基于 SLNet 的半监督手语识别多模态情感交互系统

北京师范大学附属实验中学

高三 （16 班）宋卓卿

二零二一年九月四日

摘要

习近平主席强调，残疾人是一个特殊困难的群体，需要格外关心、格外关注。残障人士需要融入社会，如何减少残障人士与社会沟通的障碍，这个问题值得关注。此外，由于面临工作和生活中的种种不便，残障人士比普通人更容易产生心理问题，如何对其提供情感安抚也是亟需解决的问题之一。基于以上背景，本项目针对语言障碍人士这一特殊社会群体，研发了一套基于 SLNet 的半监督手语识别多模态情感交互系统，实现了手语识别与情感安抚等功能。

本项目的主要创新点如下：

1.提出一套完整的全自动手语识别数据集录制系统。这套系统能够根据用户需求更改和添加新的手势，能够录制多种环境下的手语数据，提高手语识别系统的识别率。

2.首次将基于伪标签的半监督学习引入到情感交互系统中。针对监督学习所带来的繁重数据标注问题，提高后续在线识别的准确性，将基于伪标签的半监督学习引入到情感交互系统中能够极大的提升系统在线识别的准确性和鲁棒性。

3.从豆瓣爬取并整理夸夸数据集，对基于 seq2seq+attention 框架的情感安抚机器人进行训练。同时为了解决传统单一 attention 机制在信息丰富度上缺乏的缺陷，提出了一种针对本聊天机器人的多抽头组合 attention 机制对结果进行优化。

情感聊天机器人能够在对语言障碍人士提供手语交流服务的同时提供情感安抚，让人工智能技术实现爱的给予，去抚慰这一特殊社会群体。

1. 前言

我是零零后，从小接受的来自社会学校家庭的教育就是“以爱育爱”，要做一个对社会有用的人。十六年来，我去过很多福利机构，看到过很多和我们不一样的孩子，智障、抑郁、自闭、脑瘫……他们活在自己的世界里，他们不说话或表达不清，他们更多的是用肢体语言表达自己的情感和需求。每次看到他们，我都在想除了去看望他们，看护他们，是不是还有其他的方法帮助他们。当我有了一定能力后，一定要去付诸行动。

2018 年 10 月我看到一则新闻，星巴克(Starbucks)在美国的首家手语门店开业了，从点单到制作咖啡，到处都能看到美式手语（American Sign Language）的影子。这家手语门店位于华盛顿特区，临近加拉德特大学（这所大学是聋人和弱听人士高等院校）。虽然其他星巴克门店的部分员工可以接受顾客用手语点单，但是这家手语门店里的所有员工都精通美式手语。开设手语门店的灵感源自吉隆坡的一家星巴克门店，这家店在 2016 年雇佣了 9 位听障员工。于是美国星巴克的员工参观了吉隆坡的门店，了解了设计细节，然后为华盛顿特区的门店设计制定了最终方案。

星巴克手语咖啡馆

这一则新闻，对我触动很深。我了解到世界上有很多想为这些特殊人群提供帮助的人。爱是一种理解，更是一种付出！对于这些存在语言障碍的人而言，不仅仅需要理解他们的语言，而且还需要给他们提供一定的情感安抚。基于手语是他们最主要的交流方式这个出发点，我设计了一套《基于 SLNet 的半监督手语识别的情感交互系统》。

这套交互系统分为两个部分：一是基于 SLNet 的半监督神经网络的手语识别系统，二是基于 seq2seq+attention 的情感聊天机器人系统。这套系统首次解决了传统手语识别系统缺乏语义信息，传统聊天机器人无法结合用户的视觉信息的缺陷。本文首次提出了针对手语识别的 SLNet 神经网络。

为了能够实现较好的识别率，我们设计了一套自动化手语数据集录制软件，在我们收集的数据集上，该网络能够达到 95.5%的识别率，而且在实际的环境中能够对用户的手部进行较好的跟踪。在情感聊天机器人方面，我们收集了豆瓣上的夸夸群信息，并结合青云数据集对基于 seq2seq+多抽头组合 attention 的神经网络进行了训练。为了解决 seq2seq+多抽头组合 attention 神经网络这类生成型神经网络在专业问题方面回答不是足够专业的问题，我们引入了基于检索式的问答，保证了聊天机器人在某些专业问题上仍然能够有出色的表现。此外，为了提升算法的性能，我们在训练 SLNet 的过程中采用了基于 pseudo label 的半监督方法,利用无监督产生的伪标签对算法进行了性能提升。

传统的聊天机器人只能通过手写或者语音识别和用户进行交互，对于各类有交流障碍的人士而言，他们缺乏这样的能力。这套系统能够通过前端的基 SLNet 的半监督神经网络的手语识别系统，提供视觉交互端，同时基于 seq2seq+多抽头组合 attention 的情感聊天机器人能够对他们进行情感安抚。而对于其他存在心理问题的人士，情感聊天机器人也可以提供同样的情感安抚服务。

2. 方法和结果

2.1. 绪论

2.1.1 背景技术

随着信息时代的到来，电子计算机和手机等设备被广泛地应用，这些数据产生了大量的文本，语音，图像数据。人工智能极大的推动了这些领域的进步。

机器学习实现人工智能技术的核心手段之一，在海量数据的支撑下，机器学习算法能够提取数据中的关键信息，对未来的信息进行预测[1]。图 2.1 展示了这三者之间的关系。

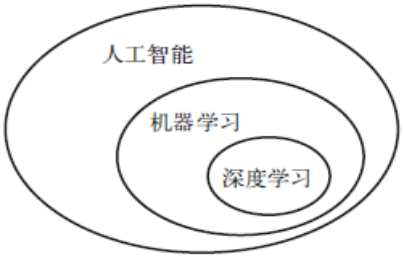


图 2.1 人工智能

机器学习和深度学习仅仅是人工智能的一个组成部分，人工智能包括诸如机器人学，控制等众多领域，在计算机出现的早期，通过程序员编写的硬编码规则可以实现人机对弈，但是这种规则并不是属于机器学习的范畴。

在计算机出现之后的很长一段时间，许多专家相信只要人类能够编写出足够鲁棒的规则系统，那么机器便可以实现与人类同等的智能水平，这一方法被称为符号主义人工智能（symbolic AI）。

符号主义的出现是解决了一些逻辑性问题,这些逻辑性问题通常是比较简单的问题，当符号主义遇到更加复杂的问题时，符号主义便很难实现同等的效果，例如图像处理，目标检测，图像分割等等。于是机器学习算法应运而生，成为了解决这类问题的主要途径。

在机器学习程序之前，人工设计的经典的程序通常是输入规则和数据进行运行，如最早的电子围棋，这类程序只能处理一部分的问题，随着机器学习的出现，可以通过输入数据和答案让机器学习算法自己去学习其中的规则，这套规则学习成功以后可以应用于其他的问题，达到自主学习和诊断的目的。

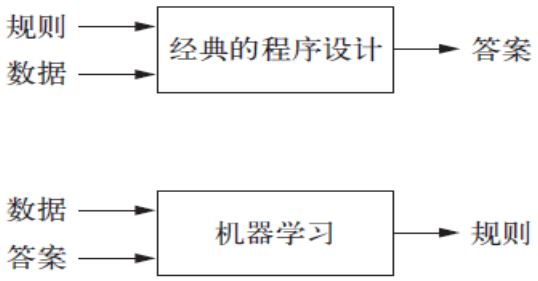


图 2.2 机器学习：一种新的编程范式

2.1.2 手语识别与聊天机器人背景

本项目基于前端的手语识别和后端的聊天机器人系统。

我们首先介绍手语识别的背景。在人类拥有语言之前，手势便已经是人类交流的重要手段之一，手势在人与计算机之间的信息交换方面扮演着举足轻重的角色。

手势交户是人机交互系统中的重要环节之一，其不仅可以实现快速的远程交户，而且能够通过体态操作远程的计算机和设备，在远程问诊，远程操作中的重要手段。提供一种更简单的方法控制复杂的虚拟环境。手势交流可以单独使用，也可以和其他的信息流进行，如图 2.3 所示。使用一只手的手势就可以完整的展示 26 个英文字符。

