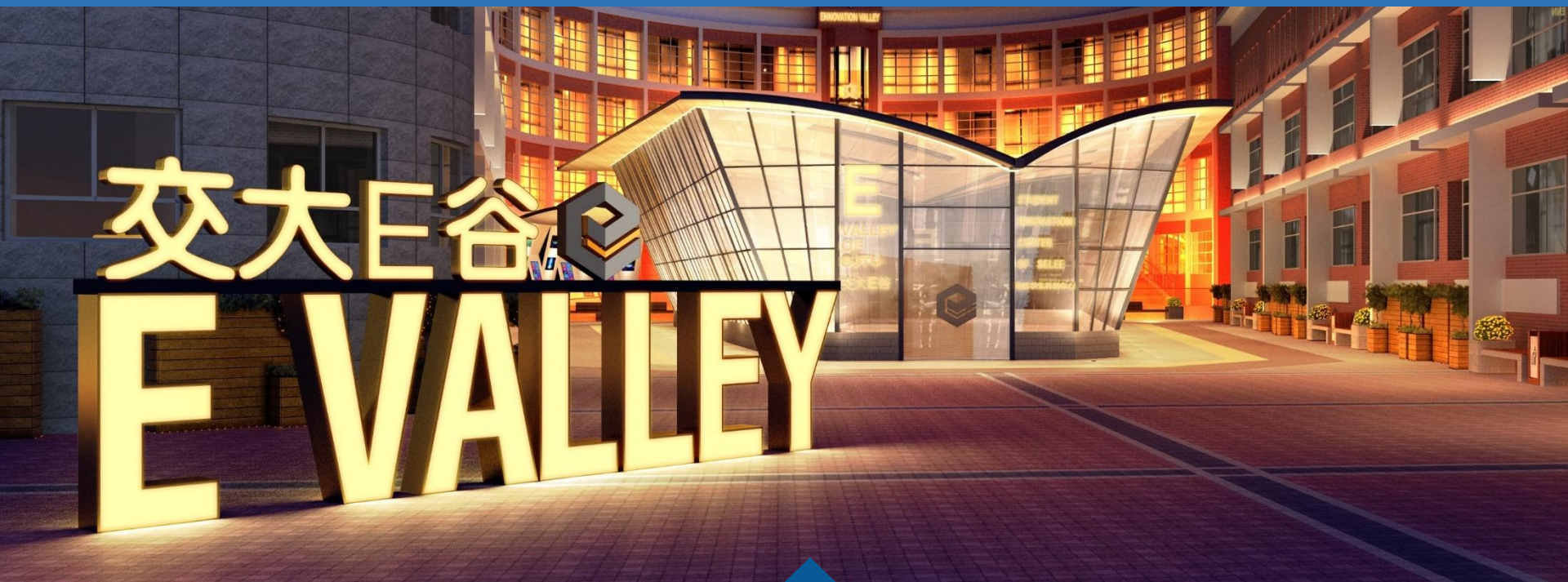




上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

工程实践与科技创新 | 实践课-脑控机器人



祁亮 微纳电子学系 助理教授

生物电路与系统实验室

内容概要

- 课程总体介绍
- 基于脑电信号实现情绪状态分类
- 机器人动作编程

第一部分

课程总体介绍

脑控机器人



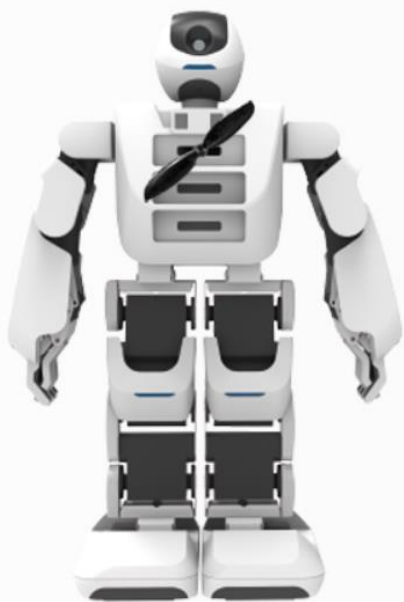
脑控机器人



- **5通道传输：** 探测全脑信息
- **无线连接：** 通过蓝牙连接PC和移动设备
- **高质量EEG信号：** 每通道128倍采样速率、14bit精度、8400 $\mu\text{V}(\text{pp})$ 动态范围
- **运动传感：** 9轴运动传感器，检测头部运动
- **软件支持：** EmotivPro实现EEG信号实时采集与传输

Emotiv• Insight

举例：脑控机器人



乐聚·Aelos

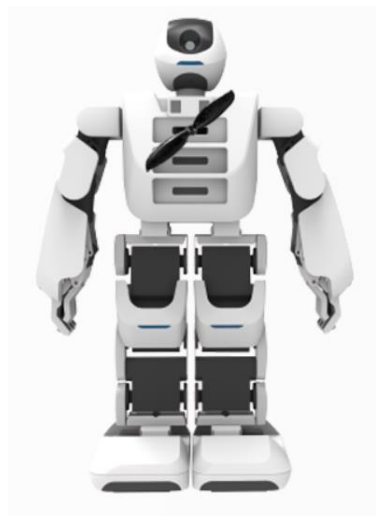
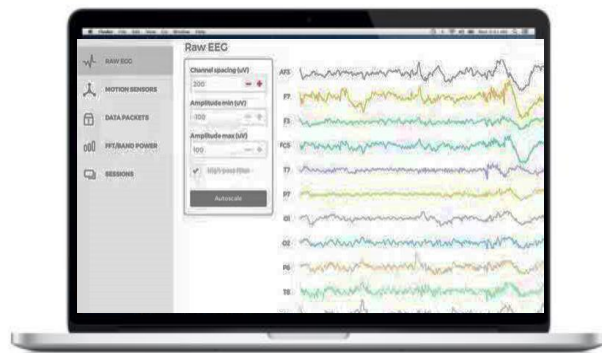
- **高度灵活：** 运动范围 180° ，运动精度 1° ，运动速度 $461^{\circ}/s$
支持舞蹈、足球、武术等千万组行为动作
- **机器视觉：** 人脸识别、颜色识别、文字识别、
物体定位、图像回传
- **自主智控：** 实现机器人的全自主控制，脱离手柄操控
- **深度开发：** 可视化积木编程：Aelos edu编程平台
代码编程：支持lua/python3等多语言环境
- **课程配套：** 基础理论铺垫，应用实例拓展

脑控机器人



数据请求 (Python)

数据发送
(PERFORMANCE)



动作控制
(Raspberry Pi)

监听请求

- 可穿戴与PC端通信
- PC端与机器人通信

第二部分

基于脑电信号实现 情绪状态分类

脑电信号——探寻人类的潜在感受



电影《阿凡达》中，利用脑机接口进行远程连接



利用脑电信号，控制机器人完成相关指令

■ 脑电信号与人大脑所感息息相关

专注

紧张

放松

疲劳

兴奋

.....

