# 智能客服聊天机器人

|  |  |
| --- | --- |
| **文档编号：** |  |
| **撰 写 人：** | 新技术研发部 |
| **创建时间：** | 2017年11月13日 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档历史记录 | | | | |
| 序号 | 主要更改内容 | 版本号 | 更改人 | 更改时间 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 |  |  |  |  |
| 16 |  |  |  |  |
| 17 |  |  |  |  |
| 18 |  |  |  |  |
| 19 |  |  |  |  |
| 20 |  |  |  |  |

目录

[二期机器学习新技术落地解决方案 1](#_Toc23241)

[1. 引言 4](#_Toc12636)

[1.1 文档编写目的 4](#_Toc30273)

[1.2 背景 4](#_Toc13861)

[1.3 词汇表 4](#_Toc20852)

[1.4 参考资料 4](#_Toc6791)

[2. 系统需求 4](#_Toc10873)

[2.1预测分析系统 5](#_Toc17550)

[2.1.1 基于经纬度的预测分析 5](#_Toc13885)

[2.1.2 基于用户的行为数据K-means聚类分析 5](#_Toc2138)

[2.1.3基于用户行为数据和借款申请信息的K-means聚类分析 6](#_Toc10168)

[2.1.4基于深度学习网络模型对用户还款违约情况进行预测 7](#_Toc9966)

[2.2 反欺诈系统 7](#_Toc30216)

[2.2.1 目标 7](#_Toc13432)

[2.2.2 数据来源 7](#_Toc32)

[2.2.3 特征工程 8](#_Toc10670)

[2.2.4 算法选择 8](#_Toc25386)

[2.2.5 算法优化 9](#_Toc16689)

[2.2.6 预期目标评估 9](#_Toc16598)

[3. 其他支持 10](#_Toc2344)

[3.1 数据支持 10](#_Toc23840)

# 引言

## 1.1 文档编写目的

通过此文档，让相关人员了解到我们目前智能聊天机器人项目的一些情况，适用人群为业务部门需求人员，大数据开发人员，架构师以及管理者等相关人员。

## 1.2 背景

基于北京分公司的智能在线客服的需求和现有的客服人员与客户的聊天语料，利用深度学习模型LSTM(Long Short Term 网络),来训练一个深度学习模型，能满足一般的客户业务咨询的诉求，减轻客服人员的工作压力。

## 1.3 词汇表

N/N

## 1.4 参考资料

N/N

# 系统需求

根据业务需求，基于自然语言处理，中文分词，词向量，LSTM网络,通过公司已经积累的客服和客户的问答语料作为模型的输入，来训练模型。最终希望借此模型，我们可以让模型来模拟客服人员，对客户所提问题进行回答。

## 2.1 LSTM模型简介

长短期记忆网络——通常也被简称为 LSTM——是一种特殊类型的 RNN，能够学习长期的依赖关系。这个网络由 Sepp Hochreiter 和 Jürgen Schmidhuber 在 1997 年提出，并加以完善与普及。因为语言类的深度学习任务，涉及上下文信息非常多，语言类的深度学习模型难以训练，LSTM可以解决学习过程中的长期依赖关系，因此LSTM 在语音识别，机器翻译，人机对话等领域取得了很大的发展。