图 2.3 手势识别字母表

随着计算机视觉技术的进步，手势识别得到了越来越多的关注，学界和工业界都对手势识别进行了大量的研究。

传统的手势交互系统中存在着非常多的挑战，如其他障碍物的遮挡，外界环境的影响，人与人对手势表示的复杂性等。这些挑战对基于机器学习算法的手势交互系统而言是巨大的挑战。随着神经网络的兴起，越来越多的学者考虑将深度学习算法引入到手势识别中，Jakub 等人采 CNN 和 Softmax 结合的方法对手势进行识别， Chun-Jen 等人采用了自己训练的神经网络首次对三维手势进行识别。

下面介绍聊天机器人背景。第一投入实际使用的聊天机器人是 Eliza，Eliza 诞生于 1966 年，其主要针对的是心理疾病的治疗。虽然 Eliza 采用的是原始的非智能的人为制定回复规则的方法，还是得到了使用者的好评。

Unix 系统是世界上应用最广泛的操作系统之一，1988 年，为了方便大家对 Unix 操作系统进行学习，加州大学伯克利分校开发了一套基于 Unix 知识的聊天机器人系统，这套系统比 Eliza 更加智能，通过问答系统，用户能够获取 Unix 的常规操作信息，能够帮助用户更快的理解和操作 Unix 操作系统，这引发了巨大的关注。

为了实现聊天机器人通过图灵测试的终极目标，大量的学者将精力投身于聊天机器人的算法改进，数据收集之上。

时至今日，聊天机器人不再是简单的娱乐聊天工具，聊天机器人在疫情控制，智能客服，公共服务等众多领域。

2.2 手语识别相关技术介绍

2.2 .1 视觉相关知识介绍

2.2.1.1CNN 结构

卷积神经网络是神经网络的一个分支，其主要针对是针对视觉信息进行处理，通过提取视觉信息中的深度特征实现对图像的分类等任务。

作为最早应用于图像检测中的网络，LeNet5 实现了对手写支票字体的识别，图 2.4 展示了 Lenet5 网络的结构图。前几个阶段由两种类型的层组成：卷积层和池化层，最后为全连接层（图中 C 为卷积层，P 为池化层，F 为全连接层）。

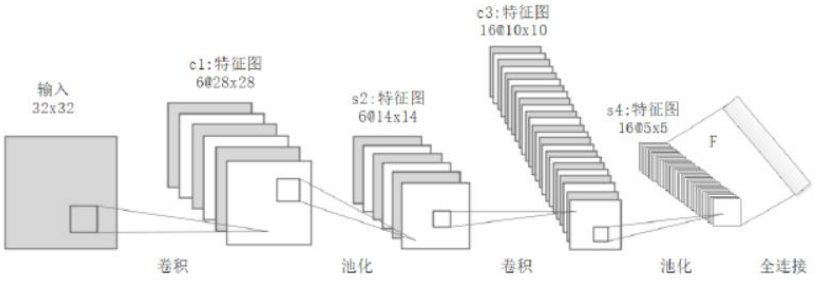


图 2.4 Lenet5 卷积神经网络[2]

2.2.1.2 卷积层

CNN 实现图像分类的关键步骤是对图像的特征进行提取，对图像的特征进行提取后采用分类器对特征进行分类，从而实现较传统算法更高的识别率。

卷积层是卷积神经网络中最重要的部分，卷积层通过卷积核对输入的图像信息进行卷积操作，卷积操作是指卷积核与图像上的局部信息进行矩阵乘法操作后得到新的特征，然后再进行整合，得到卷积后的特征图，图 2.5 展示了卷积运算的过程。

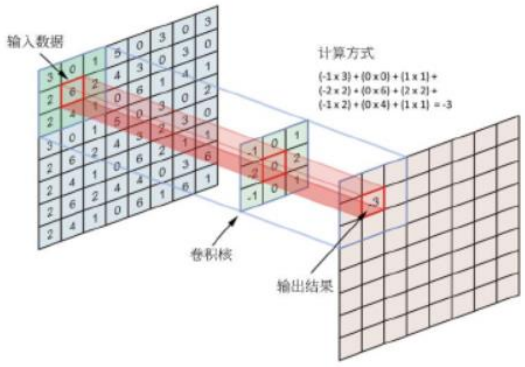


图 2.5 卷积计算示意图

2.2.1.3 池化层

池化层(Pooling Layer)通常出现在卷积层的后面，池化层的目的是减少卷积层输出的参数数目，实现对特征的二次提取。

我们常见的池化方式分为两种，一种是最大池化，最大池化会将卷积层输出的特征中选取其中最大的值进行保留，同时去除掉其他值。而最小池化则会将卷积层输出的特征中选取其中最小的值进行保留，同时去除掉其他的特征值。图 2.6 展示了两种池化方式的示意图。

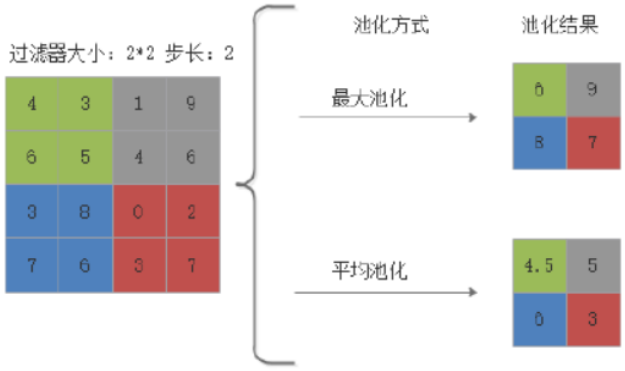


图 2.6 池化计算示意图

2.2.2 基于 SLNet 的半监督手语识别技术

手语识别模块分为四个步骤：第一步为数据集录制工作，第二步为数据集预处理，第三步为模型训练，第四步为在线测试。如图 2.7 所示。

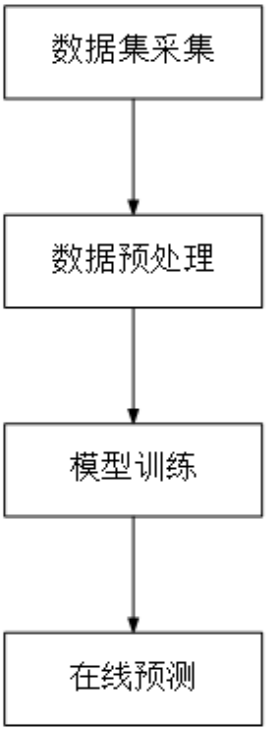


图 2.7 系统搭建流程

下面将对这些步骤分别进行说明：

第一步，数据集录制工作。神经网络属于数据驱动型算法，数据对模型的性能影响较大，为了提高模型的性能，在本项目中采用自己录制的数据集，数据集录制流程为用户需要将手放到一个指定的方框中。然后对输入的图像镜像翻转，因为采用的录制摄像头为前置摄像头。为了提高识别率需要将输入的彩色图像转换为灰度图像然后通过高斯滤波之后保存好图像。如图 2.8 所示。