EMOTIV Insight脑电采集仪

■ 获取脑电信号——需要专业的设备



5通道EEG
全脑感知



最小设置时间
只有1-2分钟



半干式聚合物传感器
易于使用和清洁



无线连接
PC和移动设备



可充电
电池长达9小时



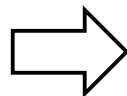
9轴运动传感器
检测头部运动



■ 通过蓝牙连接PC和移动设备

■ 128倍采样速率、14bit精度、8400 $\mu\text{V}(\text{pp})$ 动态范围，获取高质量EEG信号

■ 配套软件支持，实现EEG信号实时采集与传输



预先准备：设备所需的软件



查看实时数据流



灵活的数据存储



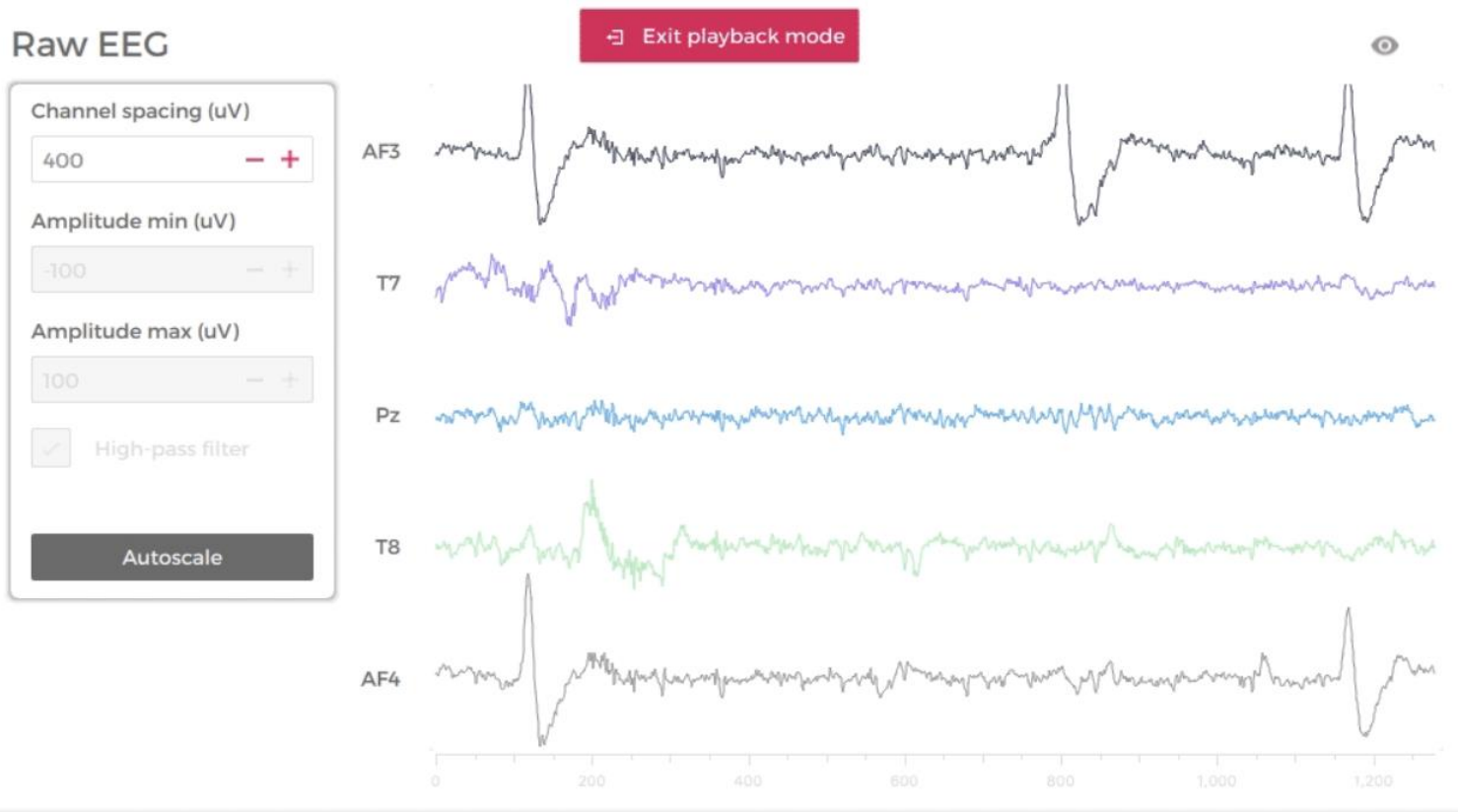
不同设备同步交流



即时FFT分析 获取功率谱图

开始信号采集

■ 五通道数据同步采集



情绪状态分类

■ 自带六种分类状态

Stress

