

用“掌控”体验机器学习

谢作如 浙江省温州中学
池梦茹 浙江省温州第十四高级中学
李旻 上海蘑菇云创客空间

涉及学科：信息技术、科学

机器学习(Machine Learning, ML)是一门多领域交叉学科,涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科,专门研究计算机怎样模拟或实现人类的学习行为,以获取新的知识或技能,重新组织已有的知识结构,使之不断改善自身的性能,是人工智能的核心。

机器学习从词面上理解就是让机器进行“学习”的技术,其最基本的做法是让计算机利用已有数据得出某种模型,并利用此模型预测结果。最近国内一线创客教师发布了一款开源硬件——掌控,掌控板支持Python语言,于是我们尝试用它做了一个简单的机器学习方面的案例——识别简单手势。

● 解决思路

市面上的手势识别解决方案有很多,如利用数据手套或光学标记。数据手套由多个传感器件组成,通过传感器将用户手的位置、手指的方向等信息传送到计算机

系统中。光学标记则需要戴在用户手上,通过红外线可获得手指的各种变化。更高级的解决方案是对视频采集设备拍摄到的包含手势的图像序列,通过计算机视觉技术进行处理,进而对手势加以识别。

我们采取的方案是利用掌控板自带的重力加速度传感器。当用户佩戴掌控完成一个固定手势时,掌控会在空中沿着一定的轨迹运动,在整个过程中加速度不断改变,将这个不断改变的加速度数据记录下来分析,可以发现同一种手势的数据变化规律是相同的。基于这个原理,我们能够将当前正在进行的手势与事先“学习”过的手势做实时对比,如果相似度较高,则判定识别成功。

判定手势是否相似的算法,我们选择了曼哈顿距离(Manhattan Distance)算法,因为实现起来比较简单,也容易理解。曼哈顿距离算法先计算一系列数据对中每对数据差值的绝对值,并将这些值累加

起来判断数据之间的差异。曼哈顿距离中距离d的定义为:

$$d = \sum_{k=1}^n |a_k - b_k|$$

当我们记录下完成一个手势的加速度数值后,可以得到时间(横轴)–加速度(纵轴)的对应关系(如图1)。通过图1可以大致判断出这是一个让掌控加速运动的手势。

录入该手势之后,我们再做一个手势与其作对比(如图2)。

根据曼哈顿距离的定义,两个

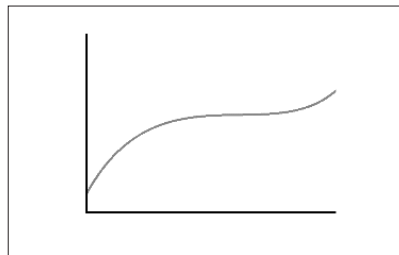


图1

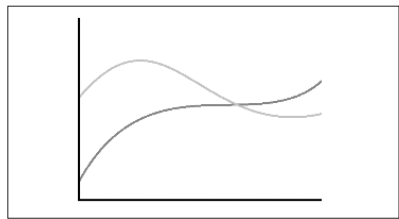


图2

手势之间的差异,可以等价于图中阴影部分的面积大小(如图3)。面积越大差异越大,面积越小则差异越小。

由此,我们可以通过此标准判断一下图4中a和b两张图的相似度。a图的面积大于b图,说明b图与录入的手势更接近。

两个手势的图形完全重合当然是不可能做到的。需要设置一个阈值,只要小于这个阈值,就表示两个手势是一致的。阈值设置太大会误判,太小则容易导致难以正确识别。

● 硬件准备

手势识别利用的是掌控板自带的重力加速度传感器,因此硬件设备只需要一块掌控板即可。为了做手

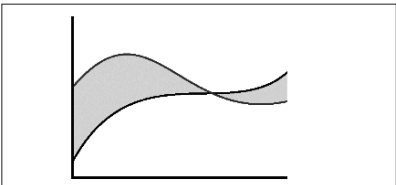


图3

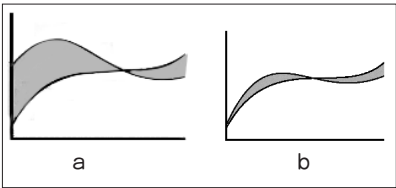


图4

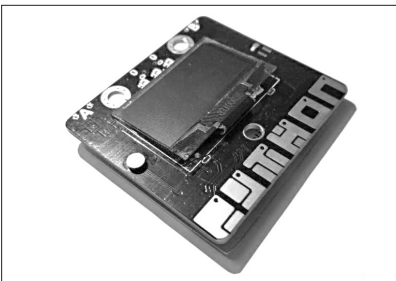


图5

势更加方便,建议使用电池供电,或者安装供电扩展板、接小型充电宝。图5为加了供电底板的掌控板。

● 解决过程

1. 流程设计

掌控板要识别手势,首先要录制需识别的手势数据,然后以此为模型判断新手势数据与录制的手势是否一致,即先学习,再识别,具体步骤如下:

①录制动作。开机后,按住A键进入到动作录制模式,录入固定长度的动作,如在空中写一个字母,将每个时间点的加速度记录在一个原始数组中。录制完成后,生成一个固定长度的原始数组。

②识别动作。完成动作后松开A键结束录制,自动进入识别模式。再次在空中比划手势,每过一固定时间间隔录入新的即时加速度,更新在样本数组中。将样本数组的每一个数值与原始数组对应的数值做差值,并取绝对值。

③结果判断。将绝对值做累加,如果绝对值小于一定阈值则判定为动作被识别,即证明动作与之前录入的动作相似,显示屏会显示“识别成功”,表示

识别成功。

2. 代码编写

用Python语言编写手势识别的代码,核心还是“学习”和“识别”两大功能,即实现按下A

键时录入动作的原始数据,松开A键默认进入识别模式。识别模式中要记录实时时间戳,并将即时的加速度数据记录下来,和原始数据做对比。下面为实现机器学习的关键代码:

①导入掌控必要的库以及数学运算的库,如图6。

②定义采样率和数组长度,如图7。

③按下A键时录入动作的原始数据,如下页图8。

④录入动作方差计算,如下页图9。

⑤默认进入识别模式并记录实时时间戳,如下页图10。

⑥将即时的加速度数据记录下来,并和原始数据做对比,如下页图11。

完整代码可通过笔者博客或者DF创客社区索取。

3. 实验测试

将掌控握于手心,OLED屏朝外,方便查看提示消息。开启电

```
from mPython import *
import math
import time
```

图6

```
NoS = 100 #Number of Samples
SR = 50 #Sampling Rate ms
g=[0]*(NoS)
c=[0]*(NoS)
i=0
j=0
Error=0
refresh = False
new_sample_time = time.ticks_add(time.ticks_ms(), SR)
```

图7

```

if button_a.value() == 0:
    display.fill(0)
    display.DispChar('开始记录', 24, 0)
    display.show()
    refresh = True
    j=0
    g = [0] * NoS
    new_record_time = time.ticks_add(time.ticks_ms(), SR)
    while button_a.value() == 0:
        # buzz.on()
        if time.ticks_diff(new_record_time, time.ticks_ms()) < 0:
            new_record_time = time.ticks_add(time.ticks_ms(), SR)
            # print('new_record_time = %d' % new_record_time)
            if j < NoS:
                x = accelerometer.get_x()
                y = accelerometer.get_y()
                z = accelerometer.get_z()
                g[j]=math.sqrt(x**2 + y**2 + z**2)
                j = j + 1
            else:
                display.fill(0)
                display.DispChar('记录已完成', 24, 0)
                display.show()
                while button_a.value() == 0:
                    time.sleep_ms(100)

```

图8

```

R = 0
for k in range (0, NoS):
    R = R + abs(g[k] - c[k])

```

图9

```

if(time.ticks_diff(new_sample_time, time.ticks_ms()) < 0):
    new_sample_time = time.ticks_add(time.ticks_ms(), SR)
    state = 0
    x = accelerometer.get_x()
    y = accelerometer.get_y()
    z = accelerometer.get_z()
    for k in range (1, NoS):
        c[k-1] = c[k]
    # new sample data add to the last position
    c[NoS - 1]=math.sqrt(x**2 + y**2 + z**2)

```

图10

```

if R < 1.0:
    c = [0] * NoS
    display.fill(0)
    display.DispChar('识别成功', 24, 0)
    display.show()
    refresh = True
    buzz.on()
    time.sleep(1)
    buzz.off()

```

图11

源后, 按住A键, 屏幕出现“开始记录”, 此时可以预先录入一个手势, 如在空中比划大写字母A。录入完毕后, 屏幕显示记录完成。任意比划时, 屏幕一直显示“正在识别……”, 当比划的手势也是A时, 屏幕将出现“识别成功”。

● 思考与总结

围棋界Master的横空出世, 让人们对机器学习充满了敬畏, 或将其想得过于神秘, 或将其想得过于万能。本案例通过Python语言让一块小小的智能硬件具有简单的机器学习能力, 掀开了机器学习的神秘面纱。虽然这是一个小数据环境下的浅层学习案例, 但原理上完全符合机器学习的技术要求, 编程实现也容易, 大多数人都可以进行研究并实现。智能时代已然来到, 学有余力且具备一定编程能力的中学生需要了解一些机器学习的相关知识, 并且, 借助智能硬件来一次真刀真枪的实操是很有必要的。

注: 本案例也可以使用micro:bit实现, 掌控板和micro:bit的Python代码规范是一致的。^[9]

如果对相关内容感兴趣, 请关注主持人博客。