所有的 RNN 都具备重复性的链条形式，而在标准的 RNN 中，这个重复模式都有着一个简单的结构，比如单层的 tanh 层（图1）。

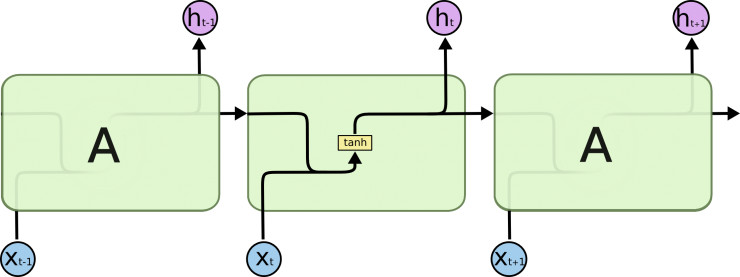


图 1

LSTM 也有这种结构化的链条，但重复性模块有着不同的结构。与 RNN 不同的是，LSTM 中有一个四层的网络，并以一种特殊的方式进行交互（图2）。

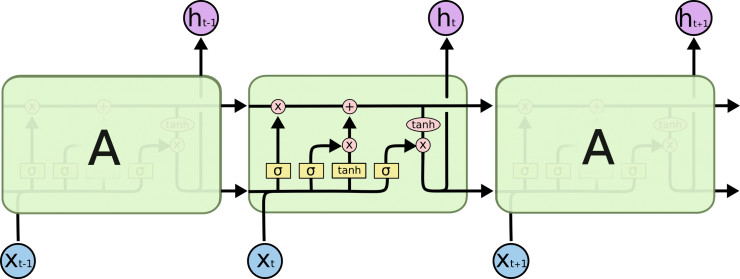


图 2

### 2.1.1 中文语料处理

不同于英文语料的词汇间自带空格，中文语料处理比较麻烦。我们首先把语料准备成问题语料和答案语料。这两份语料每一行分别代表一个问题和该问题的答案，行号需一一对应，文本编码使用utf-8。借助分词工具，我们可以把语料文档转换成一个个独立的文字。通过对语料做集合操作，我们找出这个语料有多少个不同的文字。然后把这个集合里的文字做成一个词汇映射。然后利用这个词汇映射，把之前的语料进行数字化转换和分词。每行转换后的元素为数字，该数字对应词汇映射里的文字，这样，我们就把文字进行词向量化。示例代码如图3 所示