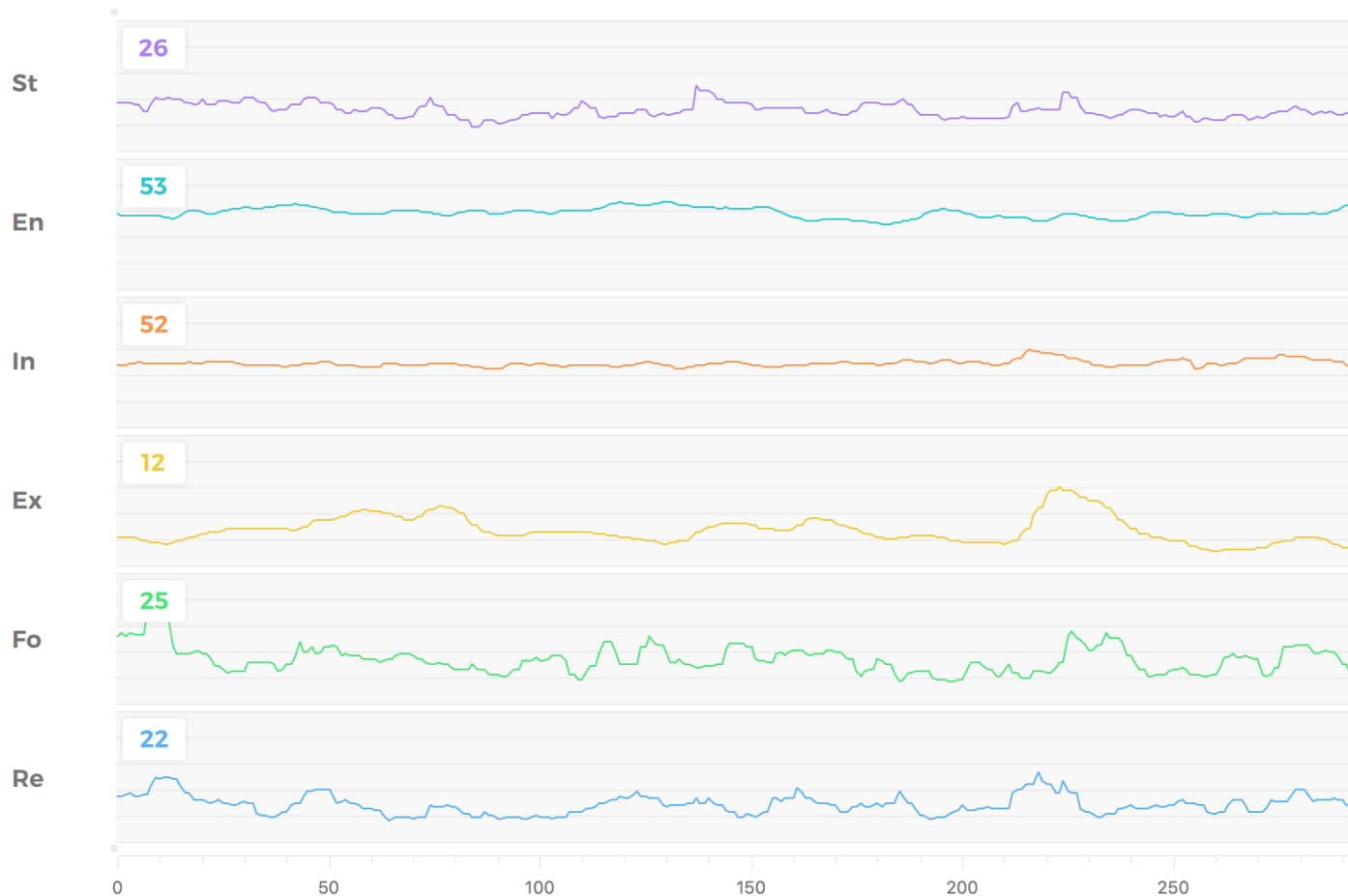
Engagement

Interest

Excitement

Focus

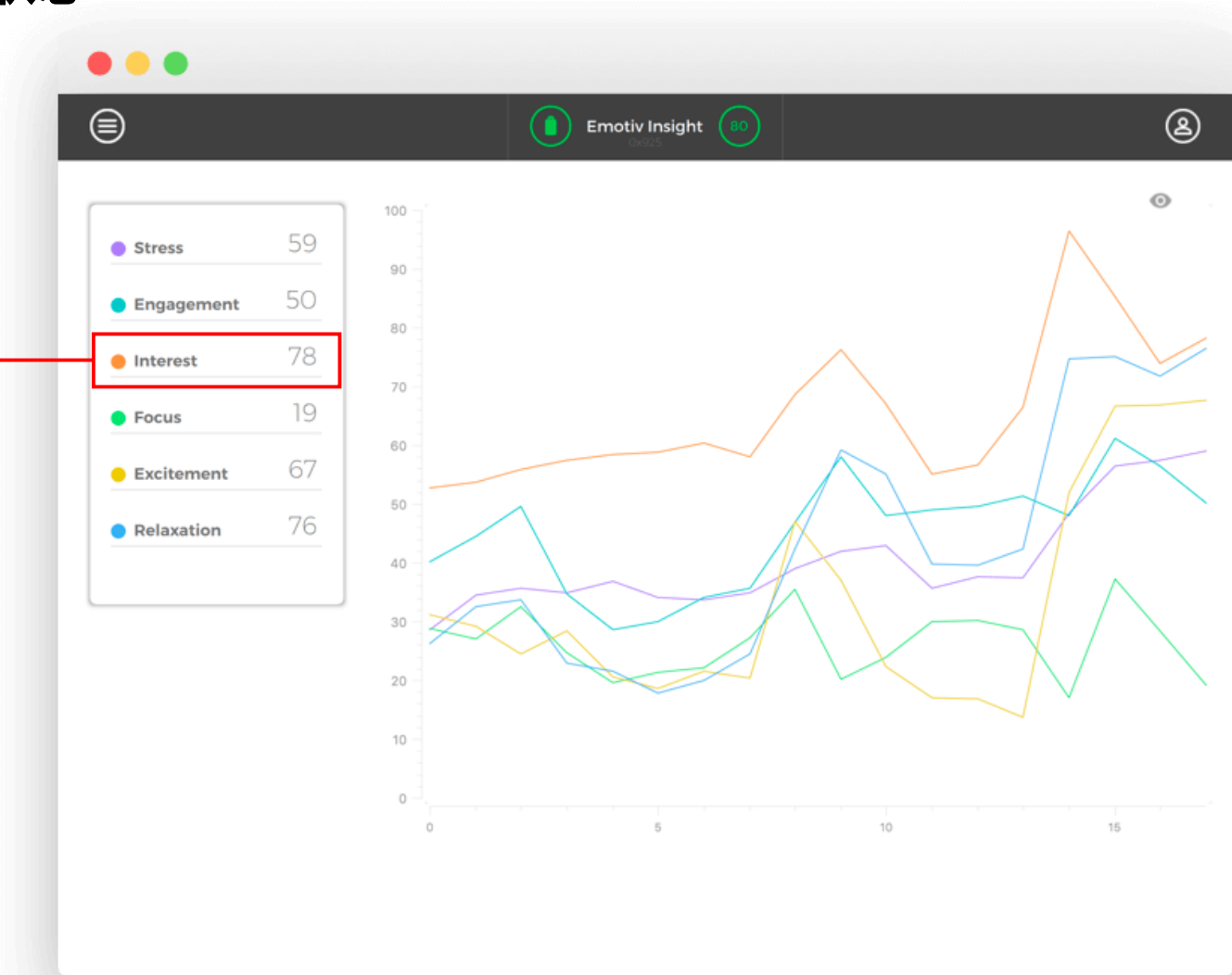
Relaxation



情绪状态分类

■ 自带六种分类状态

Interest指数最高，
判断为最有可能处
于的状态



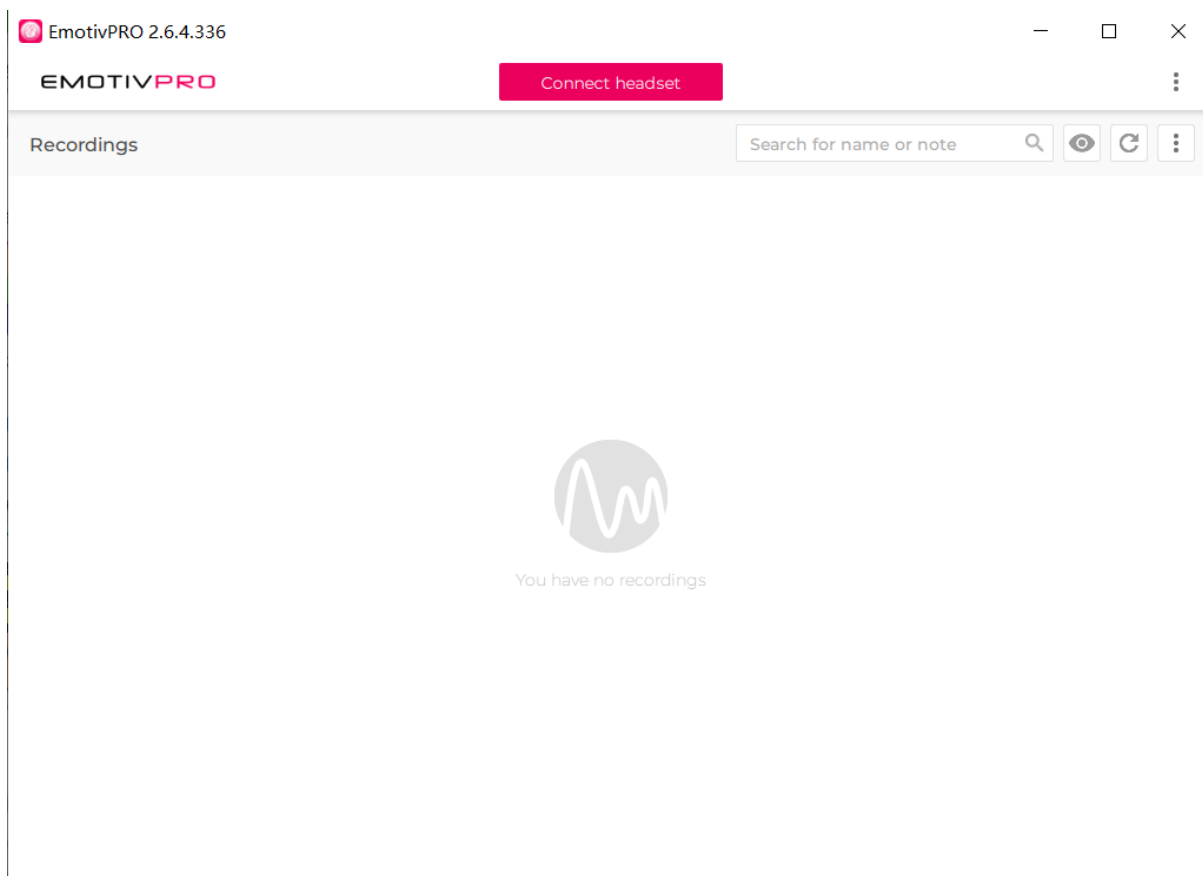
软件平台介绍

■ 数据收集的准备步骤——配置EmotivPRO

打开软件

运行相关程序

完成配置



*具体配置方法会在指导手册中提供

在开发平台上获取信号

- 在开发平台上获取采集的信号，并进行状态判断



编程语言：

请使用Python（版本至少**3.6**）



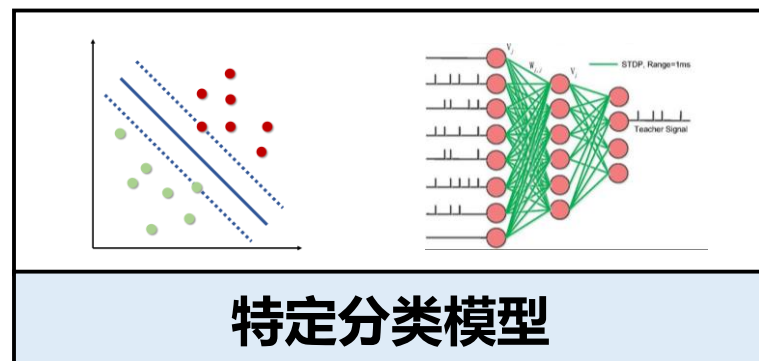
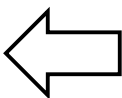