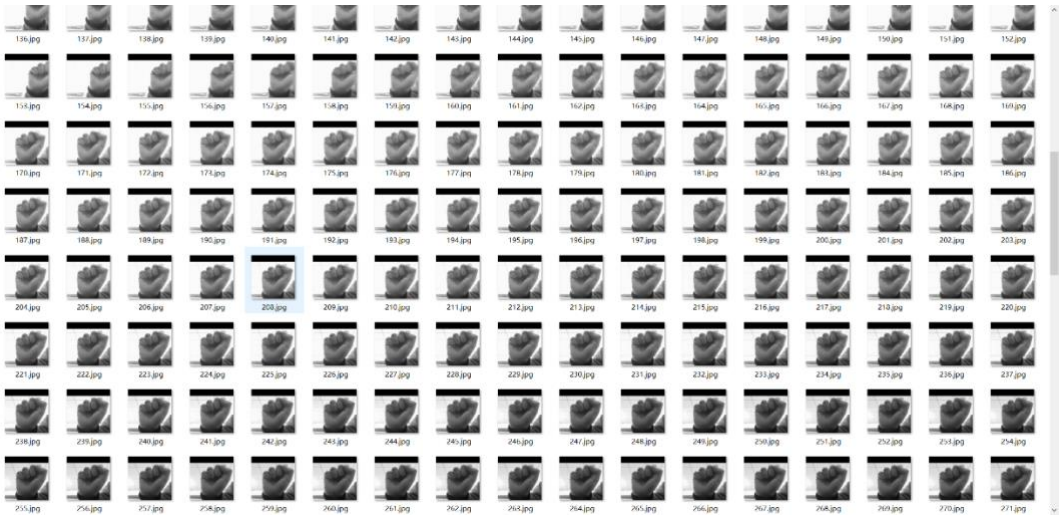


图 2.8 录制数据集

第二步，数据集预处理。需要将数据转成 numpy 矩阵的形式，然后将读入的图像重新调整为 96*96 的固定大小，这个尺寸也将是神经网络的最终输入尺寸，将标签从字符映射成为数字并保存在一个 numpy 矩阵中。

第三步，模型训练。首先采用 Python 中的 Pandas 模块进行数据集读取之后，先对数据集进行了归一化操作。归一化的目的是将图片从 0~255 的范围映射到 0~1 的范围内，加速数据集的收敛。在训练过程中我们同样需要对神经网络进行一些验证操作，保证神经网络不会出现过拟合，这个时候需要验证集的辅助，我们把训练集的后 500 张图片作为验证集，其他的作为训练集。每一个 epoch 结束后都对模型进行验证。

如下介绍手语识别的具体实现流程。我们首次提出了针对手语识别 SLNet 的半监督手语识别网络，具体的网络结构如图 2.9 所示。在该项目中输入的图像首先通过两个卷积层，然后再通过一个最大池化层，然后再通过一个 Dropout 层，Dropout 层的作用是防止过拟合。然后再通过两个卷积层，通过一个最大池化层，最后通过一个 Dropout 层。这种结构被重复了三次后通过 Flatten 层将二维数据压缩成为一维数据，最后经过一个全连接层和一个 Dropout 层，最后再经过一个全连接层得到最终的输出。

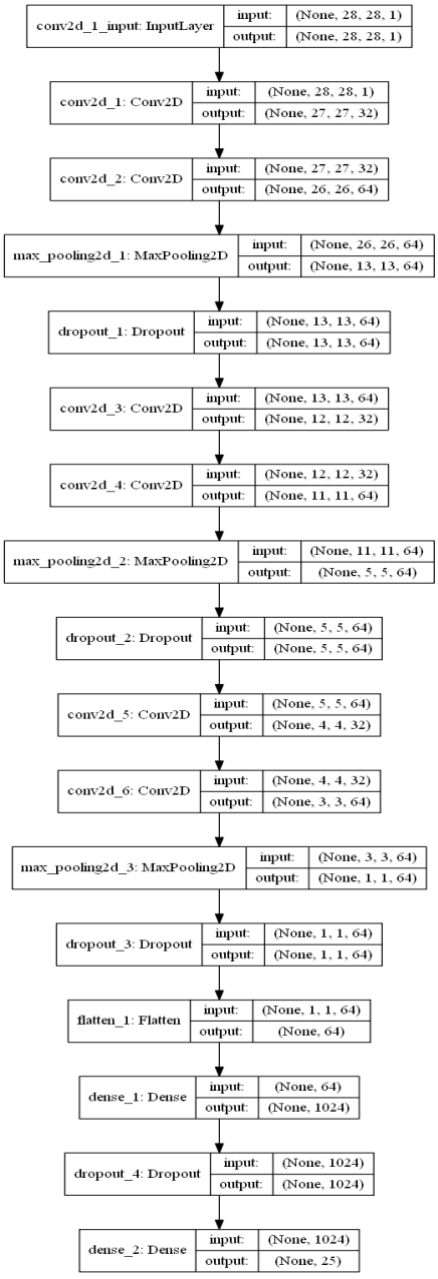


图 2.9 手语识别所采用的神经网络结构

训练的结果如图 2.10 所示，训练过程中的损失曲线和准确率曲线如图 2.11 所示。可以看到在验证集上的损失要小于在训练集上的损失，而与此同时在一个 epoch 之后训练集上的训练准确率接近 98%。可以看到模型最后在测试集上面的识别率为 0.950，在训练集上面的识别率为 0.9778。这种训练集的表现优于测试集的情况称位过拟合。

```
Epoch 1/10
26955/26955 [==============================] - 33s 1ms/step - loss: 0.0953 - acc: 0.9697 - val_loss: 0.0047 - val_acc: 0.9980
Epoch 2/10
26955/26955 [==============================] - 32s 1ms/step - loss: 0.0805 - acc: 0.9744 - val_loss: 0.0028 - val_acc: 0.9980
Epoch 3/10
26955/26955 [==============================] - 32s 1ms/step - loss: 0.0828 - acc: 0.9732 - val_loss: 0.0018 - val_acc: 1.0000
Epoch 4/10
26955/26955 [==============================] - 32s 1ms/step - loss: 0.0789 - acc: 0.9744 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 5/10
26955/26955 [==============================] - 33s 1ms/step - loss: 0.0783 - acc: 0.9752 - val_loss: 0.0031 - val_acc: 0.9980
Epoch 6/10
26955/26955 [==============================] - 32s 1ms/step - loss: 0.0828 - acc: 0.9743 - val_loss: 0.0010 - val_acc: 1.0000
Epoch 7/10
26955/26955 [==============================] - 32s 1ms/step - loss: 0.0679 - acc: 0.9782 - val_loss: 0.0024 - val_acc: 1.0000
Epoch 8/10
26955/26955 [==============================] - 32s 1ms/step - loss: 0.0784 - acc: 0.9760 - val_loss: 0.0013 - val_acc: 1.0000
Epoch 9/10
26955/26955 [==============================] - 31s 1ms/step - loss: 0.0649 - acc: 0.9794 - val_loss: 5.5888e-04 - val_acc: 1.0000
Epoch 10/10
26955/26955 [==============================] - 31s 1ms/step - loss: 0.0753 - acc: 0.9778 - val_loss: 8.9439e-04 - val_acc: 1.0000
0.9505019520356943
```

