# AI基础lab4实验报告

（There are ghosts every where）

##### Recurrent Layers-torch.nn.LSTM介绍

该类继承于基本类torch.nn.RNNBase:

torch.nn.LSTM(*input\_size*, *hidden\_size*, *num\_layers=1*, *bias=True*, *batch\_first=False*, *dropout=0.0*, *bidirectional=False*, *proj\_size=0*, *device=None*, *dtype=None*)

是预设好的基础长短期记忆递归神经网络的基础架构（torch.nn.LSTMCell的集合版本）。

关于输入输出情况（默认为多输入同步多输出）：

输入（张量）：shape（序列长，input\_size（每个时间步的信息））或（序列长，batch\_size，input\_size）或（batch\_size，序列长，input\_size）note：整条序列的信息是一次性输入的

输出（元组）：output与（h\_n，c\_n）：output（张量）shape（序列长，D \* output\_size）或（序列长，batch\_size，D \* output\_size）或（batch\_size，序列长，D \* output\_size）当开启双向递归，D = 2，否则D = 1；当proj\_size == 0 时，output\_size = hidden\_size，否则output\_size = proj\_size。h\_n，c\_n为短期，长期记忆数据，根据训练过程确定shape。

关于初始化参数：

input\_size：输入的每个时间步信息大小

hidden\_size：长期记忆（细胞状态）的信息大小

num\_layers：单元堆叠深度

bias：是否开启偏置

batch\_first：输入数据的batch\_size是否处于第一位

dropout：dropout比例

bidirectional：是否开启双向递归

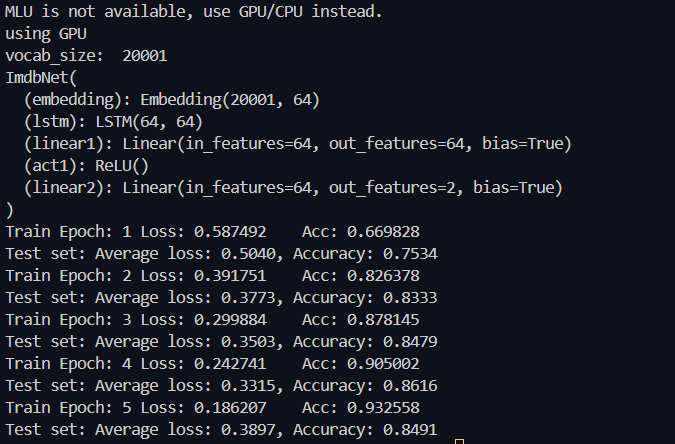
proj\_size：短期记忆（隐藏状态）投影大小，若proj\_size == 0，则短期记忆大小等同于hidden\_size

