**人工智能基础LAB1实验报告**

（Zhuiy here）

1. **np\_mnist.py**
2. 基础实现与代码分析（详见np\_mnist.py）

在实现基本代码后，我做了如下改动与尝试：

* 1. 学习率自动递减：在某个epoch完成后，若loss不降反增，则降低学习率。
  2. 每个epoch训练前打乱训练数据的顺序，避免模型学习顺序信息。
  3. 允许自定义并尝试多种激活函数、损失函数、隐藏层神经元数量、初始学习率、学习率递减比例、正则方法与正则比例等等，提高自由度。
  4. 为提高训练效率，不再在每一个step后计算一次accuracy，改为每个epoch后计算一次。

注：必须要求工作目录位于同级文件夹内才能正确运行

1. 实验结果及分析

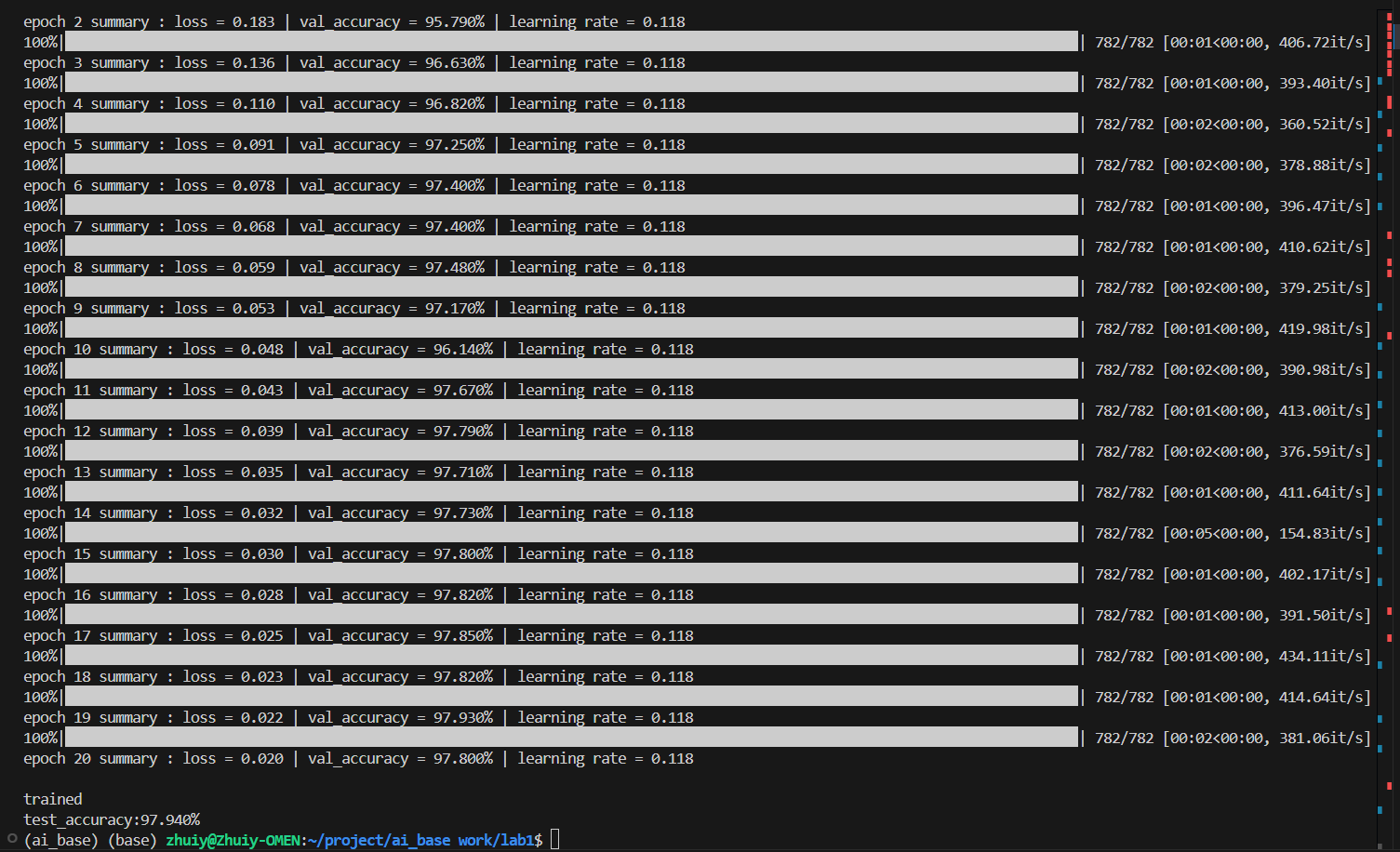
目前而言，最佳的超参数组合（单隐藏层）为：

hidden\_size = 300；batch\_size = 64；lr = 0.13

lrd（学习率递减倍率） = 1.1；epoch = 20；Activator\_type = 'relu'；Loss\_func\_type = 'compute'；Regulation\_type = 'L2'

