# 详细设计说明书

# 1.引言

本文档名为“智能聊天机器人详细设计说明书”，版本为1.0。

# 系统的结构

智能聊天机器人系统总共分为三大模块，分别为语音识别模块：识别用户的语音输入并进行拼音处理；AI图灵模块：进行内层逻辑运算，包括语音分析，情感分析等；语音合成模块：将得到的结果通过语音合成的方式输出。

# 模块一：语音识别

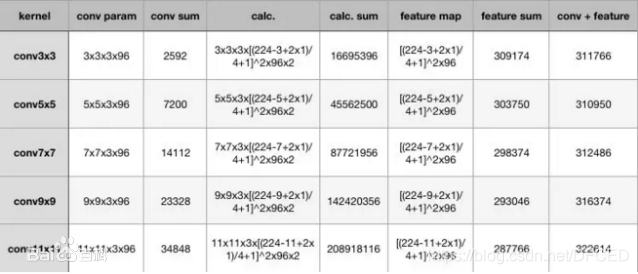
3.1简介

本模块语音识别部分通过采用卷积神经网络（CNN）和连接性时序分类（CTC）方法，使用大量中文语音数据集进行训练，将声音转录为中文拼音，并通过语言模型，将拼音序列转换为中文文本。

我们使用的的声学模型采用了深度全卷积神经网络，直接将语谱图作为输入。模型结构上，借鉴了VGG，这种网络模型有着很强的表达能力，可以看到非常长的历史和未来信息，相比RNN在鲁棒性上更出色。在输出端，这种模型可以和CTC方案可以完美结合，以实现整个模型的端到端训练，将声音波形信号直接转录为中文普通话拼音序列。在语言模型上，通过最大熵隐含马尔可夫模型，将拼音序列转换为中文文本。

3.2算法流程

1. 特征提取 ：将普通的wav语音信号通过分帧加窗等操作转换为神经网络需要的二维频谱图像信号，即语谱图。
2. 声学模型 ：基于Keras和TensorFlow框架，使用这种参考了VGG的深层的卷积神经网络作为网络模型，并训练。



（3）CTC解码 ：在语音识别系统的声学模型的输出中，往往包含了大量连续重复的符号，因此，我们需要将连续相同的符合合并为同一个符号，然后再去除静音分隔标记符，得到最终实际的语音拼音符号序列。



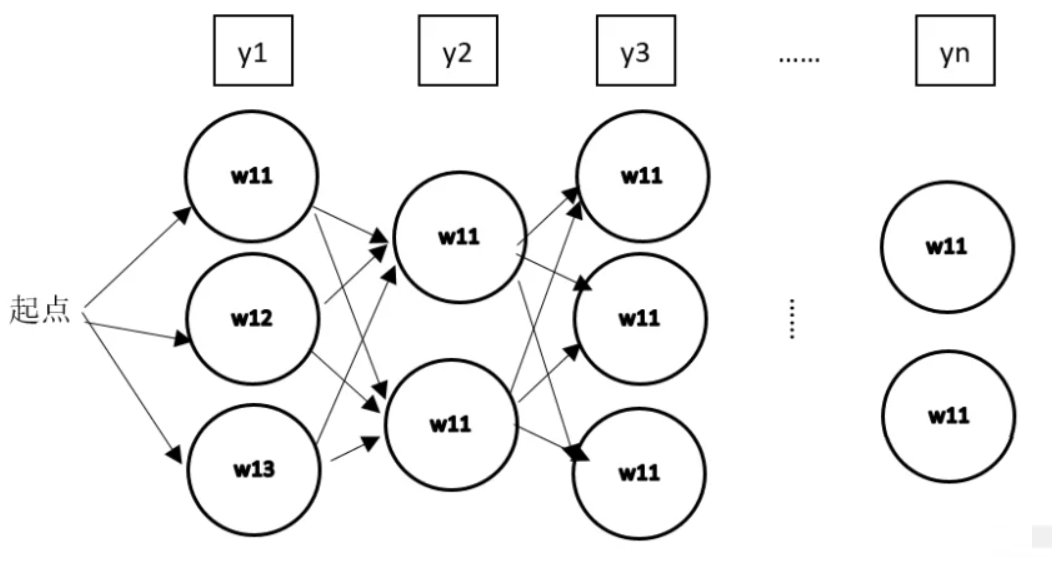
（4）语言模型 ：使用统计语言模型，将拼音转换为最终的识别文本并输出。拼音转文本的本质被建模为一条隐含马尔可夫链，这种模型有着很高的准确率。

拼音转汉字的算法是动态规划，跟寻找最短路径的算法基本相同。我们可以将汉语输入看成一个通信问题，每一个拼音可以对应多个汉字，而每个汉字一次只读一个音，把每一个拼音对应的字从左到有连起来，就成为了一张有向图，它被称为网格图或者篱笆图。

y1,y2,…,yn是输入的拼音串，w11,w12,w13是第一个音y1的候选字，w21,w22是y2对应的候选字，以此类推。整个问题就变成在有向图中寻找从起点开始，到终点概率最大的路径，我们可以使用各种最短路径算法来实现，这里使用维特比算法来进行语音到文本的解码。

