型loss

*备注：x轴值需乘以100

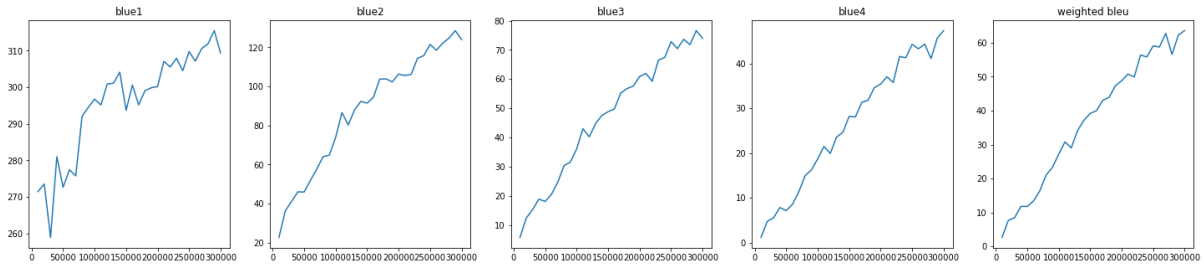

Fig11. 注意力模型bleu值

在前30万个opech的训练中，从loss值来看，注意力模型的loss值波动较大，但总体呈现和GRU模型相近的下降趋势，但是模型效果似乎比单纯的GRU模型略差一些。

这可能是因为注意力模型参数较多，所以训练速度更慢一些。也可能是模型较为复杂，但是训练集较小使得其效果没有那么好。也可能是模型其他方面的架构不同导致的。

疑惑点:是因为对文本分词不同造成的吗？

3.6.3 翻译例子

```
> 那 個 箱子 是 用 木造 的 。
= that box is made of wood .
< that box is made from . . <EOS>


> 我 比 你 帅 。
= i am more handsome than y o u .
< i m more than than y o u . <EOS>


> 許多 朋友 來為 我 送行 。
= many friends came t o see me of f .
< many friends came t o see me of f . <EOS>


> 你 还 没 洗手 ， 不 是 吗 ？
= you haven t washed your hands yet have you ?
< you re still still t not ? ? <EOS>
```

3.7 使用更大规模的平行语料

语料链接 :http://data.statmt.org/news-commentary/v15/training/

由于时间限制，code虽然修改好了，但是没有完整运行过，不知道结果会怎么样。

新加入的语料的句子是较为复杂的，会提升任务的难度。

四、翻译test.txt

4.1 继续训练GRU

在3.5后，继续训练GRU模型至60万个epoch。（共耗时279分钟）

最后实现了bleu值约略大于 100/500 = 0.2。

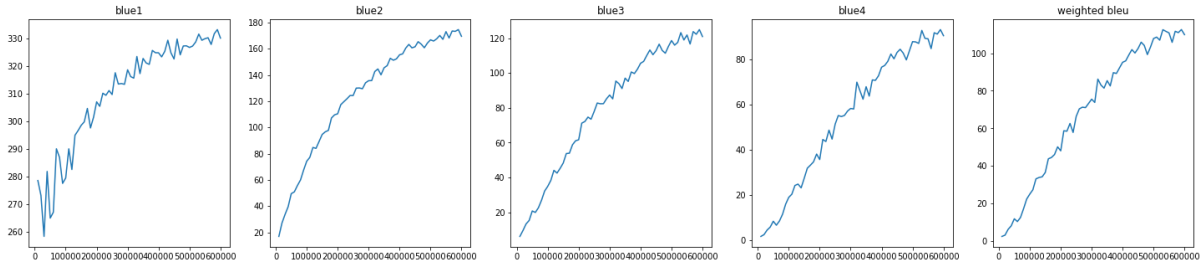

Fig. 12 GRU bleu值

Bleu值上升的速度已经开始放缓，最后就用了这个来翻译。

4.2 文件翻译代码实现

如果能翻译，就将结果写入输出文件中；如果出现了没见过的词，就跳过该句子，写上换行符('\n')。

```
with open('/content/drive/MyDrive/Colab Notebooks/xtzx/nlp2/seq2seq案例/test.txt') as file_object:
  lines = file_object.readlines()
with open('/content/drive/MyDrive/Colab Notebooks/xtzx/nlp2/seq2seq案例/GRU_output2.txt', 'w') as file_object:
  for line in lines:
    line = line.strip('\n')
    sentence = thu1.cut(line, text=True)
    print(sentence)
    try:
      trans = ' '.join(evaluate(sentence)[:-1])
      print(trans)
      file_object.write(trans)
      file_object.write('\n')
    except:
      print('\n')
      file_object.write('\n')
```