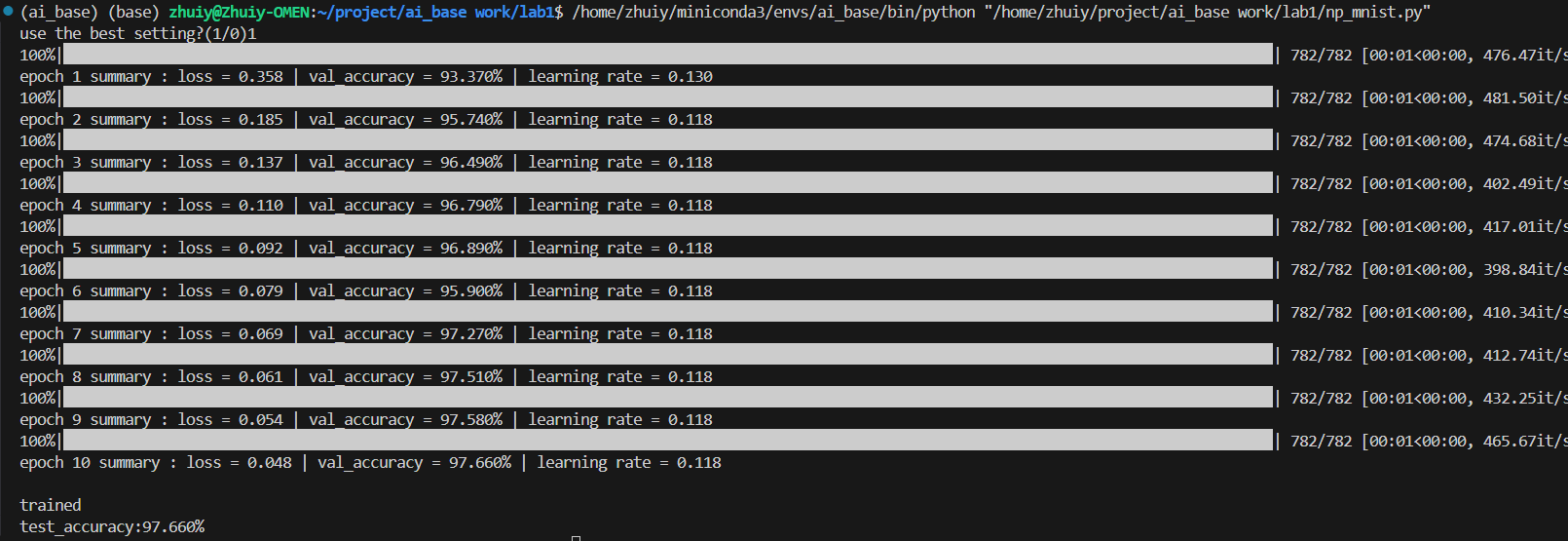
Regulation\_rate = 0.00005

可以达到如下效果：

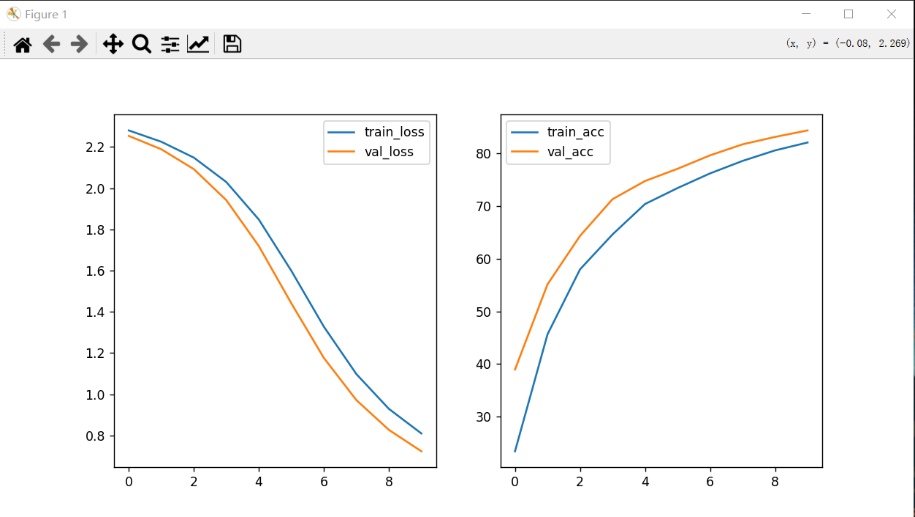


其中‘relu’激活函数的效率远高于其他两种，这很有可能是因为‘relu’可以很好地避