维特比算法是先计算第一步的概率，然后将概率按大小排序，剔除掉概率过低的路径，然后再走第二步，再剔除掉概率过低的路径，以此类推。

反复执行，直到最终到达路径终点。最后，我们可以得到一个概率最大的路径，即概率最大的一个句子，在算法执行过程中，我们实际还可以得到一系列概率相对较小的路径。



3.3依赖环境

（一）系统环境

Windows系统

（二）python以及依赖包环境

python3.5及以上版本；

TensorFlow1.14版本以上的版本；

Keras2.2.4、MatPlotLib3.0.3、h5py2.10.0；

wave、python\_speech\_features和requests包

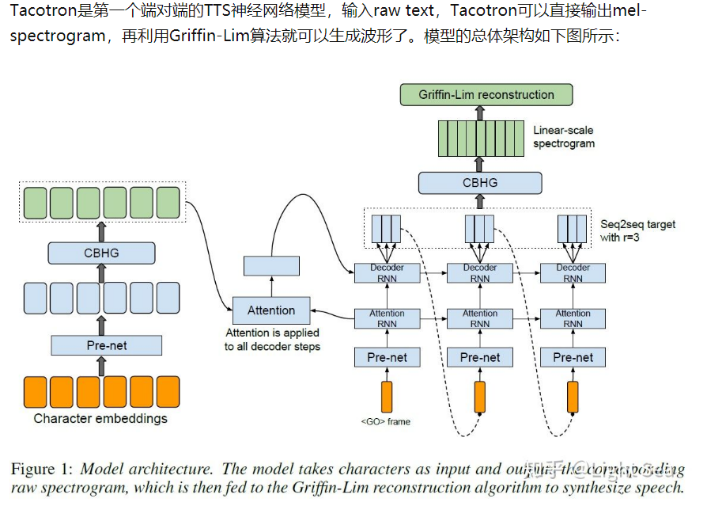
# 模块二：图灵AI

内层逻辑直接调用图灵AI的api，从而实现内部的回答逻辑

# 模块三：语音合成

5.1简介

采用TACOTRON 模型，TACOTRON是一个端到端的TTS模型，模型核心是seq2seq + attention。模型的输入为一系列文本字向量，输出mel-spectrogram, 然后在使用Griffin\_lim算法生成对应音频。模型结构如下图：



模型结构：

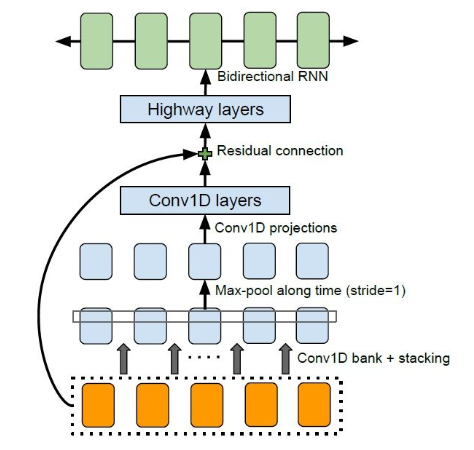
·CBHG模块

·encoder

·decoder

·post-processing net and waveform synthesis

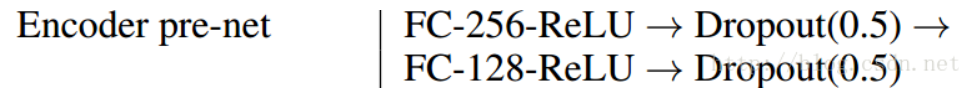
5.2 CBHG模块



其中使用了卷积+highway+残差链接+双向GRU的组合，输入序列，输出同样也是序列。

5.3 Encoder模块

输入被CBHG处理之前还需要经过pre-net进行预处理，意图是提升模型的泛化能力，以及加快收敛速度。embeding layer之后是一个encoder pre-net模块，它有两个隐藏层，层与层之间的连接均是全连接；第一层的隐藏单元数目与输入单元数目一致，第二层的隐藏单元数目为第一层的一半；两个隐藏层采用的激活函数均为ReLu，并保持0.5的dropout来提高泛化能力.



5.4 Decoder模块

decoder模块主要分为三部分：

- pre-net

- Attention-RNN

- Decoder-RNN

Pre-net的结构与encoder中的pre-net相同，主要是对输入做一些非线性变换。

Attention-RNN的结构为一层包含256个GRU的RNN，它将pre-net的输出和attention的输出作为输入，经过GRU单元后输出到decoder-RNN中。

Decode-RNN为两层residual GRU，它的输出为输入与经过GRU单元输出之和。每层同样包含了256个GRU单元。第一步decoder的输入为0矩阵，之后都会把第t步的输出作为第t+1步的输入。(这里paper中使用了一个trick，就是每次decoder的时候，不仅仅预测1帧的数据，而是预测多个非重叠的帧。因为就像我们前面说到的提取音频特征的时候，我们会先分帧，相邻的帧其实是有一定的关联性的，所以每个字符在发音的时候，可能对应了多个帧，因此每个GRU单元输出为多个帧的音频文件。

5.5 预训练模型

预训练模型

标贝数据集100K步模型

训练集：取20%

测试集：取80%

训练结果：train-loss：loss在0.49

Test-loss：loss在0.51