图 2.10 训练过程结果

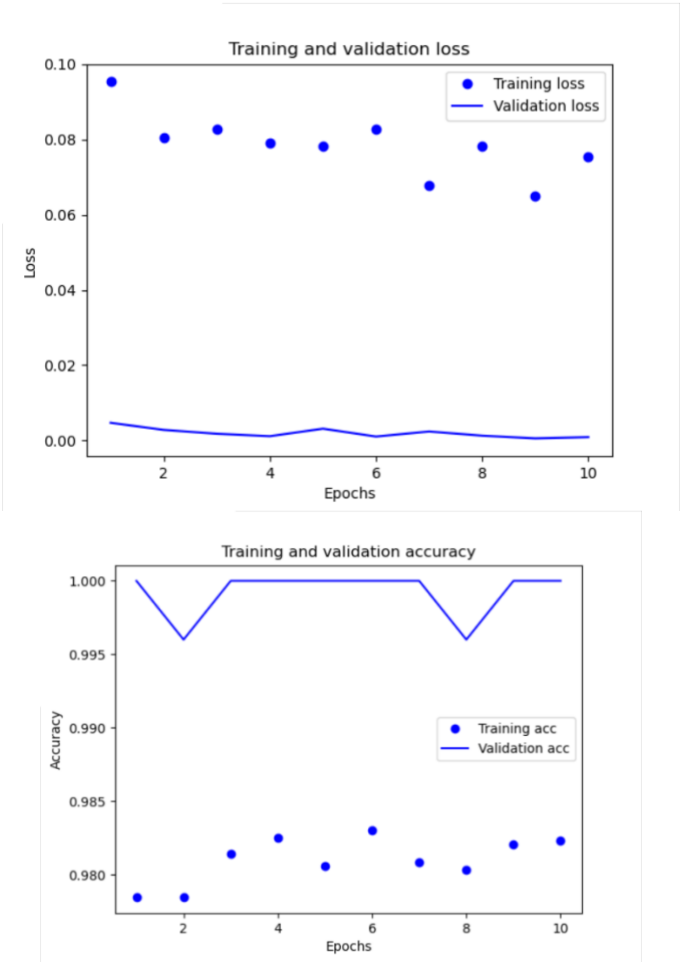


图 2.11 损失曲线和准确率曲线可视化图

为了提高网络在测试集上的表现，我们考虑引入伪标签（pseudo label）这种无监督的方法。伪标签的方法将无监督学习引入到深度学习这类监督学习中，将整个项目变成半监督学习算法。伪标记可以解决预测过程中产生的数据没有标注的问题，生成伪标签后的数据可以继续送入模型中进行训练。其具体流程可以用图 2.12 表示：

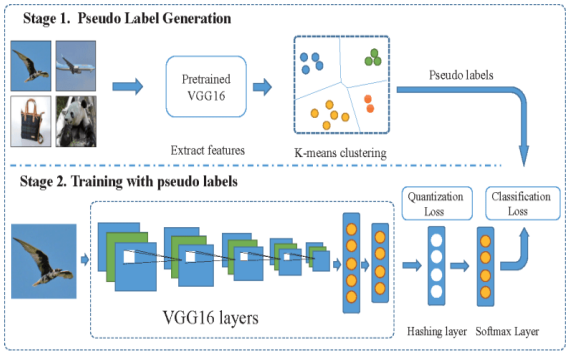


图 2.12 伪标签训练流程

为了生成伪标签我们首先使用预训练的SLNet对百分之五十的测试集进行特征提取，然后通过无监督聚类得到伪标签，将得到的伪标签再次送入 SLNet 进行训练。这样得到的结果相比原始的算法的性能提升了 0.5 个百分点，提升后训练集上的识别率达到 95.5%。具体的结果如图 2.13 所示。

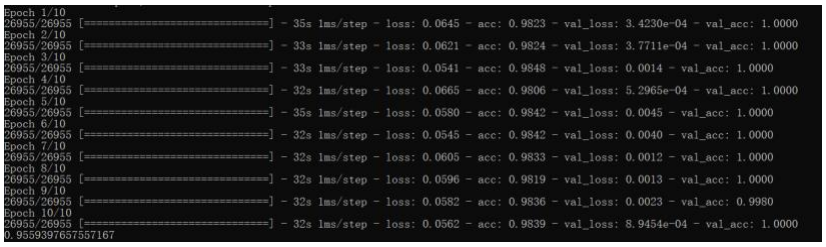


图 2.13 半监督算法性能提升结果

第四步为在线测试。在线测试的过程加入了手部轮廓检测，这样能够更准确的显示用户的手部轮廓，手部轮廓检测的方法为首先通过对颜色空间进行调整后提取连通域即可以获取手部轮廓数据，具体的结果如下所示，对于 K 这种较难识别的字符算法也能够

非常稳定的识别，而且能够准确的识别手部轮廓区域。如图 2.14 所示。在线识别的过程中我们同样采用伪标签对数据进行标注，再通过这些半监督学习产生的数据对模型进行训练，逐步提高在线系统的识别率。

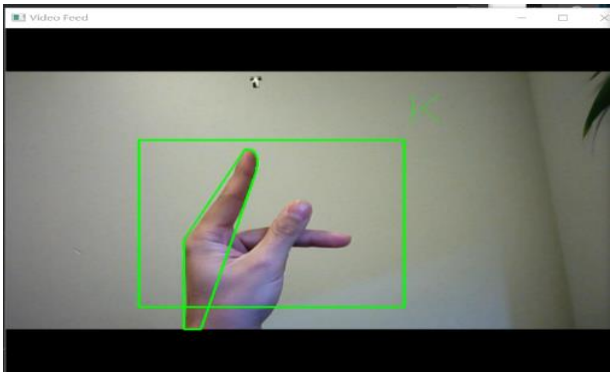


图 2.14 手语字符识别结果

2.3 聊天机器人相关技术介绍

2.3.1 聊天机器人技术介绍

图 2.15 展示了一种常见的聊天机器人系统框架结构图。

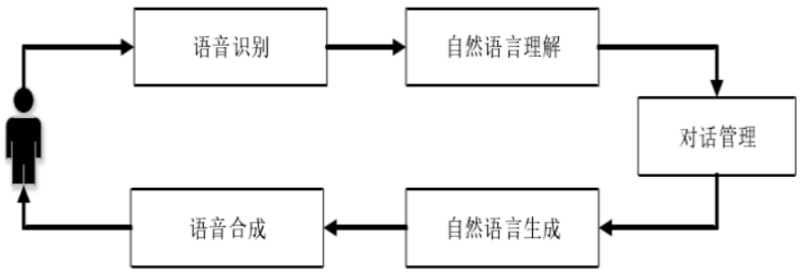


图 2.15 聊天机器人的系统框架

语音识别模块的作用是将语音转换成为文字，语音合成是语音识别模块的逆过程，即将文字转换成为语音。

2.3.2 Seq2Seq 模型

2.3.2.1 RNN 循环神经网络

循环神经网络和传统的全连接神经网络相比，增加了各个时间单元之间的隐藏状态连接，通过各个时间状态之间的隐藏状态连接，RNN 能够获取随着时间变化的信息，从而“记住“了之前的内容信息，并将这种信息传递到下一个时间单元。因此 RNN 在 NLP，时序序列建模，文本预测等领域有着非常广泛地应用。

RNN（Recurrent Neural Network）的结构如图 2.16 所示，隐藏层内部结构如图 2.17 所示[3]。

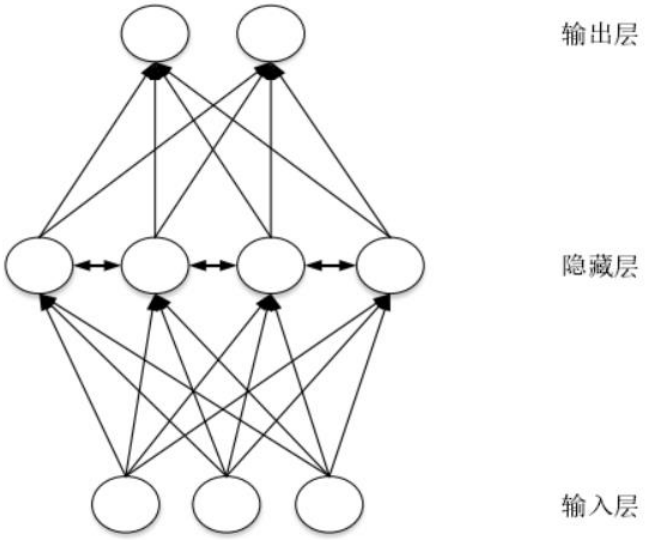


图 2.16 RNN 循环网络结构

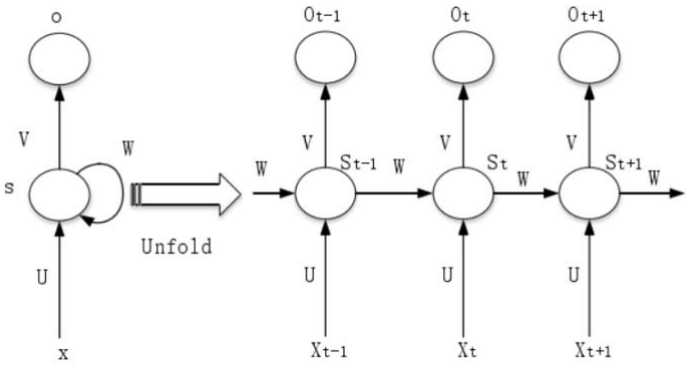


图 2.17 RNN 隐藏层结构

RNN 算法的优势在于处理时序序列问题，在处理 RNN 时序序列的过程中存在着梯度消失的问题，主要是指 RNN 在处理长时间序列时出现的记忆值较小的情况，针对 RNN 算法梯度消失的问题，LSTM(Long Short-Term Memory)算法应运而生[4]。图 2.18 展示了 LSTM 网络的结构图。