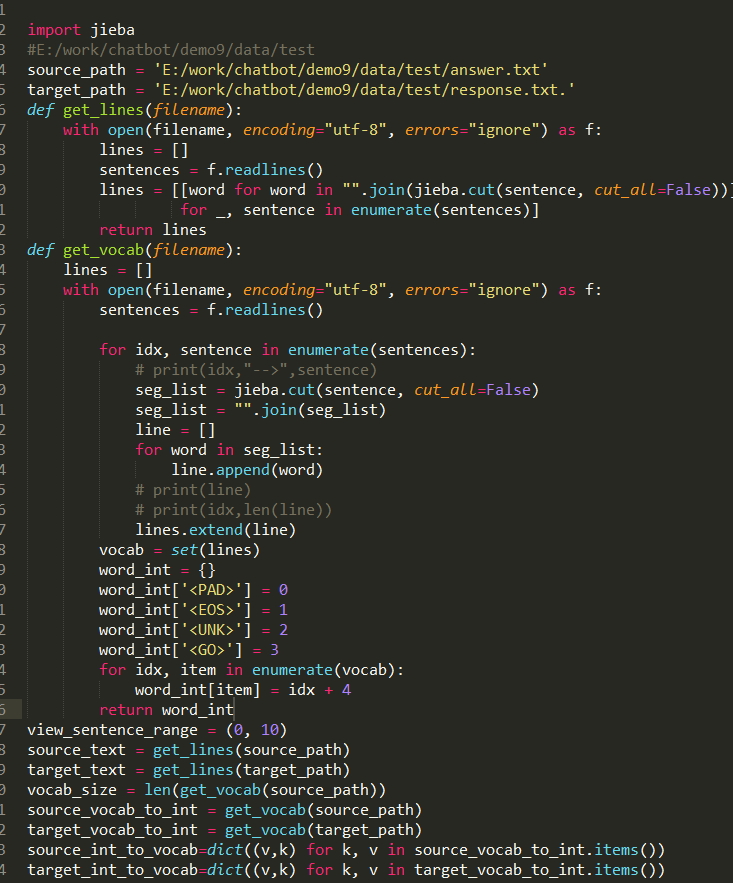


图 3

### 2.1.2 模型建立

该模型包含以下几个模块，分别是：

# - `model\_inputs`

# - `process\_decoder\_input`

# - `encoding\_layer`

# - `decoding\_layer\_train`

# - `decoding\_layer\_infer`

# - `decoding\_layer`

# - `seq2seq\_model`

对于`model\_inputs`，主要用来为序列输入，序列输出，学习速率，输入序列长度，目标序列长度等设置占位符。示例代码如图4所示

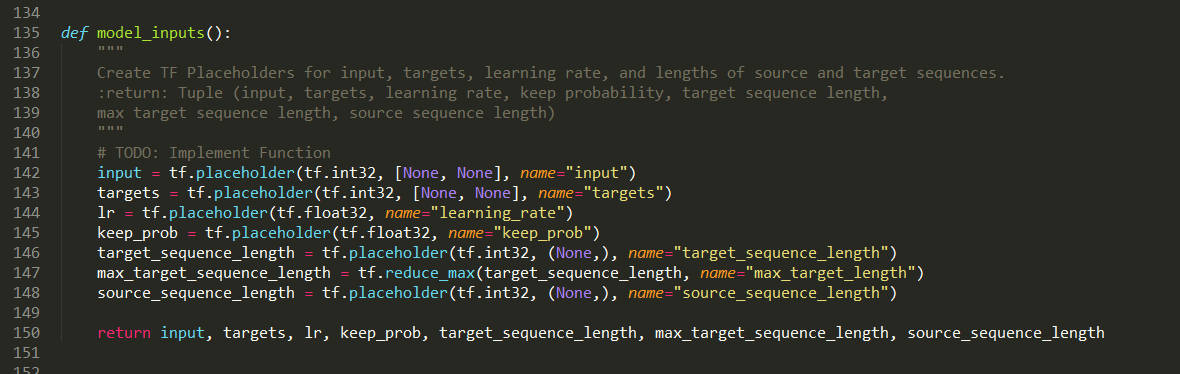


图 4

对于`process\_decoder\_input`,对待编码的目标数据，进行预处理。示例代码如图5所示

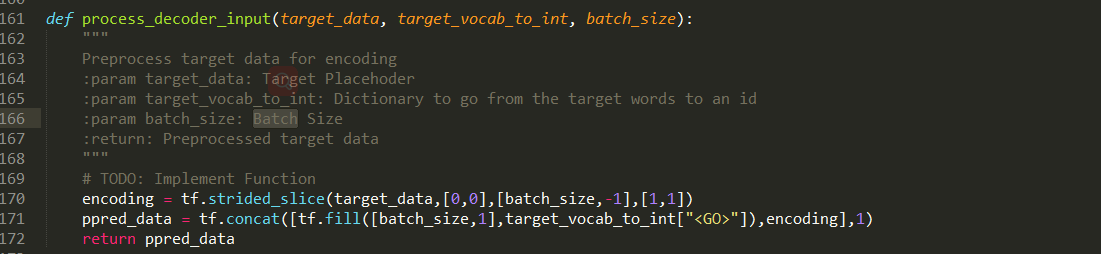


图 5

对于`encoding\_layer`，创建编码层，示例代码如图6所示

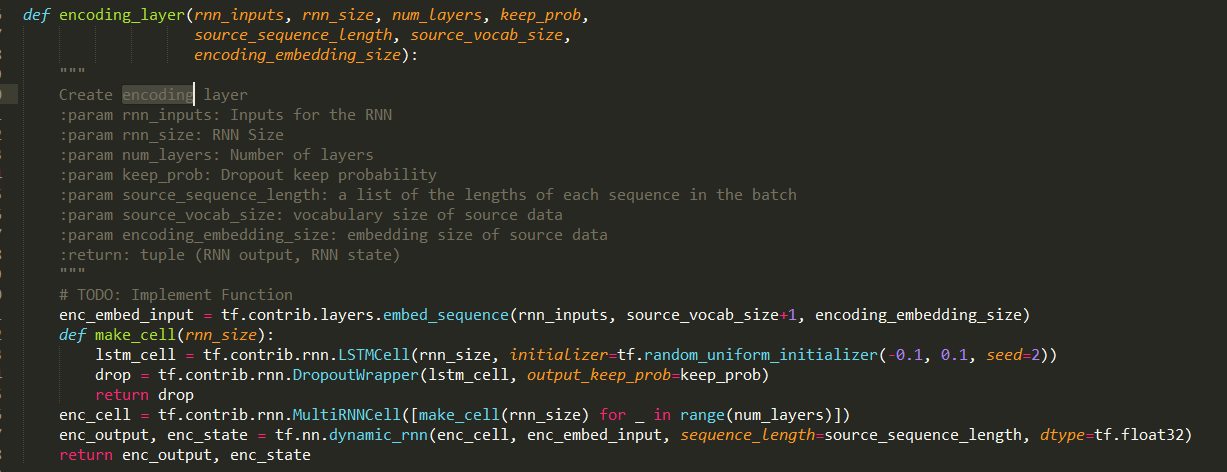


图 6

对于`decoding\_layer\_train`,创建用于训练模型的解码层。示例代码如图7所示

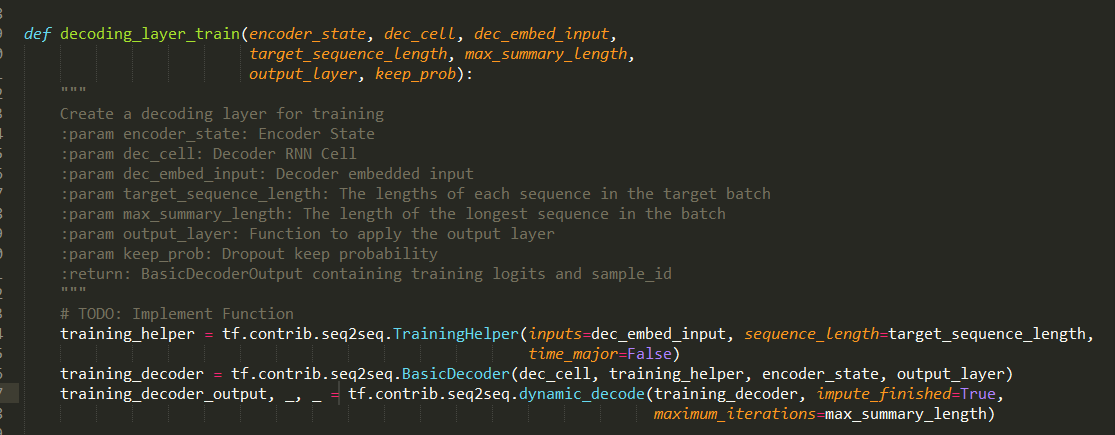


图 7

对于`decoding\_layer\_infer`，用于创建一个进行推理的解码层。示例代码如图8所示