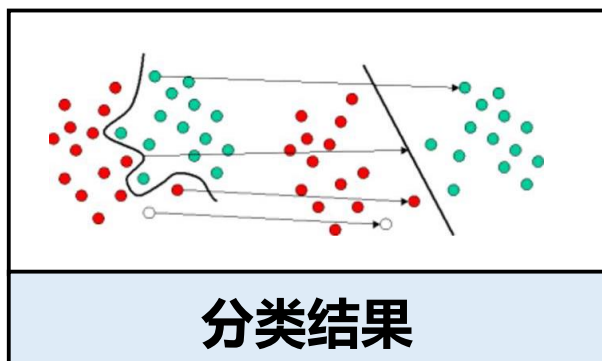
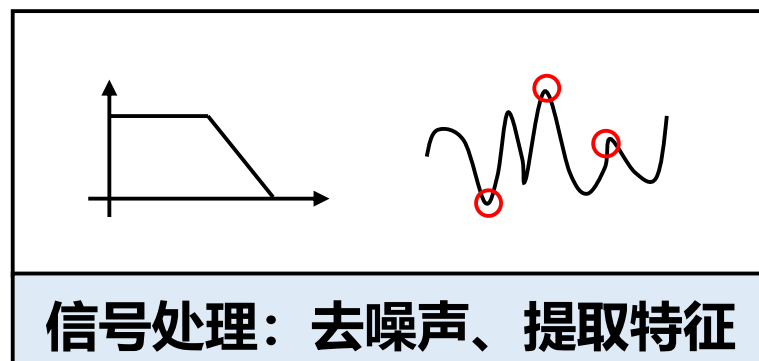
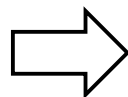
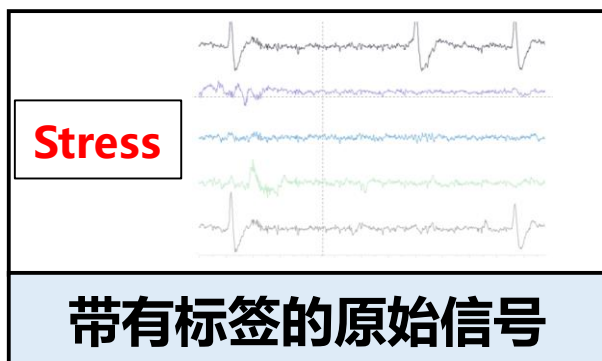
所需要模块包：

预先安装“pylsI”

```
print("looking for a stream...")
streams = resolve_stream('type', 'Motion')
print(streams)
```

通过提供的“ReceiveData.py”文件
在PC端接收数据

■ 如何用原始数据获得分类结果？



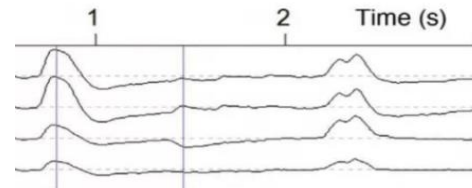
- 选做内容，作为额外创新项
- 自己进行分类，但不脱离提供的六种状态
- 可供选择算法平台：MATLAB、Python、Java.....

