RNN+Attention机器翻译说明文档

# 一、工程结构说明

.

| inference.py 模型预测文件

│ LICENSE

│ README.md

│ seq2seq.py 模型训练文件

│ utils.py 模型数据预处理文件

│

├─CBOW 词向量预训练模型

│ │ cbow.py

│ │ data\_parser.py

│ │ test\_embededword.py

│ │

│ └─data

│ cn.txt

│ en.txt

│

├─checkpoints 训练好的模型

│ checkpoint

│ dev.data-00000-of-00001

│ dev.index

│ dev.meta

│

├─data 训练及测试数据

│ cn.test.txt 中文测试集

│ cn.txt 中文训练集

│ cn.txt.ebd.npy 中文预训练词向量

│ cn.txt.vab 中文词典

│ en.test.txt 英文测试集

│ en.txt 英文训练集

│ en.txt.ebd.npy 英文预训练词向量

│ en.txt.vab 英文词典

│

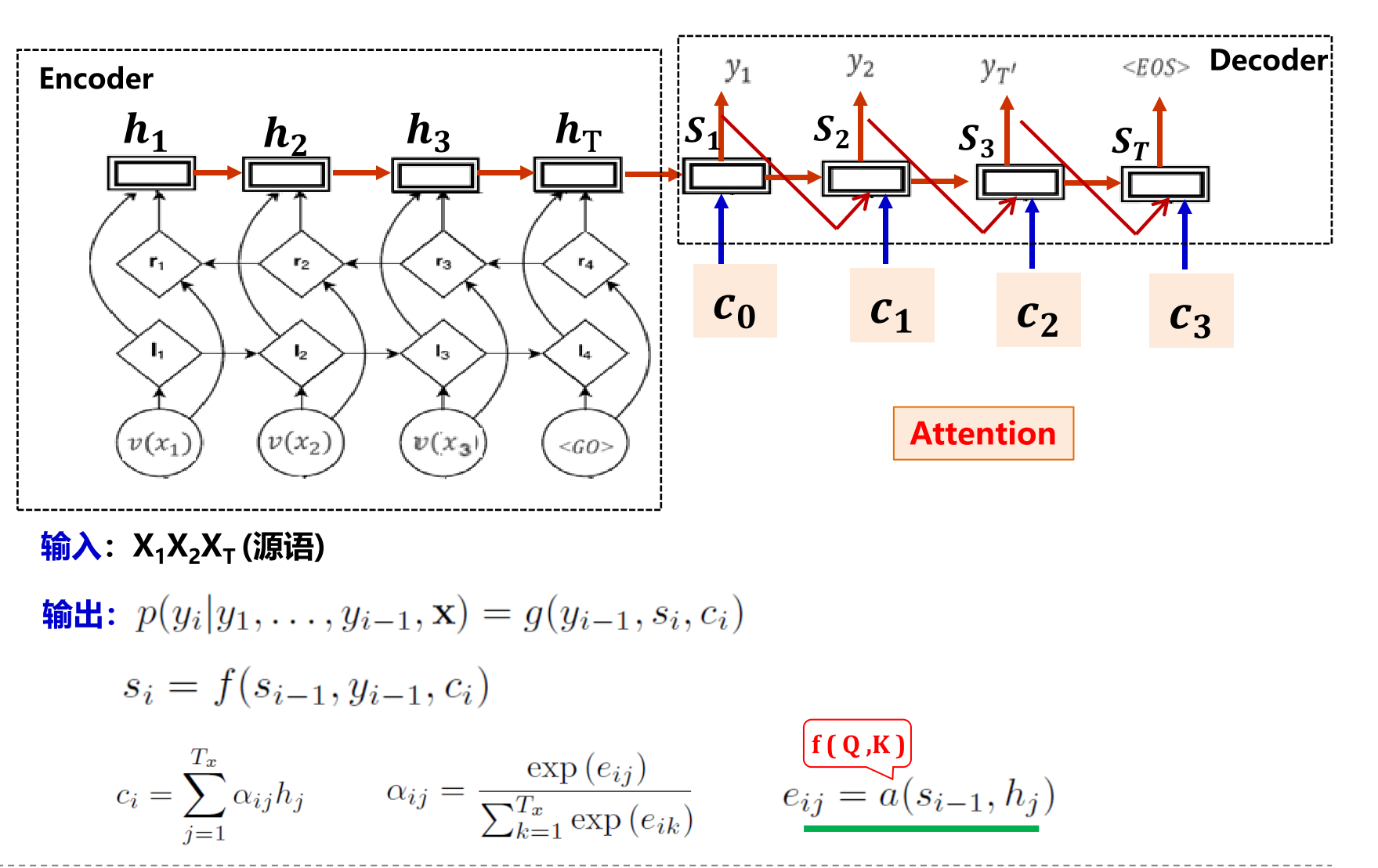
└─doc 参考文献

neural machine translation by jointly learning to align and translate.pdf

<https://github.com/jmhIcoding/machine_translation/tree/devolpe> 记录了本人开发这个项目的过程。

# 二、模型框架

使用Encoder+Decoder框架，在Decoder部分使用《Neural machine translation by jointly learning to align and translate》论文里面提到的基于一个小型神经网络来计算权值的Attention机制。



# 三、数据预处理

代码细节参见：utils.py

## 3.1 预训练

使用之前词向量作业里面的CBOW模型,对语料文件进行预处理。得到中文和英文的词向量，词典，以及一些统计信息。

因为语料库比较小，因此在生成词典的时候，没有进行词频过滤。实验发现，如果过滤的话，很容易把“<unk>”输入到网络中。

统计信息主要是得到语料中，中文句子和英文句子的最大长度、最小长度。

**----------src corpus--------------------  
{'std': 1.1084102394696949, 'mean': 21.437810945273633, 'max': 23, 'min': 20}  
----------dst corpus--------------------  
{'std': 1.1049706194897553, 'mean': 28.491805677494877, 'max': 30, 'min': 27}**

中文句子的最大长度为23，英文句子最大长度为30.这些信息需要在后面的网络中使用。

## 3.2 划分训练集和测试集

从所给的语料库中，抽取出其中的31个句子以及对应的译文作为测试集，单独保存成文件。

## 3.3 数据集文件夹目录结构

├─data

│ cn.test.txt 中文测试集

│ cn.txt 中文训练集

│ cn.txt.ebd.npy 中文预训练词向量

│ cn.txt.vab 中文词典