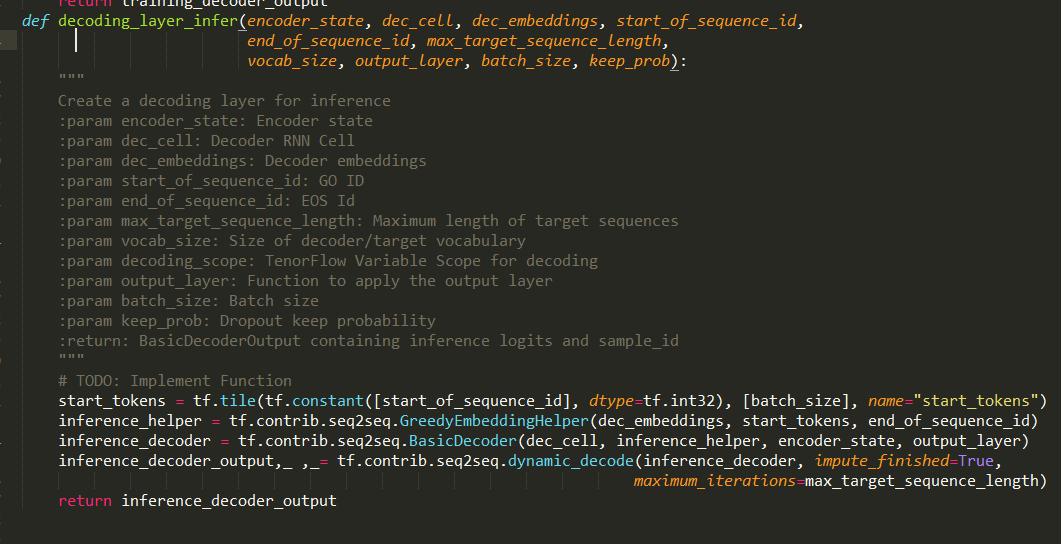


图 8

对于`decoding\_layer`,创建编码层，示例代码如图9所示

其中需要的参数：

:param dec\_input: Decoder input

:param encoder\_state: Encoder state

:param target\_sequence\_length: The lengths of each sequence in the target batch

:param max\_target\_sequence\_length: Maximum length of target sequences

:param rnn\_size: RNN Size

:param num\_layers: Number of layers

:param target\_vocab\_to\_int: Dictionary to go from the target words to an id

:param target\_vocab\_size: Size of target vocabulary

:param batch\_size: The size of the batch

:param keep\_prob: Dropout keep probability

:param decoding\_embedding\_size: Decoding embedding size

:return: Tuple of (Training BasicDecoderOutput, Inference BasicDecoderOutput)