滤波算法

■ 噪声来源：眨眼伪迹、工频干扰、基线漂移、其他电信号等

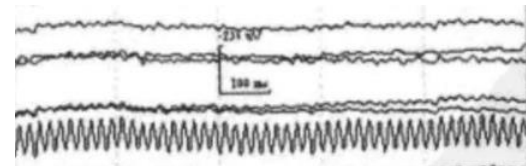
■ 眨眼伪迹：

由于眨眼等行为所产生的的噪声，表现为周期性的尖峰



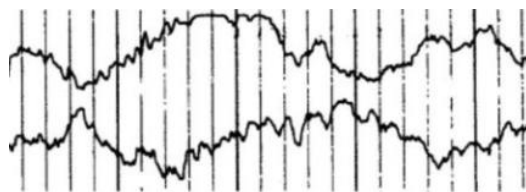
■ 工频干扰：

由市电50/60Hz的频率耦合产生，电极接触不良也会影响



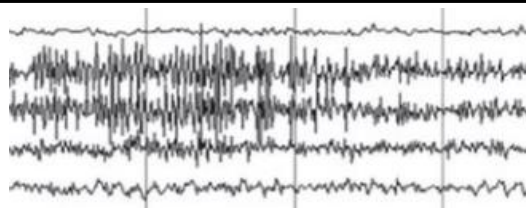
■ 基线漂移：

由采集者的呼吸等行为产生，波形在时域上发生偏移



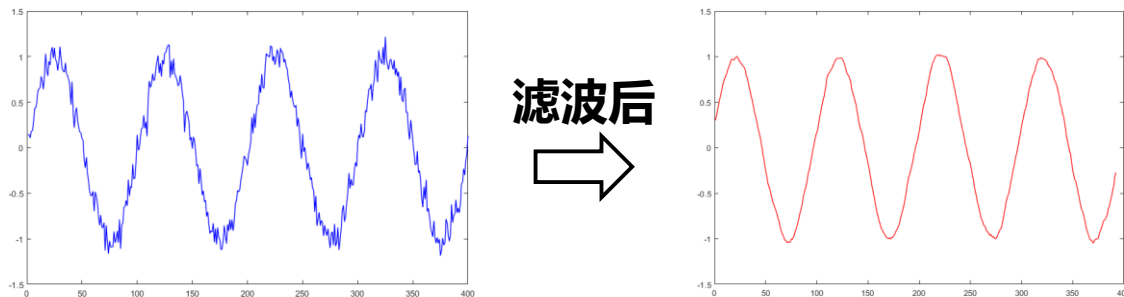
■ 其他电信号：

包括肌电信号、心电信号等，肌电表现为爆发式的噪声



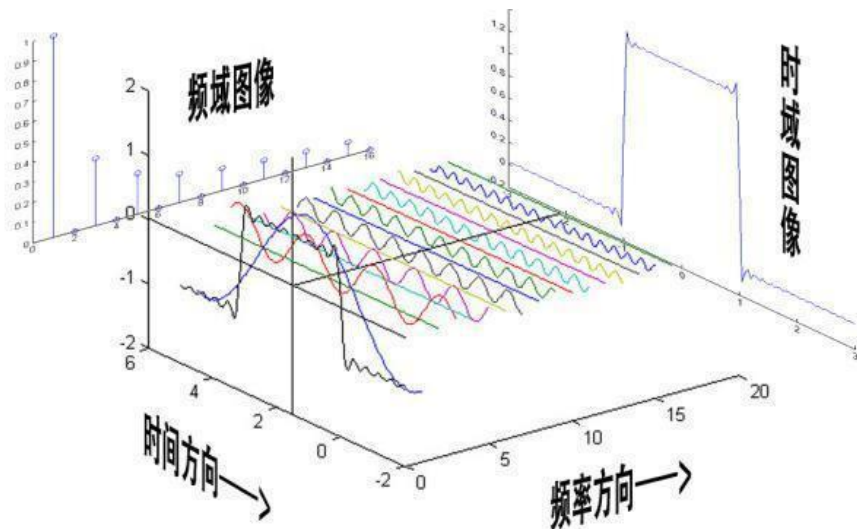
滤波算法

■ 均值滤波：一种简单的低通滤波



基本思想：
取一段时间内的平均值

■ 小波变换：将信号分解为不同的频段



将信号按照频率分解，
得到信号不同频率下的成分

特征提取

■ 提取有效的特征

时间域

常见特征:

极值点

最值

时间间隔

均方根

.....

作为后续模型的输入变量

频率域

脑电信号的频率分布:

Delta: 0.5~4Hz

Theta: 4~8Hz

Alpha: 8~13Hz

Beta: 13~32Hz

Gamma: >32Hz

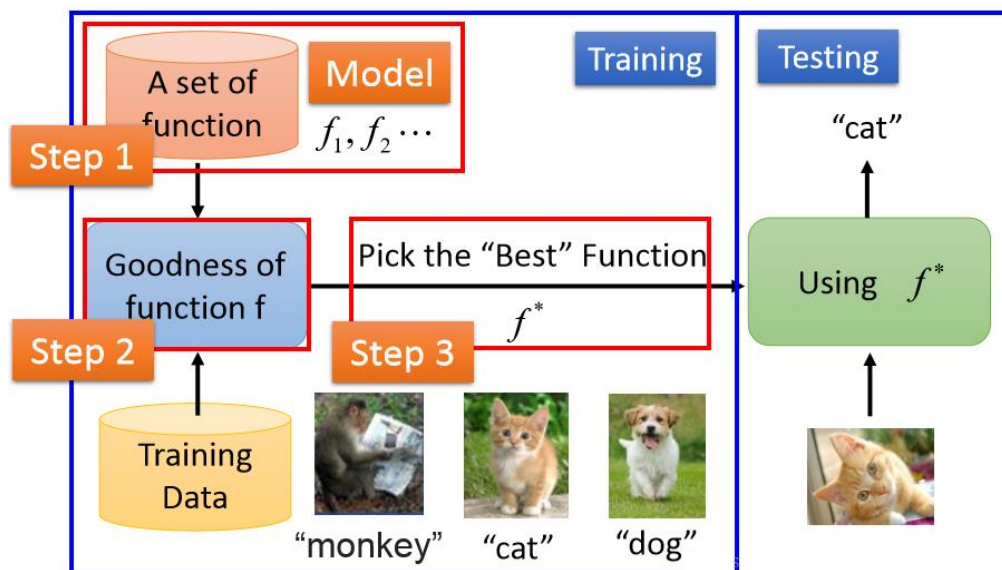
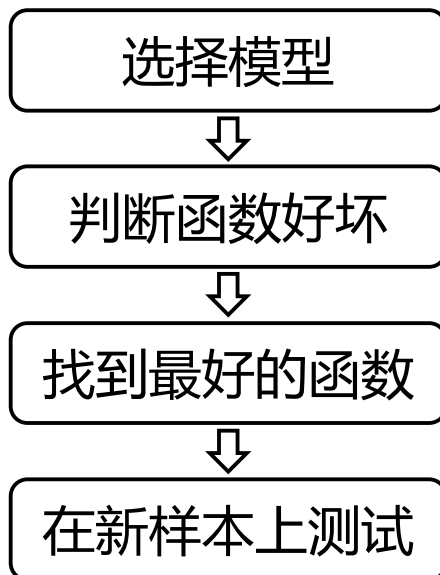
不同频段的信号与人体的
不同状态有关

从原始数据到实现分类：生成自己的模型

机器学习与神经网络

机器学习：

从大量的数据中挖掘出隐含的规律，用来预测或分类

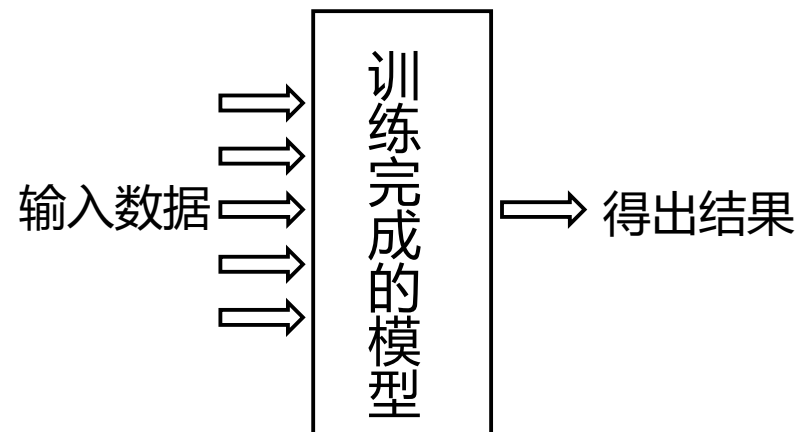
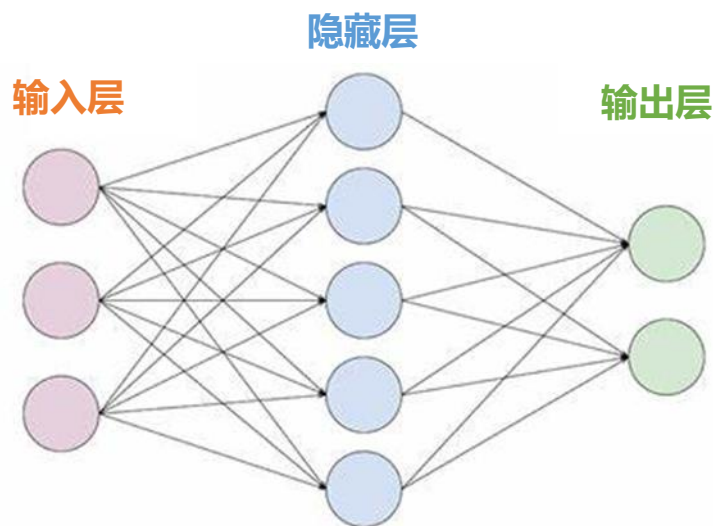


