第三次实验

一、实验目的

尝试minibatch的构建，看是否可行。

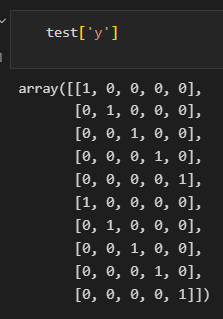
二、Minibatch的构建

我们写一个程序从每个受试者当中做随机抽取，对于五分类问题，这样两个受试者间的对比就有5+4+2+3+1共15个，其中5个正对，10个负队。然后再循环的抽取，来构造N个受试者间的对比对。为了实现对比，代码层面我们把这些数据人为排列（其实就是用代码循环排列好）好，然后batch\_size取10，为了实现对比。也就是有受试者A和B。那么前五个就是A的五种状态，后五个是B的五种状态，从而对比。

那么现在对于抽取数据有两个思路：1. Absolute Random完全随机的在受试者的不同状态中抽取，实现受试者间的对比。2. Data Augmentation指定好抽取的那一个30s片段构造对比对，然后在训练分类器的时候做数据增强。

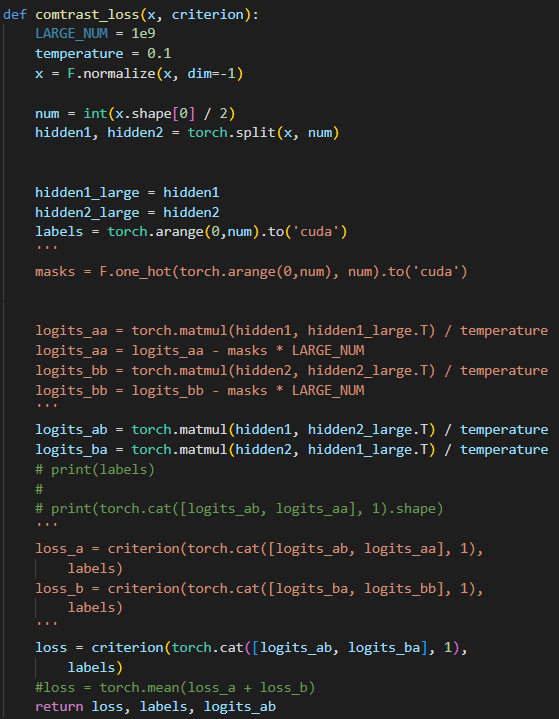
三、实验记录

我们将完全随机的抽取，对受试者A和B得到以下的array：



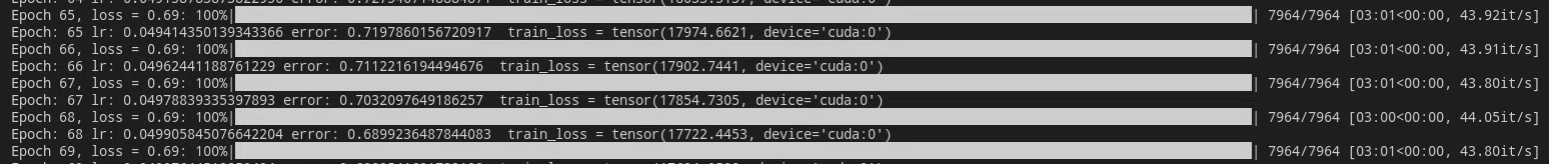
从标签上看，其中上面五行是A的五种状态，下面五行是B的五种状态。这样就构成了我们的minibatch。考虑到我们有的健康受试者大约有70人，总共152个采集好的文件。那么构成的minibatch的数量是152\*(152-1)/2。大概是10000左右，然后我们将这些minibatch拼接成一个文件，然后batch\_size取10，使得计算对比损失的时候是在者一个minibatch内完成的。

然后尝试在原来的代码基础上修改对比损失的计算，使实现跨受试者的对比。

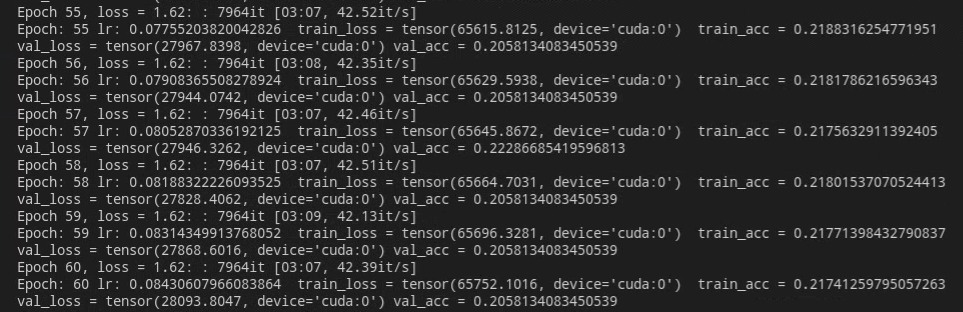


四、实验结果

在对比过程中，虽然误差在下降但是很慢。此外，每一轮的loss值都为0.69。这似乎是因为经过交叉熵函数的计算，有-log(0.5)=0.69。也就是说概率是0.5。换句话说就是在对比的时候就判断不了哪个是哪个，只能靠猜。这一点在后面的分类器训练更为明显。



可以看到分类器的损失是一样的，并且训练集和测试集的精确度都为20%左右，有因为这是个五分类问题，换句话说模型就是在完全瞎猜。



总之，现在模型存在很大的问题，暂时没法实现我想要的效果，很多细节的地方需要重新设计而不是简单的改别人的代码。

五、实验计划

实际上，对于先前的设计存在一个大的问题。对于batch\_size取10，使得计算对比损失的时候是在这一个minibatch内完成的。换个角度看，在一个minibatch上计算这其实就是将batch取为了1，是没有意义的。首先先要做的是在数据的读取上面修改，使得实际的batch不只是1。一个想法是类似于做图像任务的方法。将原来一万多个minibatch的.npz文件的每个minibatch理解为一张图片，而图像任务一次会读取很多图片，从而实现将实际的batch提升。

关于对比损失的计算我在参考SimCLR的设计。