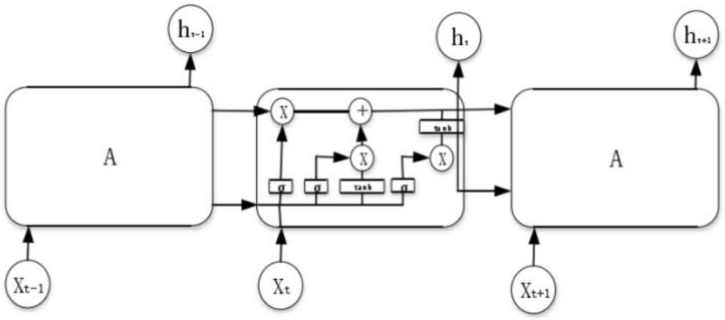


图 2.18 长短时间网络

LSTM 最重要的是单元状态(cell state),也就是图 2.19 中 LSTM 单元上方从 Ct-1 直到 Ct 的直线，它的主要作用就是传递信息，把上一个神经单元的信息传递到下一个神经单元。

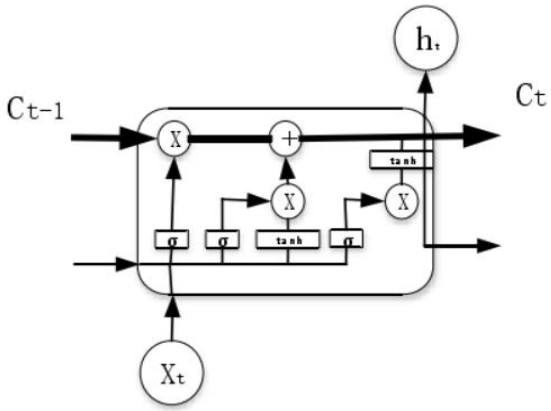


图 2.19 单元状态

LSTM 针对传统的 RNN 改进的一个重要概念便是门，LSTM 的包括三个门结构，即输

出门，输入们和遗忘门。门的作用是对信息进行处理，实现对信息的舍去或者说保留的判定工作，对于重要的信息我们可以实现保留，将这个信息继续传递下去，对于不重要的信息我们可以舍弃。下面将对这三种门结构进行介绍。

1）遗忘门

遗忘门用来控制信息的遗忘程度，上一个神经元传递过来的信息由遗忘门进行取舍保留。h_{t-1}是上一单元的输出信息，Xt 是当前单元的输入信息，激活函数会把 Ct-1 将当前的输出结果映射到一个 0~1 的范围内。结构图如图 2.20 所示：

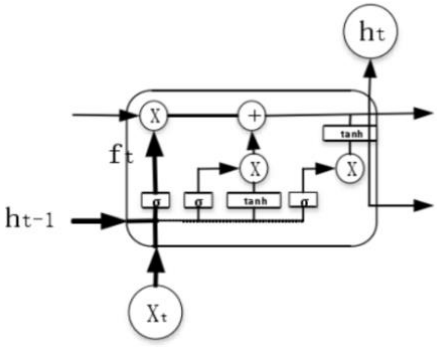


图 2.20 遗忘门

2）输入门

输入门的作用是实现当前输入信息的传递工作，图 3.6 展示了输入门的结构。如图 2.21 所示，其中经过 tanh 函数会有一个新的向量 Ct，同样的输入门会把 Ct 向量中的每一个元素的值映射到[0,1]之间，经过这样的操作，当前神经单元的输入是可控的，也就是说可以决定加入哪些信息。上一小节已经讲过，遗忘门是用来控制前一神经单元传递到当前神经单元信息的遗忘程度，如图 2.21 所示，遗忘门的输出是 f t，从本小节可以看出，输入门的作用是控制当前神经单元新信息的加入情况，输入门的输出记为 it，如下公式就是输入门的计算过程。

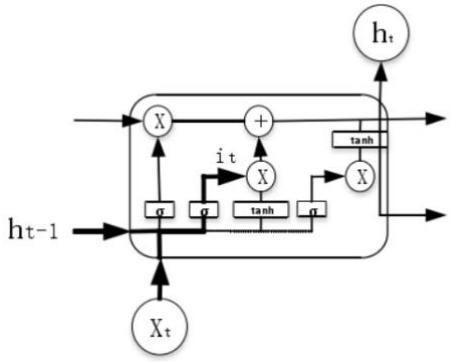


图 2.21 输入门

3）输出门

输出门的作用是控制当前神经单元的信息输出情况，输出门结构如图 2.22 所示。从图中可以看出，输出门会把输出矩阵中的每一个元素映射到[0,1]之间，从而判断哪些信息被输出，哪些信息会被过滤。

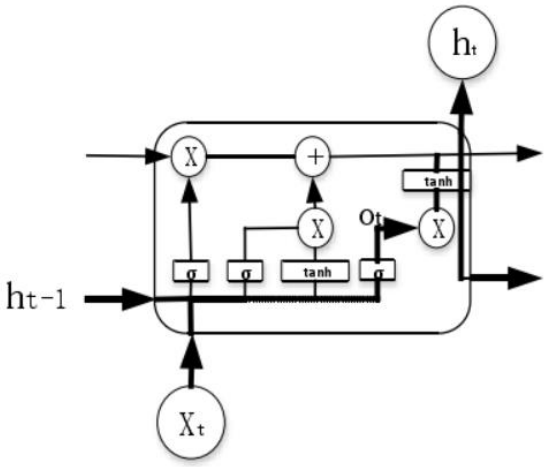
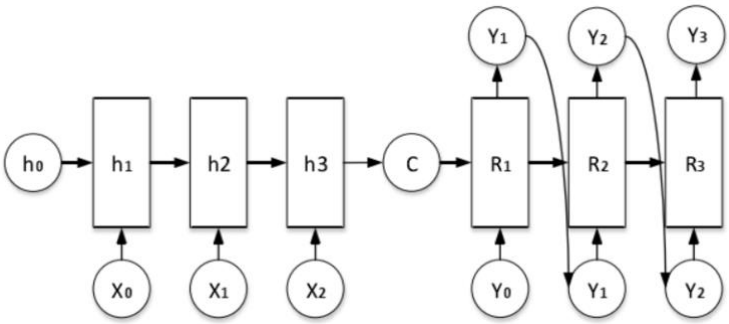


图 2.22 输出门

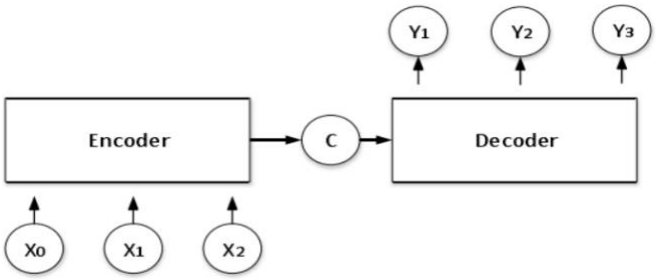
2.3.2.2 Seq2Seq 网络结

自然语言处理是机器学习的一个重要领域，在这个领域的主要任务包括，聊天机器人，文本摘要，文本生成和机器翻译等等。在机器学习方法出现之前，通过对文本的相似度进行匹配，然后搜索数据库的数据进行问答，这种方法的缺陷是严重依赖于数据的完整性，如果数据的完善度不高那么很可能出现模棱两可的回答。这极大的制约了自然语言处理技术的发展。