从原始数据到实现分类：生成自己的模型

■ 机器学习与神经网络

神经网络：

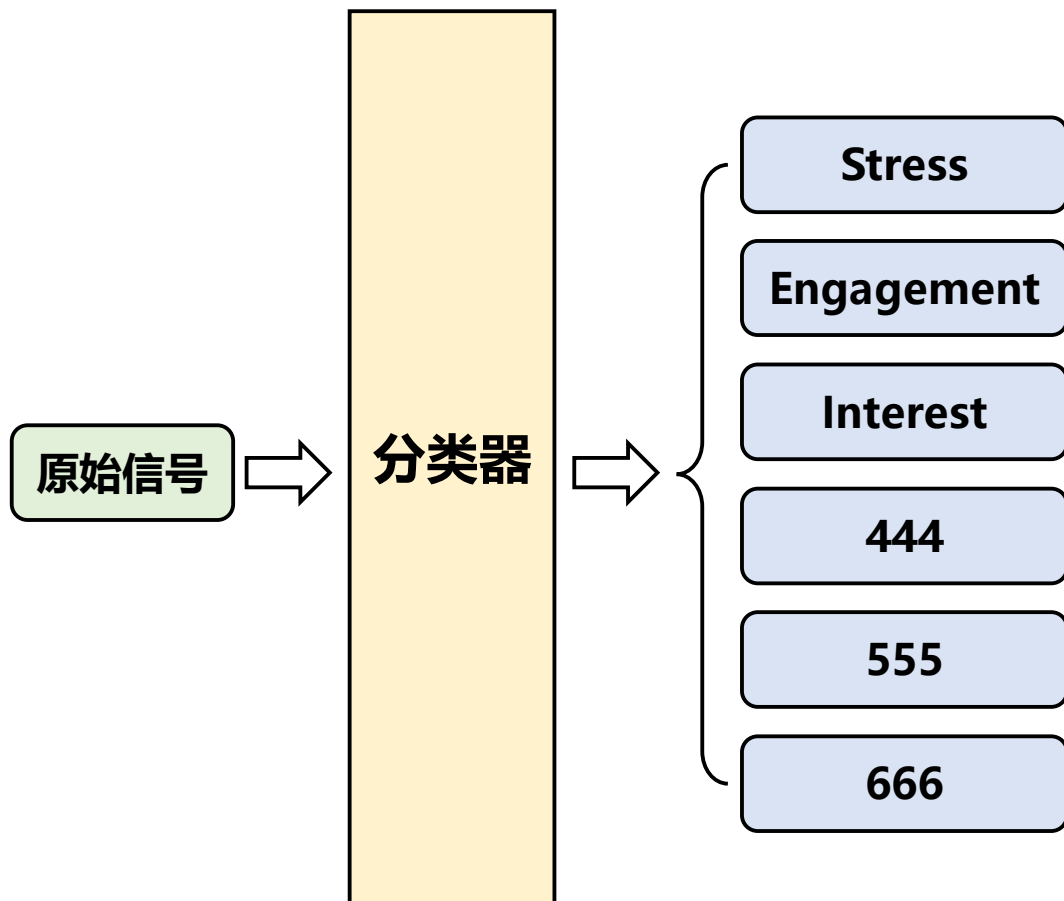
仿照人脑的运作，进行“训练”的过程，向所想要的目标逼近



*具体了解需要查阅更多的资料！

生成自己的分类结果

■ 更详细的分类？



需要完成：

- ✓ 实现一个分类的模型设计
- ✓ 简述采用的方法和训练过程
- ✓ 提供输出结果并验证准确性

注意事项：

本部分仅作为加分项，不会影响基础成绩；

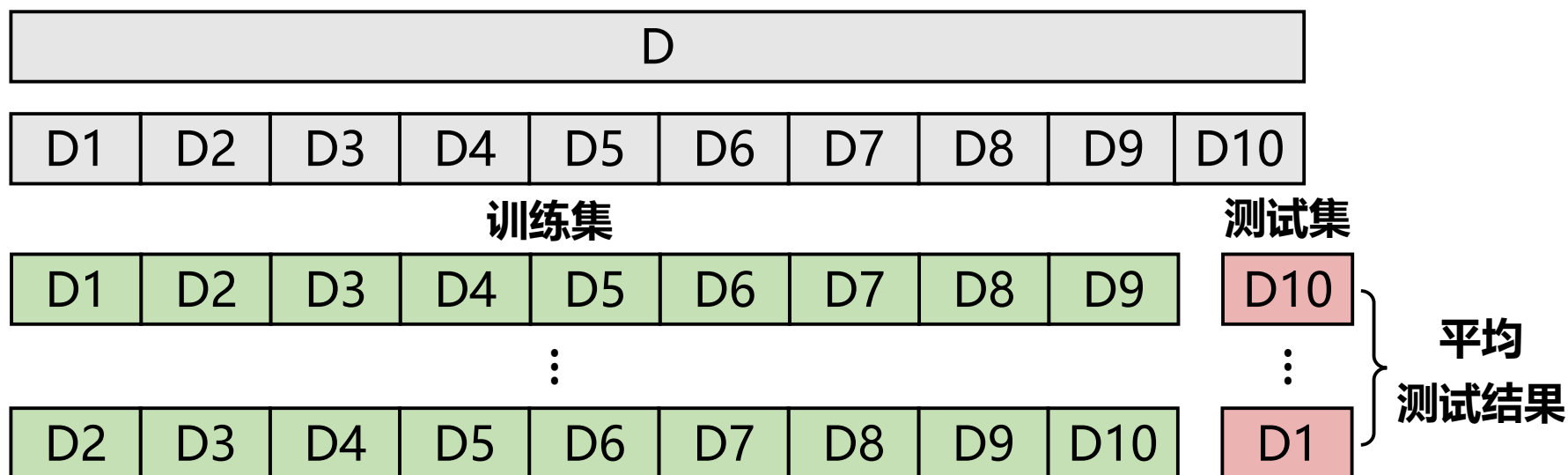
不需要与软件的输出方式一致，采用自己的评判标准，言之有理即可。

对分类结果进行验证

■ 训练集与测试集的划分



■ 交叉验证法



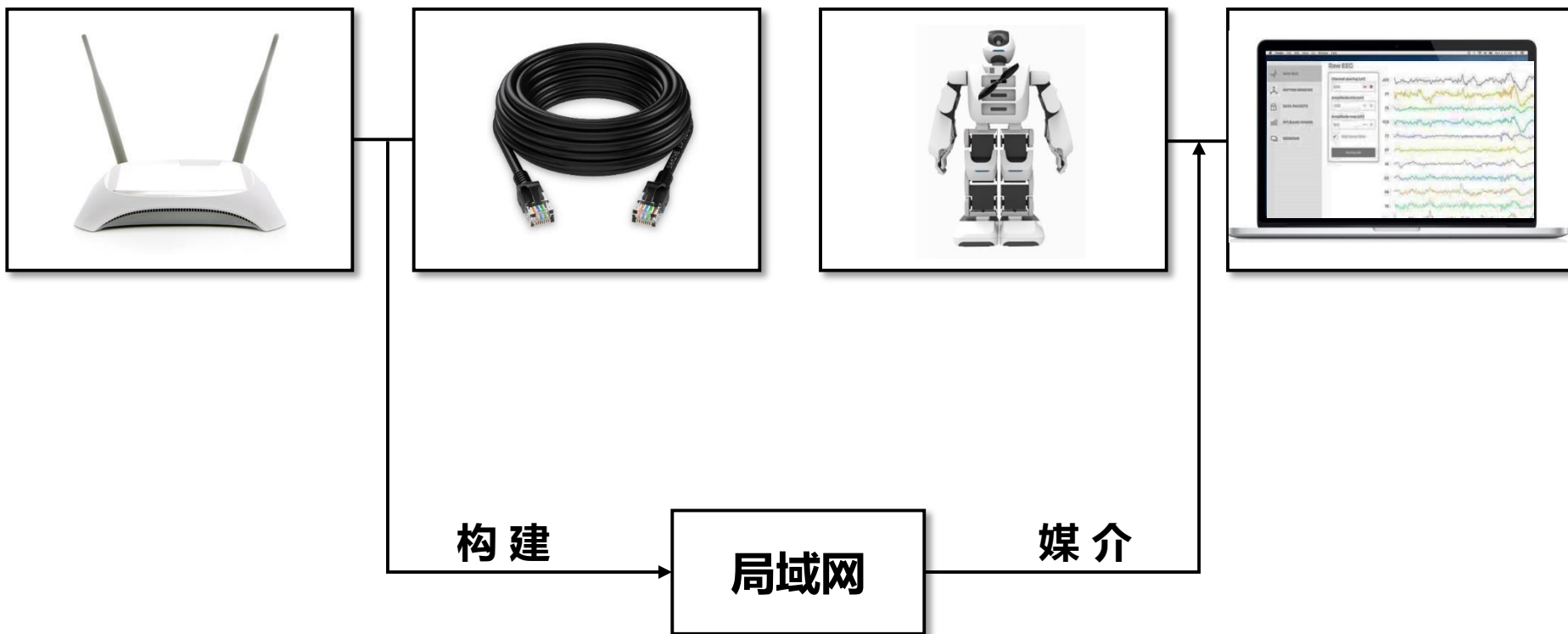
k-1个子集作为训练集，剩下1个子集作为测试集，共可以训练k次

第三部分

机器人动作编程

局域网连接

■ 需要准备的材料（硬件）



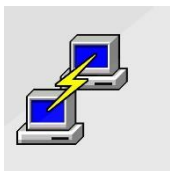
局域网连接

■ 需要准备的材料 (软件)



[MobaXterm]

下载地址: [MobaXterm free Xserver and tabbed SSH client for Windows \(mobatek.net\)](http://mobatek.net)



[PuTTY]

下载地址: [PuTTY: a free SSH and Telnet client \(greenend.org.uk\)](http://greenend.org.uk)



[VNC]

下载地址: [Download VNC Viewer | VNC® Connect \(realvnc.com\)](http://realvnc.com)

