

姿势识别项目说明文档

18373263 贺梓淇

项目说明

本项目的目标在于：利用手机传感器收集的信息，识别做出姿势的人正在比划的数字或字母。

所有的训练数据均来自：https://github.com/bczhangbczhang/Inertial-Gesture-Recognition/tree/master/raw_data

在总共 1120 个，来自 8 个不同类别的数据的训练下，利用 DNN 的方法，最终得到了近乎完美的识别准确率：

目标类别	原算法的最大值（%）	本算法最佳结果（%）
A	97.14	100.0
B	95.71	100.0
C	89.29	100.0
D	93.57	100.0
1	97.86	100.0
2	93.57	100.0
3	97.86	100.0
4	94.29	100.0
均值	94.91	100.0

详细的统计数据见 `models/xxx/result_xxx.json`

实验环境

Windows 10 2004
Python 3.7.4
PyTorch 1.6.0
NumPy 1.19.1

数据整理

原始的数据具有以下几种特征：

1. 数据格式不能直接供计算机使用
2. 序列的长短有差异
3. 同一个序列内的采样间距有差异
4. 数据总量不足（八类共 1120 个数据）

这给训练神经网络或者其它模型造成了巨大的困难。因此，在把数据传入模型进行训练前，对数据进行整理变得很有必要。

在本实现中，我主要对数据进行了三级处理：

1. 在 `data\Serializer.py` 中，我将不具有可读结构的数据集序列化为可读的 `json` 数据，这也是以后保存数据的中间文件格式
2. 在 `data\Uniform.py` 中，我对数据进行了归一化和随机化处理，具体流程包括：
 - 将时间码的起始位置统一为 0
 - 对数据进行随机化处理，生成更多样本：
 - 对加速度和陀螺仪的数据插入幅值为 `1e-3` 的噪声
 - 对整个序列进行修剪，随机裁去序列头部或尾部的 $\leq 5\%$ 全序列数据长度的部分
 - 对于每个样本，在保留原样本的情况下，利用以上规则生成四个不同的样本
 - 将非均匀采样的变长序列使用插值重新采样为长度为 200 的均匀采样序列
 - 对加速度与陀螺仪的数据进行归一化处理，让这两个三维向量的模长范围保持在 `[0, 1]` 内
 - 将加速度与陀螺仪的数据合成为六维的特征向量
3. 在 `data\Organizer.py` 中，将不同类别的序列的对应预期改写为 `one-hot` 向量，并在打乱顺序后按照 80%，20% 的比例将数据分为训练集和测试集

数据处理的根据

- 加入噪声：手机传感器在采样的时候不可避免地会产生噪声，为了保证模型对于噪声的耐受度，通过添加噪声可以降低因为过拟合而产生的精确度下降的后果
- 随机裁剪：本算法应该对于序列的开始与结束位置有一定的宽容性，裁剪掉的序列长度不超过 10% 也避免了在序列过短的情况下出现不同类之间的混淆情况
- 归一化：对加速度进行归一化实际上相当于把空间中不同大小幅度的动作统一缩放到相似的大小上。而时间上的归一化则将不同的完成用时的序列统一到一个相似的时间尺度上

模型训练

用于训练的模型是一个基础的 `DNN` 模型，它的具体参数如下：

Layer	Type	Input dim	Output dim	Activation
fc1	Linear	1200	400	relu
fc2	Linear	400	200	relu
fc3	Linear	200	100	relu
fc4	Linear	100	8	softmax

其它训练参数如下：

- 损失函数：均方误差 `MSELoss`

- 优化方法：
 - **RMSprop**：学习率 **lr=4e-5**，动量参数 **momentum=0.8**
 - 学习率衰减：在第 **100, 200, 300, 400** 次迭代时将学习率减少至当前学习率的 **75%**
- 迭代次数：**600** 次

训练结果

在不同的数据集上进行了多次测试，结果如下：

编号	随机拓展倍数	训练集	测试集	测试集准确率	整体准确率
01	1（原始数据）	80% (896)	20% (224)	96.88%	99.28%
02	5	80% (4480)	20% (1120)	100.0%	100.0%
03	10	70% (7840)	30% (3360)	100.0%	100.0%

结果分析

本次项目的数据集较为简单，一共只有八个类别，并且类别之间的冲突较少，并不会出现 **0** 和 **0**，**9** 和 **q** 之间的这种较为难以分辨的差距，因此可以较为便捷地达到较高的准确度。然而，这并不意味着这类模型在较为复杂的情况下很难具有扩展性。最为直观的应用便在于序列输入的情况下利用语境以及本模型的输出联合判断姿势输入。

在没有对数据集进行拓展的情况下，由于测试数据较少，训练时间较长等原因，最终得到的模型可能出现了一定程度上的过拟合情况。表现在准确率上就是在测试集上的准确率止步于 **97%** 左右。而在引入了噪声之后，过拟合的情况得到了缓解，因此在整体的表现上得到了巨大的提升。