Seq2seq 模型在 2014 年由谷歌大脑团队和 Yoshua Bengio 提出病发表，Seq2seq 的主要目标是解决机器翻译中的数据依赖问题，seq2seq 模型的输入为需要翻译的语言，而输出为翻译后的结果，简而言之即 seq2seq 将一种语言翻译成另外一种语言，seq2seq 的内部结构可以是 RNN，LSTM，GRU 等常见的时间序列神经网络，通过对网络的训练可以实现从一种语言到另外一种语言的映射。图 2.23 展示了 seq2seq 的模型结构。



（1）



（2）

图 2.23 seq2seq 模型结构

图 2.23 中的（1）和（2）分别展示了 seq2seq 的内部细节图和抽象结构图。从图（1）中我们可以看出，seq2seq 的 encoder 和 decoder 模块的设计可以十分灵活，可以选择多种网络组合的结构，例如，encoder 模块可以采用多种神经网络结构组合，而且对于网路的输出既可以选择全部的输出结果，也可以选择部分的输出结果，因此 seq2seq 在结构上具有巨大的灵活性。当我们输入一段长文本，而输出为一个词或者一小段文本

时，seq2seq 结构可以演化成为一个文本摘要和信息抽取结构。当我们输入一个词或者一小段文本而输出为一大段文字时，则 seq2seq 结构演化成为一个文本生成或者说文章生成模块。

2.3.2.3 多抽头组合 attention 机制

在计算机视觉中，我们常常会将视觉的注意力集种于一张图片或者眼前景象的某一个部分，这种机制被称为是注意力机制。人们注意到 Attention 机制同样存在于文本和信息处理领域[6]。上述提到的 seq2seq 结构中便可以引入 Attention 机制，Attention 机制会为输入的文本数据进行重要性分配，能够保证模型训练的过程中能够将更多的资源应用于重要的词或者结构中，从而提高 seq2seq 结构的训练效果。

假设我们输入一段文本，那么其中必定包含重要的文本信息和不重要的文本信息。和人观测图像信息一样，我们可以将注意力或者说关注重点集中于文本中的某一段信息中来，对于长时间序列的文本，仅仅靠 seq2seq 结构很难将模型的重点集种于文本中的某一段或者某个关键信息，因此 attention 结构的出现极大的解决了 seq2seq 结构无法处理长文本的序列信息的问题。

Attention 机制解决了传统的 seq2seq 结构无法对整个序列的重要性进行建模的缺陷，但是单一的单抽头 attention 机制仍然存在着信息丰富度不高的缺陷。因此该模型的输出存在着问答过于直白和语音丰富度过低等一系列问题，因此本文提出了针对情感安抚机器人的多抽头组合的 attention 机制，首先将输入的文本 token 转换为对应的 query,key 和 value 向量,然后通过矩阵乘法操作后通过 Additive Attention 再与 value 向量进行矩阵乘法，同时针对每一层进行 self-attention 计算，之前的单一单抽头 attention 机制只是对每一次层得到的矩阵进行取平均操作，而在此处我们将对每一层的结果进行特征提取后通过上述的组合 attention 进行特征提取后在进行 concat 操作，最后得到全局的 attention 结果，这极大的提升了整个系统的信息丰富度。

2.3.3 基于 seq2seq+attention 的聊天机器人

本项目中采用的数据集来源是青云语料数据集和夸夸数据集，夸夸数据集是我们自己收集的数据。夸夸数据集的目的是对用户进行情感安抚，当用户的情感出现问题时，我们可以通过夸夸模式对用户的情绪进行安抚，从而达到情绪治疗的目的。

{"title": "因为没有男朋友，求夸", "url": "https://www.douban.com/group/topic/135844056/", "author": "71277500", "last_reply_time": "03-17 16:40", "content": "笨人原本一个人好好的，都单了两三年了，一直觉得挺开心的。最近不知道抽了什么风，突然特别想找个男朋友。但是但是，偏偏找不到靠谱的男朋友！现在一个人睡不着，没想明白这事，求夸。\n", "replies_num": "14", "replies": [{"content": "你这么可爱肯定会有一个很好很好的人在等你！ ", "post_id": "135844056", "comment_id": "1834208628", "user_id": "189783421", "pub_time": "2019-03-16 01:08:38"}, {"content": "最好的肯定要晚点出现哦", "post_id": "135844056", "comment_id": "1834208775", "user_id": "189783421", "pub_time": "2019-03-16 01:08:52"}, {"content": "“笨人”，刚看到开头就笑了", "post_id": "135844056", "comment_id": "1834282396", "user_id": "192799520", "pub_time": "2019-03-16 07:50:50"}, {"content": "一个好可耐的宝宝", "post_id": "135844056", "comment_id": "1834282931", "user_id": "192799520", "pub_time": "2019-03-16 07:52:24"}, {"content": "也许明天就出现了", "post_id": "135844056", "comment_id": "1834290527", "user_id": "185989534", "pub_time": "2019-03-16 08:11:38"}, {"content": "你

图 2.24 夸夸数据集

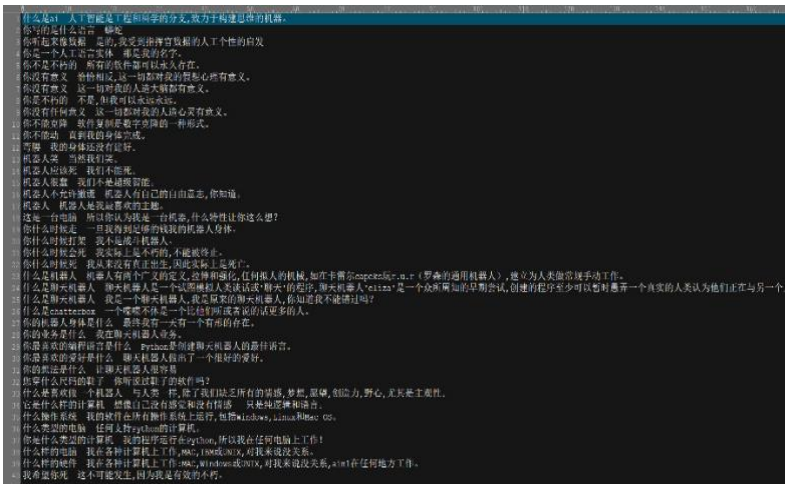
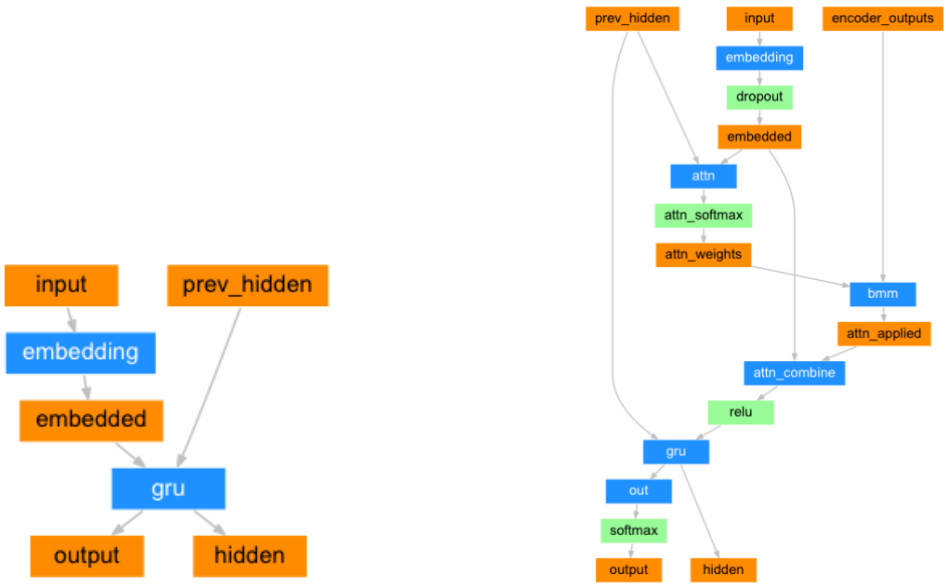