五、小结和反思

小结：在本次实验中，学习实践了seq2seq翻译模型，尝试了RNN、GRU、含有注意力机制的GRU等模型。发现使用分词、GRU能提升模型的性能，训练60万个epoch后的模型能进行一些翻译，但模型效果仍然有提升空间。

反思:由于开始写作业较晚，没有足够的时间来训练模型（然而也把Google colab GPU跑上限了，苦涩.jpg），没有进行GridSearch来参数调优，而只是在几个大方向上进行了探索与尝试。也没有深度学习注意力机制的模型，导致理解不足，没能找到为什么他还不如GRU模型的原因。

附录1：模型翻译例子对比

原模型3.1	加入分词后3.2	GRU3.3
pair 0 : > 我是见汤姆的最后一个。 = i was the last one to see tom . < i have a book for . . <EOS> pair 1 : > 你想加入哪一組？ = which group do you want to join ? < what do you think of this ? ? <EOS> pair 2 : > 我以为这是真的。 = i thought it was true . < i am very happy to . <EOS> pair 3 : > 你应该提前付租金。 = you should pay your rent in advance . < you should have for the . . . <EOS> pair 4 : > 你給測驗評分了嗎 ？ = did you grade the tests ? < did you buy this work ? <EOS> pair 5 : > 他为啥生气？ = why is he angry ? < he is that ? <EOS> pair 6 : > 你有道理。 = you re right . < you re very . . <EOS>	pair 0 : > 我是见汤姆的最后一个。 = i was the last one to see tom . < i m my my . . . <EOS> pair 1 : > 你想加入哪一組？ = which group do you want to join ? < how about you ? <EOS> pair 2 : > 我以为这是真的。 = i thought it was true . < i m very tired . <EOS> pair 3 : > 你应该提前付租金。 = you should pay your rent in advance . < you should be right . <EOS> pair 4 : > 你給測驗評分了嗎 ？ = did you grade the tests ? < did you know the that ? ? <EOS> pair 5 : > 他为啥生气？ = why is he angry ? < what did he say ? <EOS> pair 6 : > 你有道理。 = you re right .	pair 0 : > 我是见汤姆的最后一个。 = i was the last one to see tom . < i was a new yesterday s . . <EOS> pair 1 : > 你想加入哪一組？ = which group do you want to join ? < which would you like to your ? ? <EOS> pair 2 : > 我以为这是真的。 = i thought it was true . < i think it s true . <EOS> pair 3 : > 你应该提前付租金。 = you should pay your rent in advance . < you should have up your . <EOS> pair 4 : > 你給測驗評分了嗎 ？ = did you grade the tests ? < did you get up ? <EOS> pair 5 : > 他为啥生气？ = why is he angry ? < does he angry ? <EOS> pair 6 : > 你有道理。 = you re right .

pair 7 : > 我想去商场。 = i d like to go to the mall . < i want to eat something . <EOS> pair 8 : > 這個計劃徹底的失敗了。 = the project was a complete failure . < this ticket is very . . . <EOS> pair 9 : > 他是个天才。 = he is a genius . < he is my father . <EOS>	< you re really idiot . <EOS> pair 7 : > 我想去商场。 = i d like to go to the mall . < i want to eat something . <EOS> pair 8 : > 這個計劃徹底的失敗了。 = the project was a complete failure . < this is is t to . . . <EOS> pair 9 : > 他是个天才。 = he is a genius . < he is a very . . <EOS>	< you re right . <EOS> pair 7 : > 我想去商场。 = i d like to go to the mall . < i want to go to america . <EOS> pair 8 : > 這個計劃徹底的失敗了。 = the project was a complete failure . < the weather is is on the . <EOS> pair 9 : > 他是个天才。 = he is a genius . < he is a teacher . <EOS>

附录2：实验记录链接

https://docs.google.com/document/d/1PBUV4tuF5dWObzlGeOY6jO0pFFFabnRFPQwcreY4UmU/edit?usp=sharing