device：存储以及运算设备

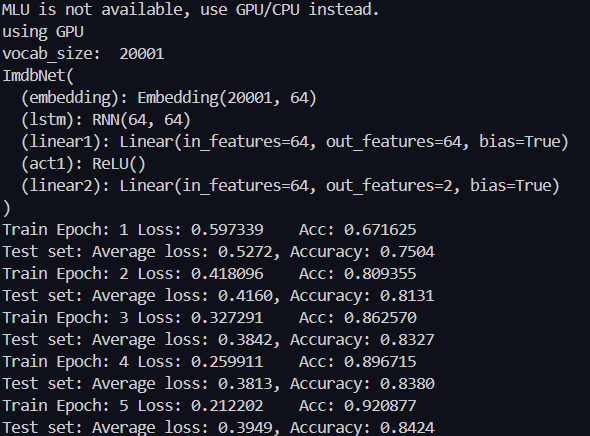
dtype：是否为dtype类型对象

1. **三种模型对比实验结果**

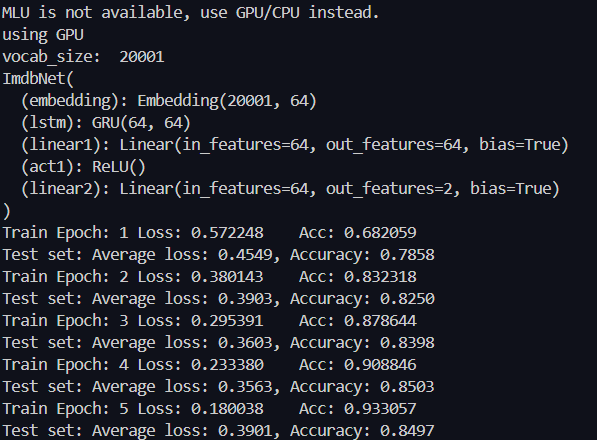
LSTM：



RNN：



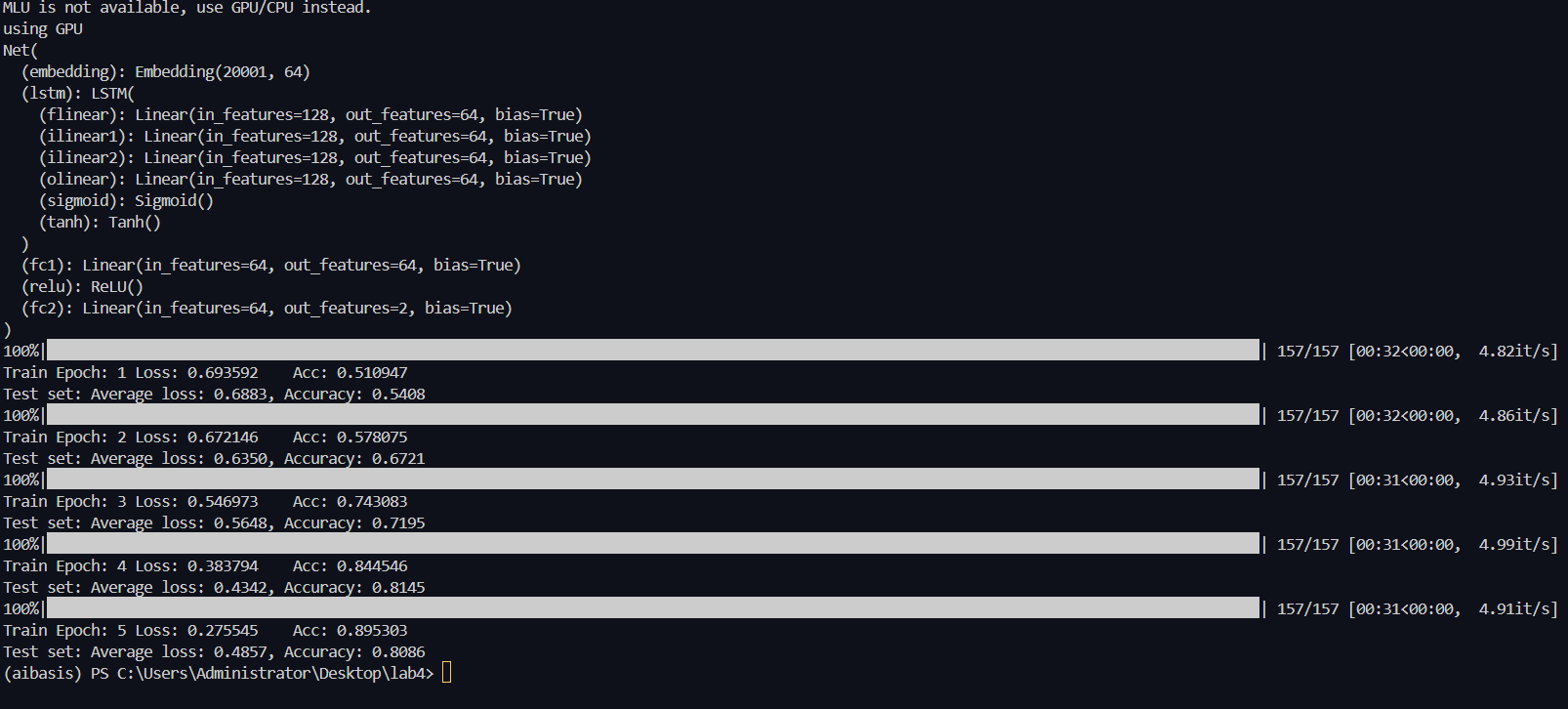
GRU：