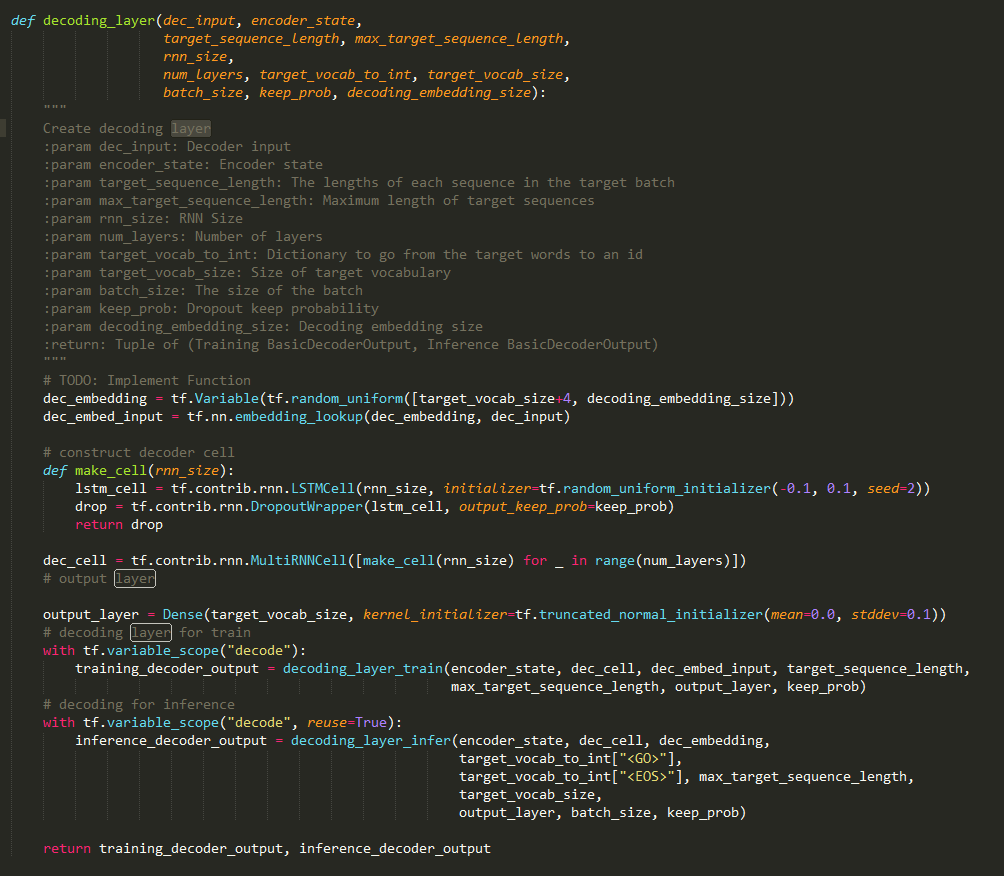


图 9

对于`seq2seq\_model`，创建一个基于深度神经网络的“序列-序列”模型。示例代码如图10

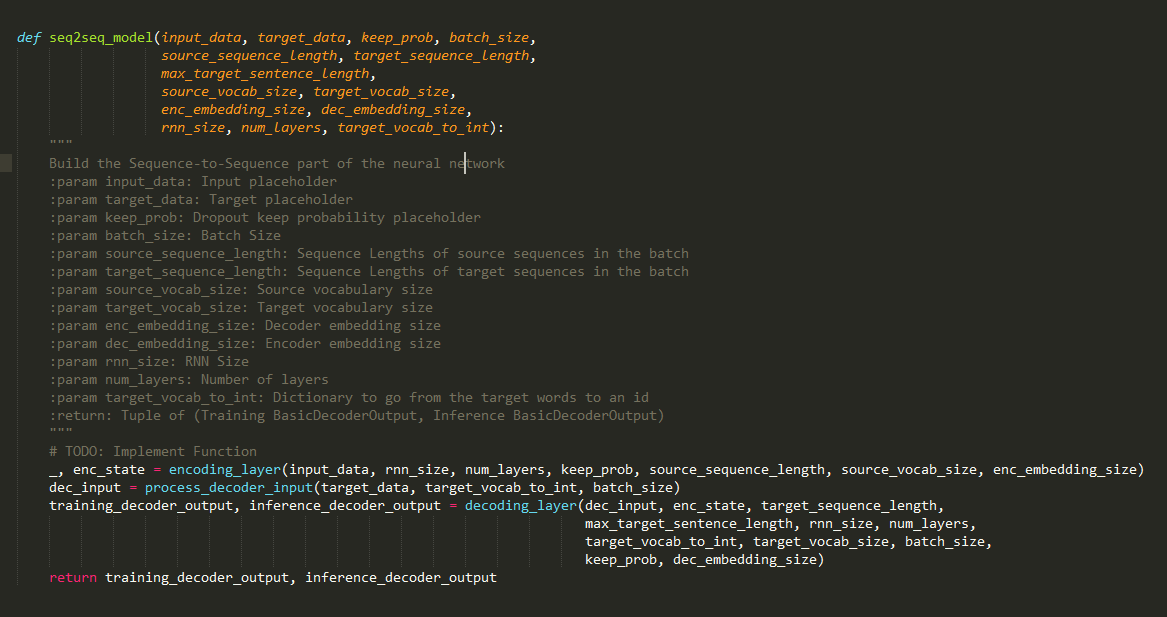


图 10

至此，模型建立的步骤基本完成。

### 2.1.3 模型参数设置及启动

设置模型超参

# Number of Epochs

epochs = 10

# Batch Size

batch\_size = 128

# RNN Size

rnn\_size = 512

# Number of Layers

num\_layers = 6

# Embedding Size

encoding\_embedding\_size = 1024