│ en.test.txt 英文测试集

│ en.txt 英文训练集

│ en.txt.ebd.npy 英文预训练词向量

│ en.txt.vab 英文词典

# 四、模型训练代码

参见seq2seq.py,模型同时构建了训练和预测的图结构。

训练过程的损失：

···

Epoch 324 - Valid Loss: 0.3102

Epoch 325 - Valid Loss: 0.3023

Epoch 326 - Valid Loss: 0.3137

Epoch 327 - Valid Loss: 0.2787

Epoch 328 - Valid Loss: 0.2124

Epoch 338 - Valid Loss: 0.0755

Epoch 339 - Valid Loss: 0.0738

Epoch 340 - Valid Loss: 0.0703

Epoch 341 - Valid Loss: 0.0740

Epoch 342 - Valid Loss: 0.0670

····

Epoch 497 - Valid Loss: 0.0048

Epoch 498 - Valid Loss: 0.0047

Epoch 499 - Valid Loss: 0.0047

Model Trained and Saved

# 五、模型测试

代码细节参见inference.py，代码通过tf.Saver.restore来载入训练得到的参数，以及网络结构。然后在通过sess.run里面给占位符填充好需要翻译的句子，跑模型得到结构即可。

一些好的翻译结果：

{'src': '新华社 北京 3 月 1 日 电 59 岁 的 原 黑龙江省 省长 田凤山 今天 被 任命 为 国土 资源部 部长 <END> <END>'}

{'dst': '<START> beijing , 1 mar ( xinhua ) -- tian fengshan , former heilongjiang governor who is 59 years old , was appointed minister of land and resources today <END>'}

Next Line

{'src': '出生 于 黑龙江省 肇源县 的 田凤山 1961 年 7 月 参加 工作 , 1970 年 3 月 加入 中国 共产党 <END> <END> <END>'}

{'dst': '<START> tian fengshan , who was born in zhaoyuan county , heilongjiang province , took part in work since july 1961 and joined the cpc <END> <UNK> <UNK> <UNK> <UNK>'}

Next Line

{'src': '这 应该 是 经济 发展 的 自然 扩散 过程 , 和 60 年代 “ 三线 建设 ” 工厂 搬家 是 不同 的 <END>'}

{'dst': '<START> this should be a natural process set off by economic development ; the " third tier construction " of the 1960s involving factory relocation was something entirely different <END>'}

Next Line

看起来，还是挺通顺的。