局域网连接

■ 使用PuTTY进行SSH连接

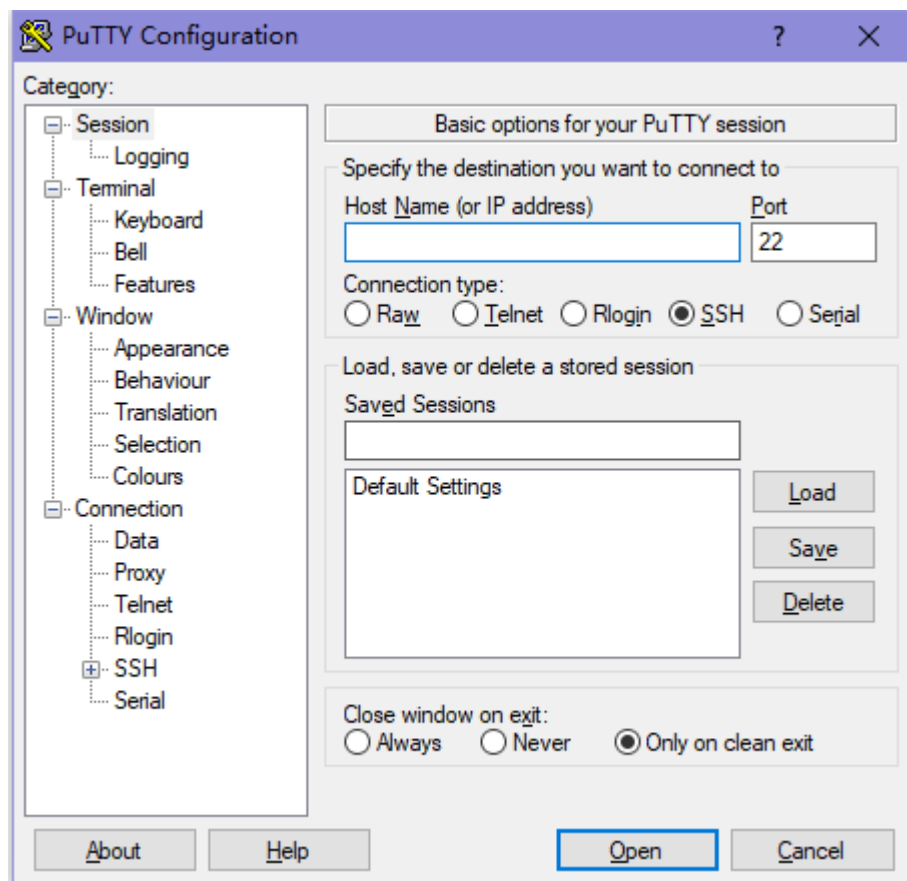
在**Host Name**处输入刚刚看到的**IP地址**



点击**Open**
输入用户名为**pi**
再输入密码为**mm111111**



我们现在成功连上了机器人搭载的树莓派！



局域网连接

■ 机器人与wifi连接

输入以下命令：

- ✓ \$nmcli device wifi list （查看WIFI列表）
- ✓ \$sudo nmcli dev wifi connect "WIFINAME" password "PASSWORD"
- ✓ （连接指定WiFi）
- ✓ \$nmcli dev status （查看连接状态）
- ✓ \$ifconfig （查看WIFI分配的IP地址）

```
pi@raspberrypi:~$ nmcli device wifi list
IN-USE SSID MODE CHAN RATE SIGNAL BARS SECURITY
--
红外 1 130 Mbit/s 100 
红外 1 130 Mbit/s 100 WPA2
红外 4 130 Mbit/s 100 WPA2
红外 6 54 Mbit/s 100 WPA2
红外 8 270 Mbit/s 100 WPA1 WPA2
红外 11 130 Mbit/s 100 WPA2
红外 11 130 Mbit/s 100 WPA2
红外 11 130 Mbit/s 100 WPA1 WPA2
红外 11 130 Mbit/s 100 WPA2
红外 11 65 Mbit/s 100 --
红外 36 270 Mbit/s 100 WPA1 WPA2
红外 36 270 Mbit/s 100 WPA1 WPA2
红外 48 270 Mbit/s 100 WPA2
红外 48 270 Mbit/s 100 WPA2
红外 13 130 Mbit/s 97 WPA2
红外 4 130 Mbit/s 95 WPA2
红外 11 540 Mbit/s 94 WPA1
红外 11 405 Mbit/s 94 WPA1 WPA2
红外 11 65 Mbit/s 94 --
红外 2 130 Mbit/s 92 WPA1 WPA2
红外 6 270 Mbit/s 90 WPA1 WPA2
红外 4 130 Mbit/s 89 WPA2
红外 4 130 Mbit/s 87 WPA2
红外 8 130 Mbit/s 84 WPA2
```

```
pi@raspberrypi:~$ sudo nmcli dev wifi connect WALI_4006EV password doming123
成功用 "f20bfb6d-f539-4ba8-a476-0d7c7b63228c" 激活了设备 "wlan0".

DEVICE TYPE STATE CONNECTION
--
eth0 ethernet 已连接 有线连接 1
wlan0 wifi 已连接 WALI_4006EV
lo loopback 未托管 --
pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.3.20 netmask 255.255.255.0 broadcast 192.168.3.255
    inet6 fe80::5a55:680f:c02e:ce25 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:6d:22:6f txqueuelen 1000 (Ethernet)
    RX packets 1466 bytes 136173 (132.9 KiB)
    RX errors 0 dropped 1 overruns 0 frame 0
    TX packets 1286 bytes 248203 (242.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

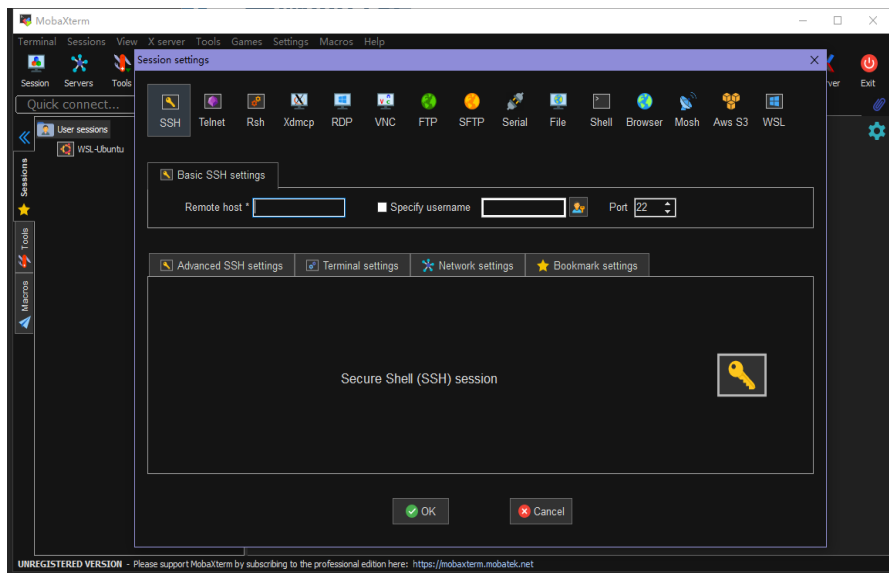
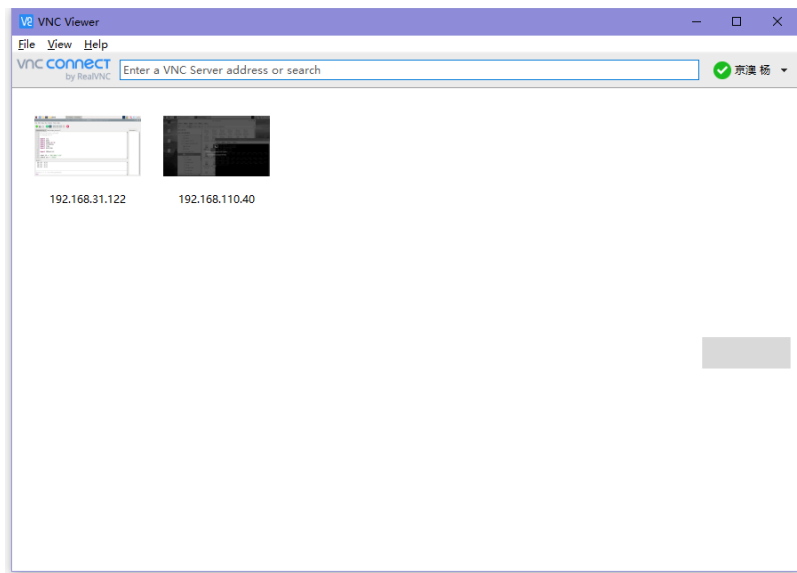
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 25 bytes 1316 (1.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 25 bytes 1316 (1.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.3.14 netmask 255.255.255.0 broadcast 192.168.3.255
    inet6 fe80::1f6:ab8:6540:6abf prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:6d:22:70 txqueuelen 1000 (Ethernet)