decoding\_embedding\_size = 1024

# Learning Rate

learning\_rate = 0.001

# Dropout Keep Probability

keep\_probability = 0.7

display\_step = 20

设置读取语料和检查点，如图11所示

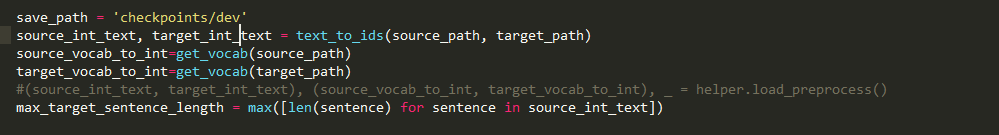


图 11

设置相关计算张量，并设置张量图，如图12 所示

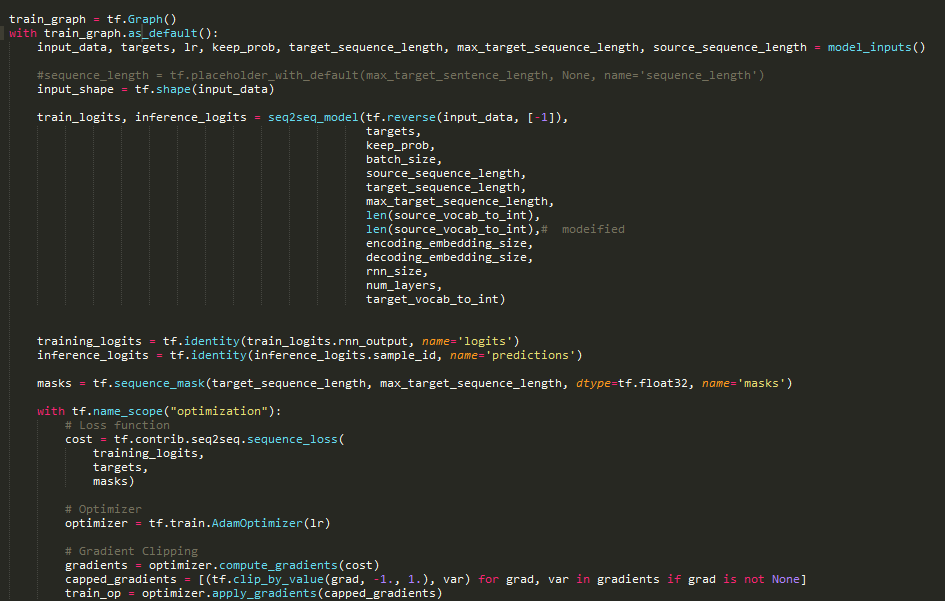


图 12

设置 pad， batch函数，对语料文件的每一句语料做处理，pad函数可以确保每条句子的长度一致，batch函数可以确保对语料文件进行切割，分批次灌入模型训练，避免全部语料灌入时，内存占用过大。示例代码如图13所示

