**Neuracle在线数据接收API文档说明**

**Class DataSeverThread(Thread, )**

特点： 继承模块threading.Thread类

属性：

1. *device*：string

设备类型， 目前支持可以解析的设备： 'Neuracle', 'Neuracle-32', 'DSI-24', 'DSI-7', 'NeuroScan'

1. *n\_chan*：int

上游recorder转发数据的导联个数（nEEGChan + TRG）

1. *srate*：float

EEG放大器设备的采样率

1. *t\_buffer*：float， 单位是秒

用于设定ringbuffer的长度： *nChan by (t\_buffer\*srate).*

方法：

1. *DataServerThread.Connect(hostname,port)*

*hostname*： string

放大器设备转发时主机地址，默认值：‘127.0.0.1’

*port*：int

放大器设备转发时端口号，默认值：‘8712’

创建socket, 上游服务器建立tcp/ip连接

返回值 *notconnnect*：bool

若和上游成功建立连接 *notconnnect = False，*不成功建立连接 *notconnnect = True，*

1. *DataServerThread.ResetDataLenCount(count)*

重置buffer累计更新数据长度，默认值*count =0。*

目的： 在异步算法中，用于判断更新的数据长度

1. *Count= DataServer.GetDataLenCounter()*

获取当前buffer已经累计更新的数据长度，单位点数

1. *DataServerThread.ResetTriggerChanOfBuff(triggerList)*

重置buffer的最后一行/导的trigger数据，data的长度**必须**和buffer的长度一致， 默认设置 *triggerList = zeros(1,(bufferSize\* sampleRate))*

目的：在同步算法中，检测trigger事件并满足处理数据长度后，单次SSVEP刺激结束，给出一次反馈，防止定时器20ms轮询重复计算和输出。

1. *[data] = DataServerThread.GetBufferData()*

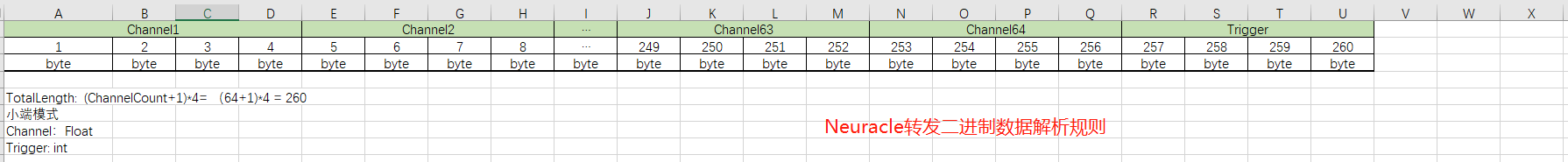
获取buffer数组

1. *DataServerThread.start()*

开启线程，并获取上游服务器socket数据

1. *DataServerThread.stop()*

停止线程



**Class Ringbuffer (n\_chan,n\_points)**

特点：先进先出

属性：

1. *buffer* : numpy.array

buffer矩阵的维度： n\_chan \* n\_points

1. *currenPtr*: int

buffer当前的指针位置

1. *nUpdate*：int

buffer 当前更新的累积点数

方法：

1. *RingBuffer.appendBuffer*(self, data)

将解析后数据包data喂入*buffer*， 并对*currentPtr*和*nUpdate*更新

1. *RingBuffer.getData*(self)

获取buffer内容

1. *RingBuffer.resetBuffer*(self)

重置*buffer*：清空*buffer*，*currentPtr* = 0， *nUpdate* = 0

PS： 此*RingBuffer*模仿了C语言对指针的操作，目的加快matlab对矩阵操作

**Demo：**

|  |
| --- |
|  |