Python 版本3.6.9

```
sudo apt install python3-pip
```

1.安装tensorflow（适用于NX）

NX安装方法选择对应版本（使用的是Python 3.6+**JetPack4.5**）的安装教程就好

```
sudo apt-get install libhdf5-serial-dev hdf5-tools libhdf5-dev zlib1g-dev zip libjpeg8-dev liblapack-dev libblas-dev gfortran
sudo apt-get install python3-pip
sudo pip3 install -U pip testresources setuptools==49.6.0
sudo pip3 install -U numpy==1.16.1 future==0.18.2 mock==3.0.5 h5py==2.10.0 keras_preprocessing==1.1.1 keras_applications==1.0.8 gast==0.2.2 futures protobuf pybind11
# TF-2.x
$ sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redist/jp/v45 tensorflow
# TF-1.15目前使用的是这个版本的
$ sudo pip3 install --pre --extra-index-url https://developer.download.nvidia.com/compute/redist/jp/v45 'tensorflow<2'
```

若安装报错，则去网址下载到本地安装

```
pip3 install 文件路径/文件名.whl
```

2.安装pyqt5,在nano上安装比较特殊（qt5在ubuntu默认库中）

```
sudo apt-get install qt5-default
sudo apt-get install python-pyqt5
sudo apt install python3-pyqtgraph  # 如果是python2则是python2-pyqtgraph
```

如果不是nano/nx则使用`pip install python-qt5`命令安装

3.安装其他环境包（可能不全，但是主要的包都包含在里面，其他的可能需要自己安装）

```
pip install -r requirements.txt # 可能需要换源
```

4.覆盖pylsl下的libls64.so文件

原因：nx是arm架构的，直接pip的是x86架构的，不兼容。

自己进行编译（参考链接）

```
#sudo apt install cmake gcc
#git clone https://github.com/sccn/liblsl
解压liblsl-1.13.0-b7压缩包
cd liblsl-1.13.0-b7 # cd liblsl
mkdir build
cd build
cmake ..
cmake --build . --target install
```

覆盖代码参考：

```
cp -rf liblsl-1.13.0-b7/build/install/LSL/lib/.
/usr/local/lib/python3.6/dist-packages/pylsl
```

重新编译后查看USB端口ls -l /dev/ttyUSB*，运行lsl_data.py检查LSL

4.2 首次使用Ubuntu时，usb设备对用户不开放权限，导致usb转串口数据无法读写。

解决方法：

单次访问权限解决方法：sudo chmod -R 777 /dev/ttyUSB0。

永久权限：~~sudo gedit /etc/group；在dialout: x：20: 后加上username（好像不可以）~~

5.代码结构

ui3.py 界面渲染

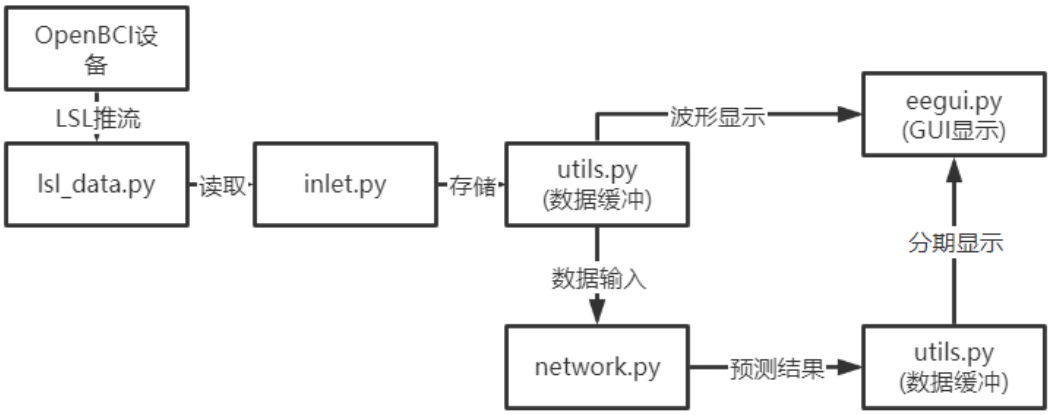
eeg_ui3.py 界面绘图

lsl_data.py nano与OpenBCI建立连接

inlet.py 数据读取并建立数据缓存

network2.py 运行分期网络

utils.py 一些函数方法



6.测试代码

ui_test.py 测试界面能否使用，读取的数据是本地eeg_data.npz的数据

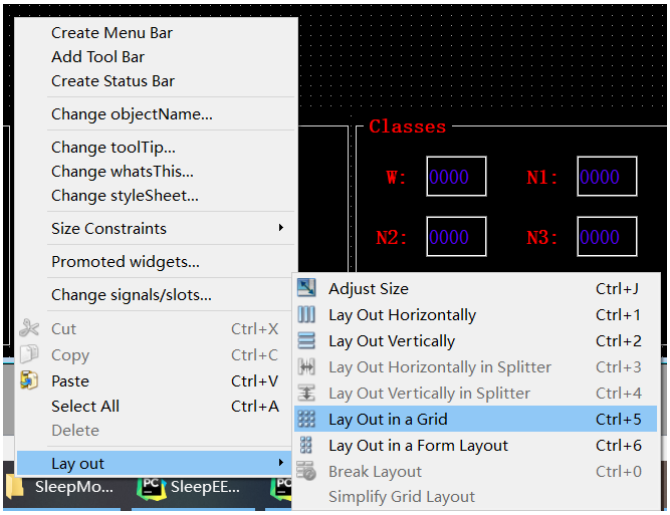
eeg_ui_test.py 测试界面直接读取LSL流的数据（注意，Python的scipy版本包对不对，如果版本太低会报错）

7.运行代码

①python lsl_data.py 与 OpenBCI 建立连接

②新开一个终端：python network2.py 运行网络，显示GUI。

8.QTdesigner使用



lay out in a grid，可以全屏适配

break layout 取消全屏适配

1.[qt5使用]

9.网络更改，修改network2.py文件

```
global pre_labels, filter_buffer  # 主函数部分
i = 0  # 数据集移动index
j = 0  # the index of npz files
start_timestamp = time.time()
while True:
    if epoch_buffer.get_raw_data_state(i):
        raw_data = epoch_buffer.get_raw_data(i)  # filtered raw data type=list
        i = i + 1
        filter_buffer = filter_buffer + raw_data  # save dict x
        # down sample: 1.self-defined method 2.signal.resample
        # data = down_sample(raw_data)
        data = signal.resample(raw_data, 3000)  # type=array
        # normalize each 30s sample such that each has zero mean and unit variance
        tmp_data = np.reshape(data, (1, 3000))
        tmp_data = (tmp_data - np.expand_dims(tmp_data.mean(axis=1), axis=1)) / np.expand_dims(tmp_data.std(axis=1),
                                                                                               axis=1)
        x = np.reshape(tmp_data, (1, 1, 3000))  # predict: input the normalized data (1,1,3000)
        # 替换为预测输出，y_pred, y_list = output
        y_pred, y_list = evaluate_model(hparams, x)  # type list
        pre_labels = pre_labels + y_list  # save dict y
        print_n_samples_each_class(y_pred, classes)
        epoch_buffer.set_label(y_list)
        if time.time() - start_timestamp >= 600:  # 开始保存文件
            save_dict = {
                'x': filter_buffer,
                'y': pre_labels,
                'fs': sample_rate
```

判断数据量是否满足网络输入要求

获取原始数据

数据预处理

预测网络输入结果，

保存预测结果到buffer里

Looks like you're usi

10.[wifi配置]