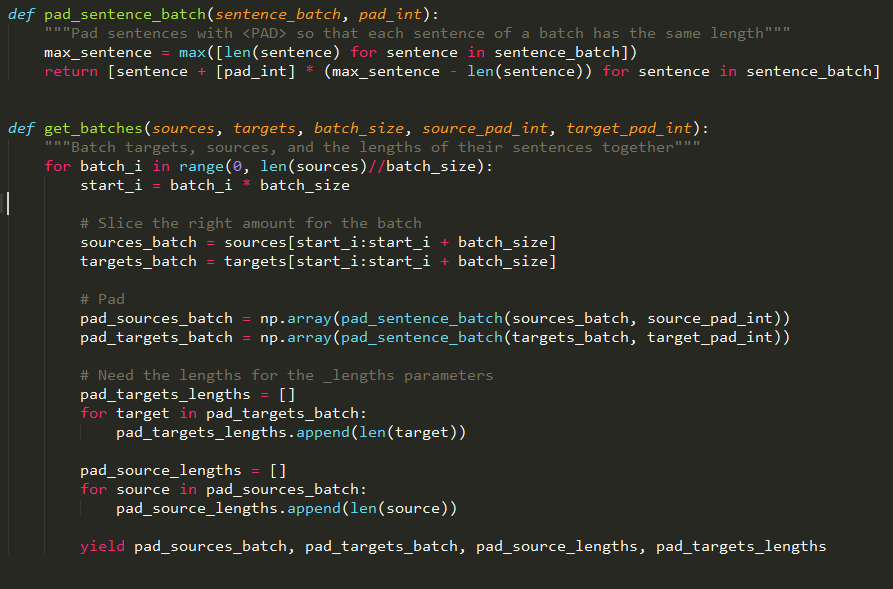


图 13

设置模型准确度估计函数，示例代码如图14所示

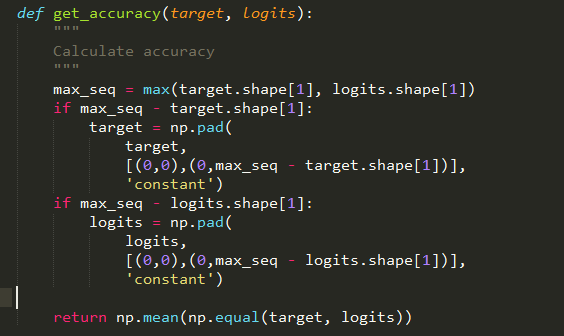


图 14

### 2.1.4模型训练及模型保存

设置好模型的参数后，我们就可以开始训练模型。在训练模型前，先进行数据切割操作，防止一次性灌入模型的数据量太大，示例代码如图15所示

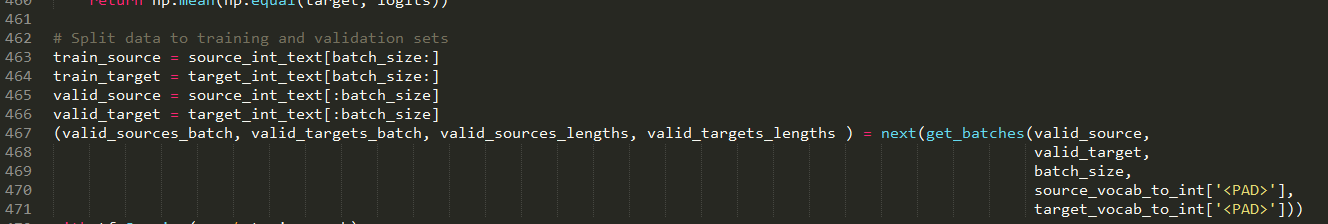


图 15

接着建立新的会话，初始化所有变量，开始训练模型，为了方便的看到模型训练过程中的，参数变化对模型的影响，我们设置打印相关的信息。最后保存模型，示例代码如图16所示

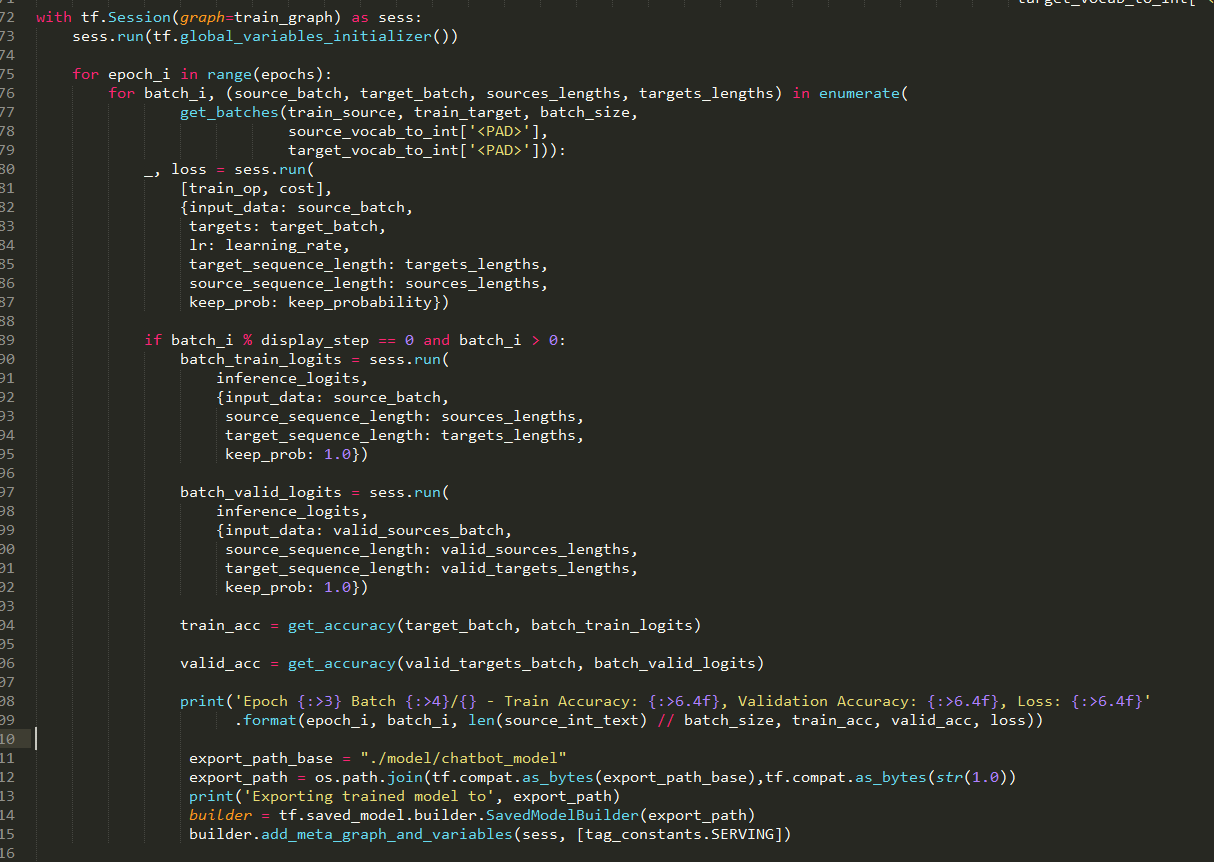


图 16

