CNN手势识别

在本次任务中，我实现了一个简单的卷积神经网络手势识别工程，首先，谈一下我的卷积神经网络。

与普通的神经网络不同的是，卷积神经网络的数据在进入神经元之前，会先进入卷积层，完成卷积和池化操作。卷积的主要目的是提取特征，用一个核去和输入矩阵上和该核大小相等的区域进行卷积操作，从矩阵左上角开始，依次遍历完整个矩阵，步长可设。池化则是用一个数去代替一个区域内的数据，比如最大值或平均值，以实现数据聚合，在保留矩阵特征的同时，降低随后的神经网络层运算复杂度。在我搭建的卷积神经网络中，有两个卷积层，两个卷积层的卷积核大小都是3，第一层将输入的单通道通过卷积操作变为32通道，随后用2x2的核，步长为2进行最大值池化，第二层通过卷积操作将32通道变为64通道，随后依然用2x2的核，步长为2进行最大值池化，卷积层结束后，有一个0.5的dropout层，以防止过拟合。随后便进入神经网络层，隐藏层的神经元个数为256，六分类器，输出层神经元个数自然是6，隐藏层之后的激活函数为relu，有0.5的dropout以防止过拟合。优化器采用Adam优化器，损失函数是交叉熵，即概率分布误差，实际上也正是通过计算概率的方式来确定图片的类型。起初的学习率设为0.01，但是出现了神经元死亡的情况，后来发现这是因为relu激活函数本身的梯度下降比较快，而算上卷积层的两个relu，已经隐藏层的relu，一共三个relu，这样的话，这个CNN在学习时梯度下降的速率是非常大的，因而在学习的过程中，会出现某个梯度，让网络中的神经元进入一个不再可调的状态，且随后的其他数据进入网络也无法再次将其激活，即神经元死亡，这样的后果非常恶劣，根本无法学习，在训练失败了很多次之后，我终于发现了原因，只是简单地把学习率改为了0.0001就解决了问题。这样一个简单地卷积神经网路框架就完成了，虽然这个模型比较简单，但是有效，将其用来训练MNIST时，准确率可以达到近百分之百，将其用来训练我自己制作地tfRecord手势数据时，准确率最高也能达到98%左右。

卷积神经网络构建好之后，就需要把数据喂入神经网络了，为了能够识别我比较心仪地手势，我选择自己制作数据集，数据集形式为tfRecord，将采集的图片制作成tfRecord后，读取数据非常方便，形式简单。六种手势，每种手势准备了两千张图片，1600张用于训练，400张用于测试，一共12000张图片，由于这是卷积神经网络，特征提取能力强，因此这个数据量已经足够。在训练超过数千次时，loss就可以降到的数量级，训练次数过万时，loss的数量级已经达到，测试时准确率可以达到接近100%，时间问题，就没有继续训练，直接进行识别，实时捕捉的手势都可以准确识别。

整个工程下来，模型其实很简单，由于是第一次搭建卷积神经网络，在调参方面经验欠缺，因此在调参方面花的时间是最多的。参数对于整个网络来说着实重要，对于CNN来说，卷积层要完成特征提取的任务，必须有合适的参数，才能尽可能的把特征放大，神经网络层也应该有合适的参数来让训练效果最优。

对于比赛中要实现的手势识别，基本的要求应该是，空中机器人能及时对手势做出响应，难度有两个，其一，在比赛中要识别的是动态手势，即识别手势的一个变化过程，进而判断命令并执行，其二，由于角度，光线，动作等因素存在，空中机器人采集到的图片的特征很有可能并不明显，因此卷积层至关重要。

此外，并不是所有的时间点都要通过手势给机器人下命令，因此手势识别系统应该不是一直处于工作状态，合理的情况应该是，机器人实时检测手势动作的出现，如果检测到了手势，进行识别，否则处于正常飞行状态。这个手势识别系统要识别的应该是手势特征点的位置变化，我的想法是，用一定数量的特征点来描述手，这些特征点在一段时间内的不同时间点内有不同的位置，位置需要用x和y来表示，构造一个二维矩阵，两个维度分别为时间和特征点编号，每一个元素是某个特征点在某个时间点的point，这样一个矩阵就能反映手势在一段时间内的动作变化，用这样的矩阵来训练神经网络，就能够让机器识别出手势的动作变化。