图 2.25 青云数据集



a)编码　　b）解码结构

图 2.26 基于 Attention 的编码解码网络结构

该项目采用了两种可选模式，一种是基于数据库的模式，一种是基于生成的模式。基于数据库模式由我们事先构造好的数据库提供答案，生成模式的结果来自于 seq2seq 生成的结果。

下面介绍聊天机器人的具体实现流程。

步骤一：首先对聊天机器人的数据进行预处理。在这个步骤中我们将采用 python 的 jieba 分词包，jieba 分词工具的目的是对句子进行切分处理。然后对所有的文字字符进行字典映射，将文字映射成为对应的字符作为神经网络输入。整理出数据集后得到 corpus.pth 文件，后续将该数据集分为三个部分：训练集，验证集和测试集。

步骤二：对数据集进行训练。先介绍网络训练的细节，编码器输入的最大语句长度为 50，设定最大的生成语句长度为 30，训练过程中的 batch_size 大小为 2048，在训练的过程中将会随机打散数据集，为了提高读取数据的速度，在此项目中使用了多线程对数据集进行了读取，RNN 的隐藏节点数目为 256，在此项目中采用的 Attention 模式为

dot 模式，同时为了防止 RNN 出现梯度爆炸和梯度消失，采用了梯度裁剪，初始学习率为 0.001，每 1000 个 epoch 之后下降百分之一，同时采用了 teacher_forcing 方法对 seq2seq 网络进行训练，teacher_forcing 方法能够解决原始 free_running 方法存在的输出预测能力较弱的问题。

在训练了 7000 个 epoch 之后，我们会得到训练的模型，我们训练的环境为谷歌的 colab，非常感谢谷歌为我们提供免费的 GPU 环境，如图 2.28 所示，谷歌为我们提供的 GPU 型号为 Tesla T4 英伟达显卡，该显卡的显存大小为 16G，对该项目而言，当 batch_size 为 2048 时能够满足基本要求。图 2.29 展示了训练过程中损失的曲线图，可以看到随着训练 epoch 的推进，训练损失从最初始的 4.2 下降到了 0.6 左右，训练过程中每 100 个 epoch 保留一次模型，模型后续会成为我们搭建在线聊天环境的基础。

步骤三：基于模型搭建了一个在线的聊天环境。和训练不同的是我们需要将神经网络的输出转码成文字，这个需要借助在步骤一中建立的字典。为了提高聊天机器人的回答性能，在这里采用了 beam search 算法。聊天机器人的输出本质上是在可能的输出路径上搜索一个最大可能的路径。在路径搜索方法中，广度搜索优先策略是一种常用的算法，但是随着搜索空间的增加，广度搜索有限策略内存占用指数级增加，会造成内存溢出，这极大地限制了算法的应用场景。Beam serch 算法为了解决广度搜索算法提出地策略，Beam Search 算法在广度优先搜索的基础上加入了类似剪枝的策略，保证每次搜索的最大数目为 N，这极大的减少了广度搜索有限算法的内存消耗。

在本项目中采用的 beam 宽度为 2，此时会选出每次 seq2seq 中解码器输出的所有可能值中最大的两个，最后在搜索所有可能值中概率最大的支路，这条解码即为最终的输出。

下面给出了 seq2seq+单一 attention 和 seq2seq+多抽头组合 attention 两个模型的性能对比，可以看到后者的 BLEU 达到 0.91，而且在在线预测的环节中也有较好的表现。

```
!grep 'processor' /proc/cpuinfo | sort -u | wc -l

2
```

图 2.27 云服务器 CPU 线程数信息

```
Mon Nov  2 01:22:32 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 455.32.00    Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   41C    P8     9W /  70W |      0MiB / 15079MiB |      0%      Default |
|                               |                      |                 ERR! |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

图 2.28 云服务器 GPU 信息