### 2.1.5模型导出

代码构建完毕，接下来使用tensorflow-serving-api，来训练模型并导入模型到指定的目录

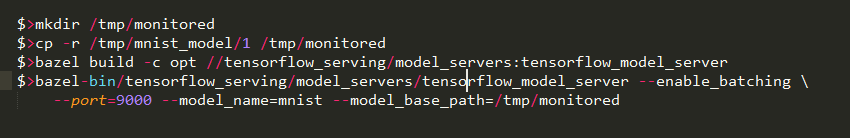
使用Python命令：

python tensorflow\_serving/example/chatbot.py /tmp/chatbot\_model

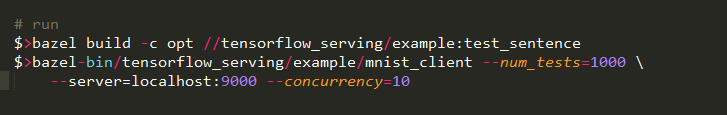
注：/tmp/chatbot\_model 是导出模型的目录

### 2.1.6 测试并提供服务

使用一下命令完成测试：



运行test数据：



注：test\_sentence需要进行切词，词向量化操作。

# 其他

由于语料和本人水平有限，模型还有不足之处，还需花时间改善，不足之处还请指出。

E-mail:672087110@qq.com,tel:15818603403