    RX packets 12 bytes 1312 (1.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 30 bytes 5016 (4.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```

局域网连接

■ 进入树莓派



- 接下来可以使用IP地址连接树莓派桌面
- 使用MobaXterm通过该IP地址可以轻松在PC和树莓派间传输文件

动作开发

■ 机器人是怎么动起来的？

1

Python调用API
(`action_append('str')`)

2

在lua文件中查询str对应的函数

3

函数中保存的参数 → STM32F103

4

指挥机器人19个舵机运动



LEAVE YOU ~

X

走你~

动作开发

■ 配套软件协助开发

图形化开发直接生成lua文件。



注意需要新建与机器人对应版本的工程才能进行开发！

新建 打开 保存 另存为 下载 代码框 U盘模式 导入动作 设置 语音模块 信道 视频回传 WIFI 联网 无线连接 串口

控制 控制器 基础动作 拳击 足球 自定义 音乐 数学 变量 视觉 函数

开始 XiaDun 动作模块

顺序执行

动作模块包含一组动作

名字	速度	延迟模块	舵机
刚度帧	30	0	4
动作预览	14	500	8
恢复站立			
删除动作			
增加动作			

19个可调节的舵机

版权归属：乐聚（深圳）机器人技术有限公司 版本：1.10.1_release

动作开发

开发完自己想要的动作后，还需要把数据导入机器人

打开项目所在文件夹，打开main.lua文件

名称	修改日期	类型	大小
action	2020/4/28 20:58	文件夹	
import	2020/4/28 20:58	文件夹	
aaaa.abe	2020/4/10 18:27	ABE 文件	1 KB
lib.lua	2020/5/6 15:54	Lua 源文件	15 KB
main.lua	2020/5/6 15:54	Lua 源文件	1 KB
test.xml	2020/5/6 15:54	XML 源文件	8 KB

```
1 dofile('0:/lua/lib.lua')
2
3 while(true)
4 do
5     MOTOrigid16(40,40,40,40,40,40,40,40,40,40,40,40,40,40,40)
6     MOTOssetspeed(14)
7     MOTOmive19(80,30,100,100,142,145,77,100,120,170,100,100,58,55,123,100,128,71,100)
8     MOTOWait()
9
10 end
```

动作开发

■ Lua函数简析

```
1  dofile('0:/lua/lib.lua')
2
3  while(true)
4  do
5      MOTOrigid16(40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40)
6      MOTOssetspeed(14)
7      MOTOmmove19(80,30,100,100,142,145,77,100,120,170,100,100,58,55,123,100,128,71,100)
8      MOTOWait()
9
10 end
```

MOTOrigid16(x1,x2...): 为前16个舵机设置刚度，改变舵机的软硬程度。

MOTOssetspeed(x1,x2,...): 设置舵机运动的速度，值5~150。

MOTOmmove19(x1,x2...): 设置19个舵机移动的参数，范围10~190°

MOTOWait(): 等待动作执行完成。

动作测试

先用按键复位机器人

再使用先前的工具(PuTTY、VNC、MobaXterm) 运行CMDcontrol.py



输入lua函数名即可看到效果

也可以自己编写python文件, 调用action_append('str')函数, 其中str为lua函数名

```
pi@raspberrypi:~ $ ./RunningRobot/CMDcontrol.py
boardreceive_error
please act_name:XiaDun
please act_name:Stand
please act_name:█
```

■ 课程要求总结

- 分组完成本项目，三人一组
- 脑电的原始数据→信号处理→完成分类
- 对机器人的动作进行编程，实现指定动作
- 根据分类结果→完成对机器人的控制
- 提交完整的报告和工程文件，并完成现场演示

极客们，你们的舞台来了！



LEAVE YOU ~
X
走你~