最终三种方案的运行结果其实相差不大，我甚至怀疑5个epoch过短，很多工作都是由最后的MLP完成的。

1. **手写lstm:**

基础达标结果：

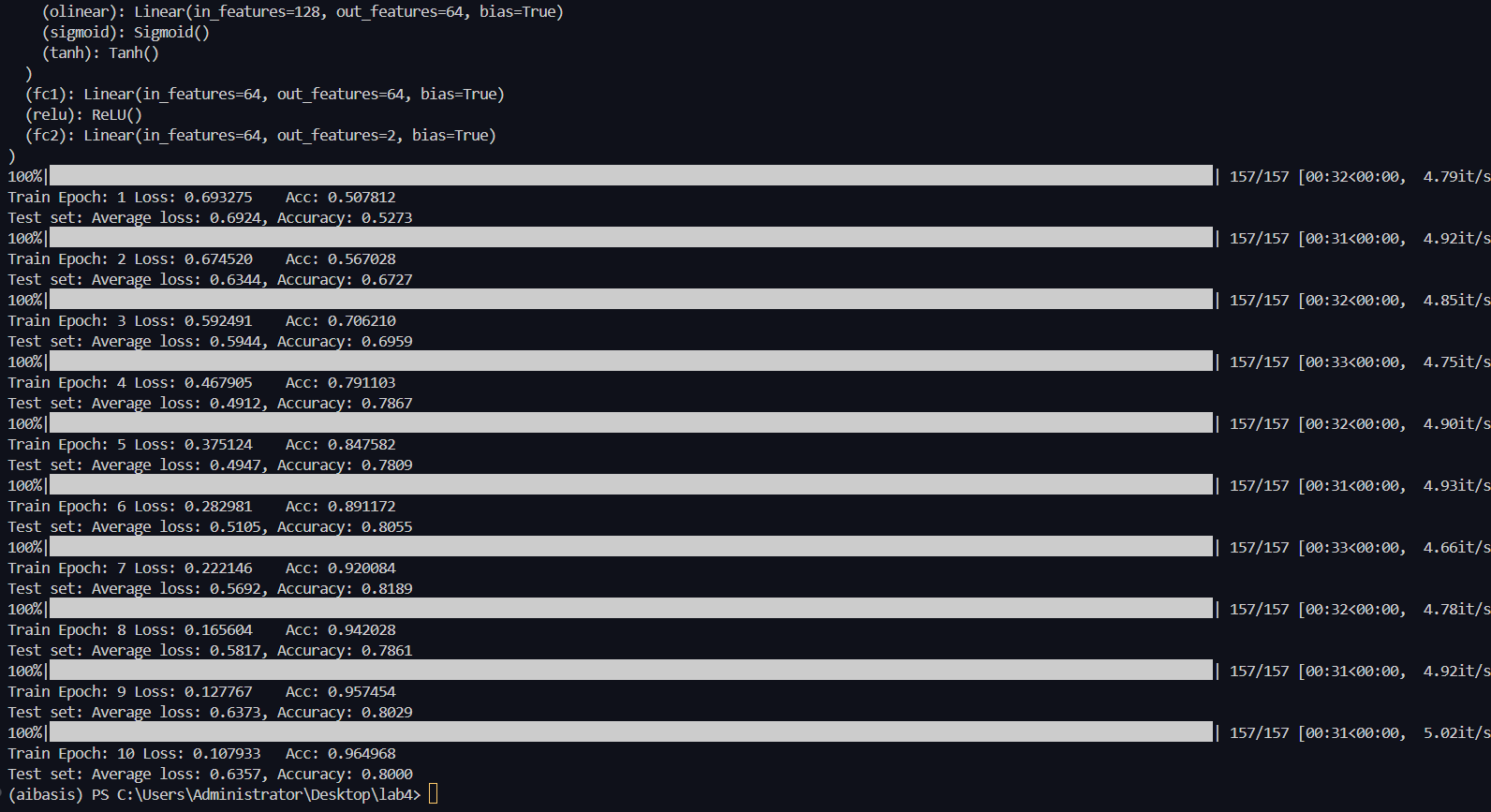


最好一次可以达到0.81

调整实验：

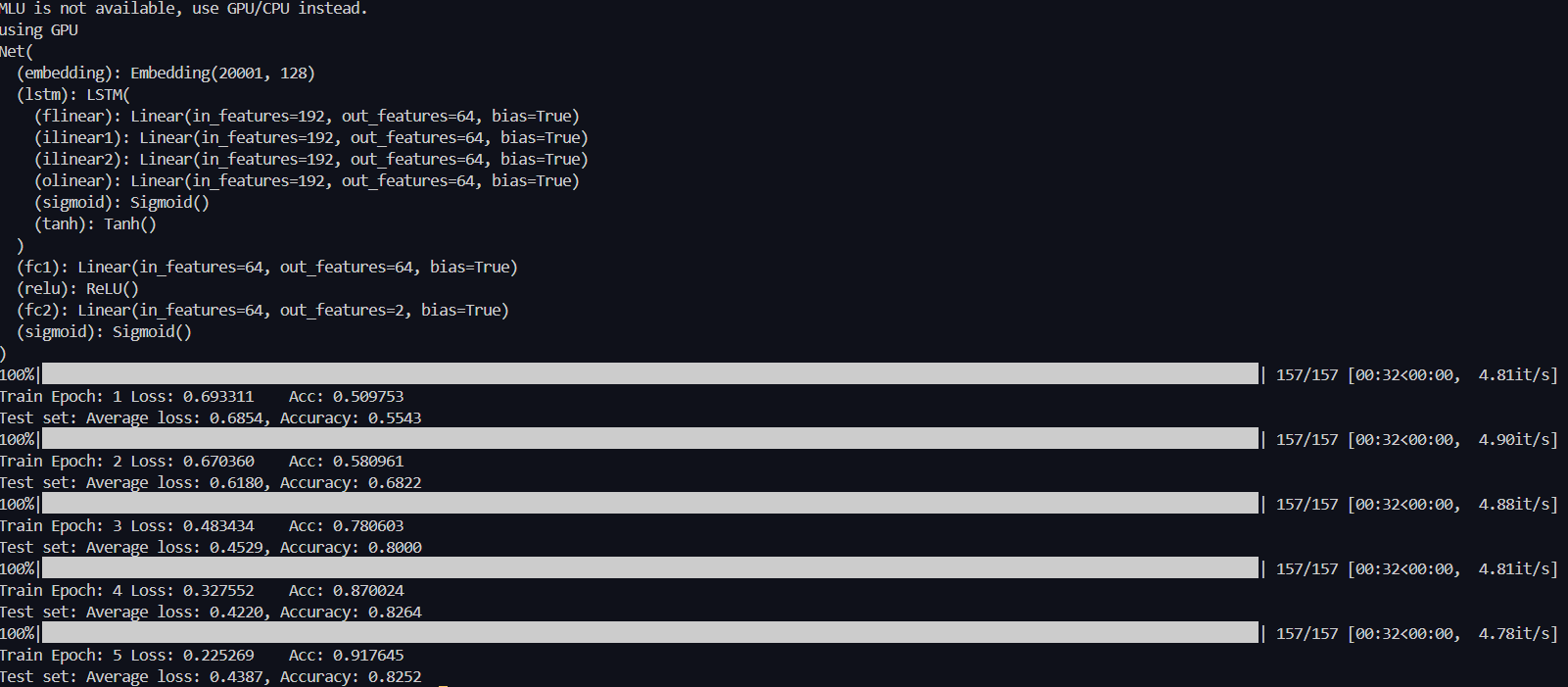
隐藏层结构：隐藏层size调整后并未有太大变化，怀疑是因为最后两个全连接层收敛太快

训练流程：把batch\_size加大到了128，可以略微提高运行速度（其实通过矩阵运算可以再提高速度，但是没有想清楚）；epoch提高到10用处也不大：



损失函数：调整为均方差之后效果差，接近0.6（自行补充了一个sigmoid输出）

Embedding：尝试把embedding维数翻倍，有两个百分点的提高：



最终结果：可以达到大概82%