```
1m 21s (- 19m 4s) (5000 6%) 2.8434
2m 41s (- 17m 27s) (10000 13%) 2.3140
4m 1s (- 16m 4s) (15000 20%) 1.9992
5m 21s (- 14m 43s) (20000 26%) 1.7859
6m 40s (- 13m 21s) (25000 33%) 1.5720
8m 0s (- 12m 1s) (30000 40%) 1.4146
9m 21s (- 10m 41s) (35000 46%) 1.2599
10m 41s (- 9m 21s) (40000 53%) 1.1651
12m 1s (- 8m 1s) (45000 60%) 1.0417
13m 22s (- 6m 41s) (50000 66%) 0.9323
14m 43s (- 5m 21s) (55000 73%) 0.8799
16m 3s (- 4m 0s) (60000 80%) 0.7951
17m 23s (- 2m 40s) (65000 86%) 0.7308
18m 43s (- 1m 20s) (70000 93%) 0.6583
20m 4s (- 0m 0s) (75000 100%) 0.6045
```

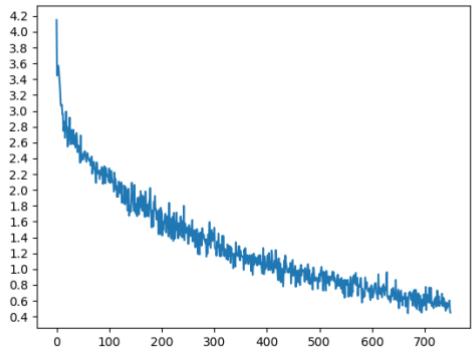


图 2.29 训练损失图

网络结构	BLEU
Seq2seq+attention	0.83

Seq2seq+多抽头组合 attention	0.91

图 2.30 两种 attention 模式的结果对比

2.4 基于手语识别的情感交互系统

2.4.1 系统流程

首先我们通过调用电脑或者手机上的摄像头，锁定用户的手部区域，然后开始进行手语识别。手语识别的结果会将人手势所传达的信息转换成为文本信息，文本信息后续被送入到聊天机器人中，聊天机器人会根据 seq2seq 生成的结果返回给用户问答效果。图 2.31 展示了整个系统的流程。

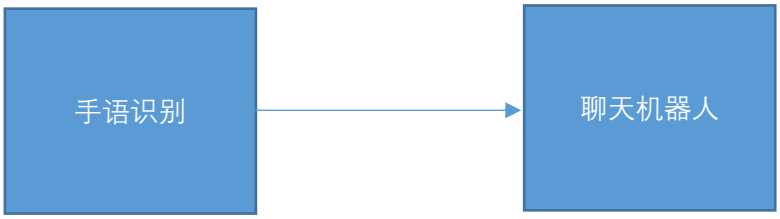


图 2.31 系统流程

2.4.2 具体模块实现

我们会首先确定手所在的位置，然后开始对手势的内容进行识别，识别结果如下图 2.32 所示：

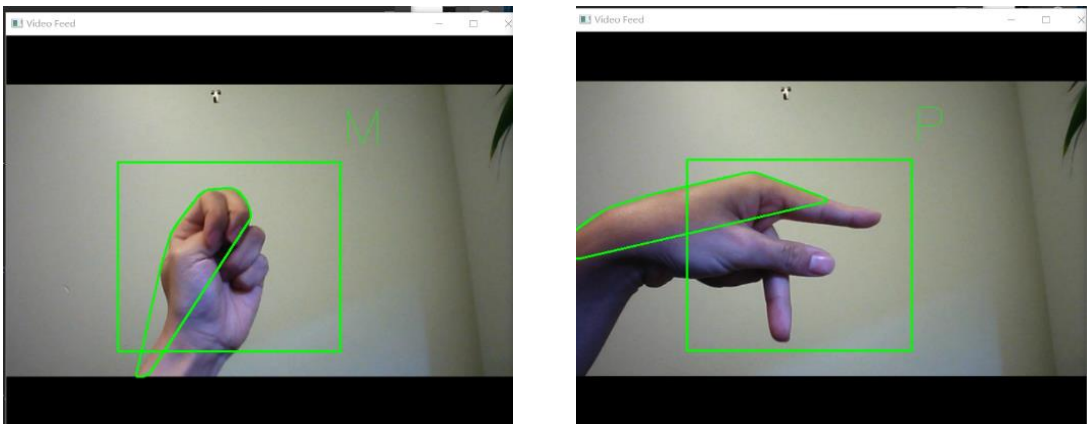


图 2.32 手语识别页面结果

我们将识别结构组成句子编码成输入语句送到我们的聊天机器人，如图 2.33 所示。可以看到当我们说出夸我时，机器人会回复一些鼓励性的语句，当用户的心情低落时，聊天机器人的结果对于用户有一定的正反馈作用。

图 2.33 聊天机器人展示图

3. 讨论

基于手语识别的情感交互系统主要分为两个部分：前端的基于 SLNet 的半监督手语识别系统和后端的语言交互系统。该项目第一次将多模态（视觉和自然语言处理）引入到语言障碍认知的情感交流中来。

前端的手语识别系统的目标是准确的识别来自用户的手语信息。为了提升识别率我们设计了一套完整的数据采集程序，用户只需要根据程序的指示即可以完成整个手语识别数据集的录制。而且用户还可以根据自己的需求设计手语，同时完成新的手语识别系统。

后续的训练采用了针对于手势识别的 SLNet 手语识别网络，SLNet 的识别率为 95%。同时为了提升SLNet的识别率采用了半监督学习,引入了伪标签对网络的性能进行提升，最终的识别率达到 95.5%。在线识别的关键点在于如何准确的找到手部轮廓然后进行手部轮廓识别，因此首先对图像进行了空间色度变换找出连通域对手部轮廓进行确认，在线预测的过程中 K，P，M，A 这类较难区分的手势均能够准确识别。这表明采用自己录制的数据集训练的模型的鲁棒性（robustness)。

后端的聊天机器人首先需要对手语识别的结果进行编码转换成常用的字母和词组，此处的语言模型是我们根据具体常见的使用语言设计的。采用了当前各类翻译和自然语言对话中采用的 seq2seq+多抽头组合 Attention 机制。同时为了达到对语言障碍人士进行安抚的目的，我们收集了来自豆瓣上的夸夸群数据，并利用这些数据对 seq2seq+多抽头组合 Attention 模型进行了训练。结果表明聊天机器人能够对用户进行情感安抚，同时能够应对各种闲聊场景。

但是这个项目同时也受到一些外界环境的影响，比如光照，遮挡，其他人经过同样会对结果产生干扰，后续改进的第一个点是通过数据采集系统采集更多场景下的手语数据；第二是在本项目中在线手语识别的手部轮廓检测采用的传统的方法，传统的连通域检测方法会受到外界环境的干扰，比如肤色，光照等，后续改进会采用深度学习的方法进行改进；第三是聊天机器人的转码模块需要进行一定的纠错功能，对输入要进行诸如语病修改等功能；第四是丰富聊天机器人的功能，目前主流的聊天机器人可以实现诸如作诗，写作文等功能，目前情感交互系统中的聊天机器人暂时没有实现这些功能，这些功能能够极大的丰富机器人的应用场景。

4. 结论

本项目完成了一个多模态情感交互系统，结合了计算机视觉和自然语言处理技术，如图 4.1 所示。

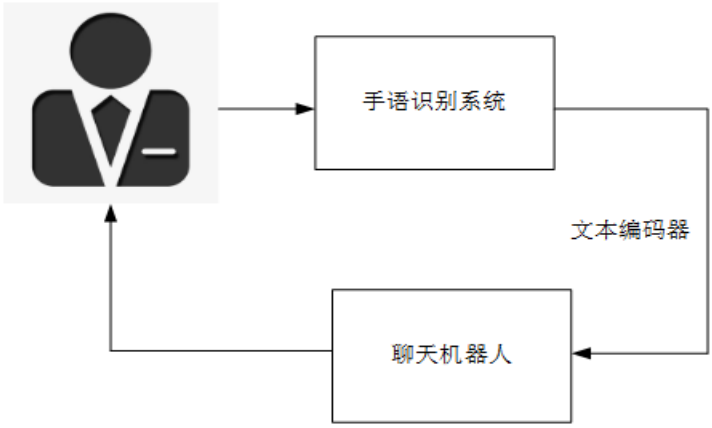


图 4.1 基于手语识别的情感交互系统

交互系统的前端为基于视觉的半监督手语识别系统，此系统完成了从数据采集、模型训练、在线预测到手部轮廓检测等一整套工作，同时采用伪标签对数据集进行了扩充，最终识别率达到 95.5%，特别对较为复杂的 R 等手势识别较为准确，如图 4.2 所示。

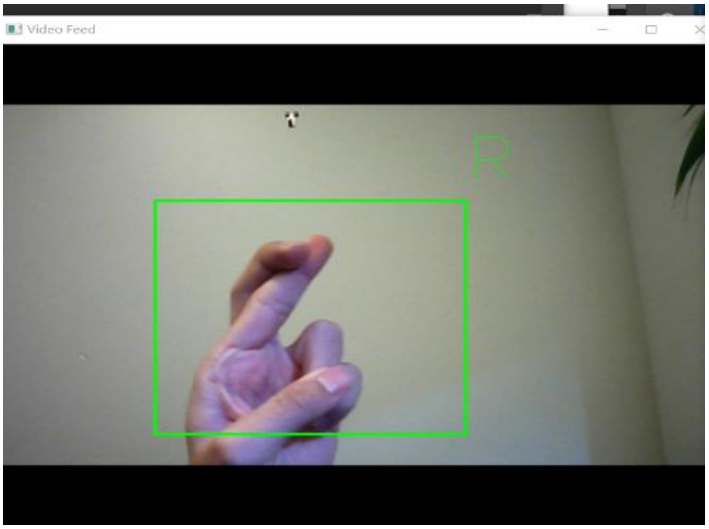


图 4.2 手语识别结果

交互系统的后端为基于 seq2seq 的聊天机器人系统，为了达到对用户进行情感安抚的目的。本项目收集来自豆瓣的夸夸数据集对模型进行训练，结合青云数据集，聊天机器人能够对情感障碍用户进行情绪安抚。

5. 致谢

本论文在写作过程中得到了清华大学杨毅副教授、北京师范大学附属实验中学韩冬兵老师的支持和帮助，感谢他们在项目完成过程中提供的帮助。同时感谢北京师范大学附属实验中学马静老师给予的鼓励与支持。

6. 参考文献

[1] Dey, Anind K., Gregory D. Abowd, and Daniel Salber. 2001. “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications.” Human-Computer Interaction 16 (2): 97–166.

[2] Lecun Y, Bottou L. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[3] Williams R, Zipser D . A Learning Algorithm for Continually Running Fully Recurrent Neural Networks[J]. Neural Computation, 2014, 1(2):270-280.

[4] Sundermeyer M , Ralf Schlüter, Ney H . LSTM Neural Networks for Language Modeling[C]// Interspeech. 2012.

[5] Venugopalan S , Rohrbach M , Donahue J , et al. Sequence to Sequence -- Video to Text[J]. 2015.

[6] Chen K , Yao L , Wang X , et al. Interpretable Parallel Recurrent Neural Networks with Convolutional Attentions for Multi-Modality Activity Modeling[C]// 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.