分割训练测试集的时候训练集比例不宜太高，0.5-0.75左右即可，过高容易过拟合，导致测试集/预测数据降维后漂移

插值维度可以缩小，实测10-20维效果最佳且损失不大（往小了插值实际上效果和滤波差不多，能排除掉噪声和极端值，反而能提高模型鲁棒性）

试用了两层全连接神经网络，120-64-10，效果感觉比LDA好；考虑到esp32堆栈很小，神经网络尽可能压缩，尝试hidden层设置为32节点，正确率没有明显下降，

删除了letterH的动作数据，因为这个动作和phi，W，lighting都有一定程度的相似；增加动作到16中（后面不需要的话可以把上下左右动作删了，操作不当容易误测）

Esp32增加配置文件（WIFIconfig，类别数 等）烧录在flash中，可以通过webui进行设置；无法连接WIFI时启动自身热点，通过网关ip访问webui修改WIFI配置

训练时增加导出为json的功能，esp32从flash中通过json加载模型，包括one\_hot编码的action类型文件，torch.nn的权重文件（weight和bias）；

重新手写relu函数，插值函数，前向传播，将***插值降维120dim->经过神经网络forward得到10dim的output->获取最大值index，通过索引在actions中得到预测动作名称***的流程移植到esp32上，推理阶段可以不再依赖PC上部署的python

训练时尝试2层卷积层和一层fc，效果更好，模型更小

Output之后增加softmax，最大概率>80%才算预测成功

移植8266，头文件引入顺序如下，不然会出现冲突

#include <WebSocketsServer\_Generic.h>

#include <ESP8266WebServer.h>

1. 使用WebSockets\_Generic包，8266和32通用
2. ESP8266WebServer不能在前面，因为两个包重复定义了一个函数

8266内存不足，根据官方文档“当 ESP8266EX 运⾏在 Station 模式下，连上 AP 后，在 Heap + Data 区⽤户可⽤ SRAM 空间最⾼为 50 kB。”