免梯度消失问题；同时，交叉熵作为损失函数效果优于协方差，这其中，有softmax-compute联合梯度可以优化运算，减少误差的原因。

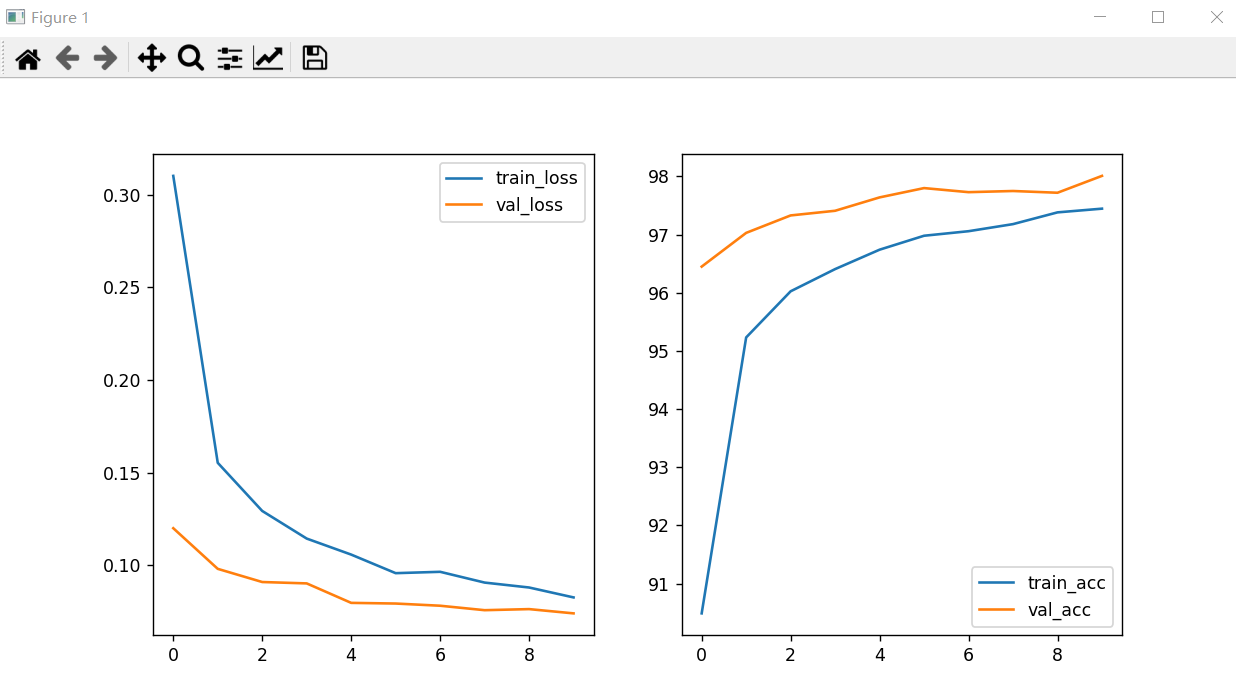
特别地，将隐藏层从一个扩为两个并无法有效地提升正确率，仍约为98%，考虑到计算效率，本代码保留但隐藏层的情况。

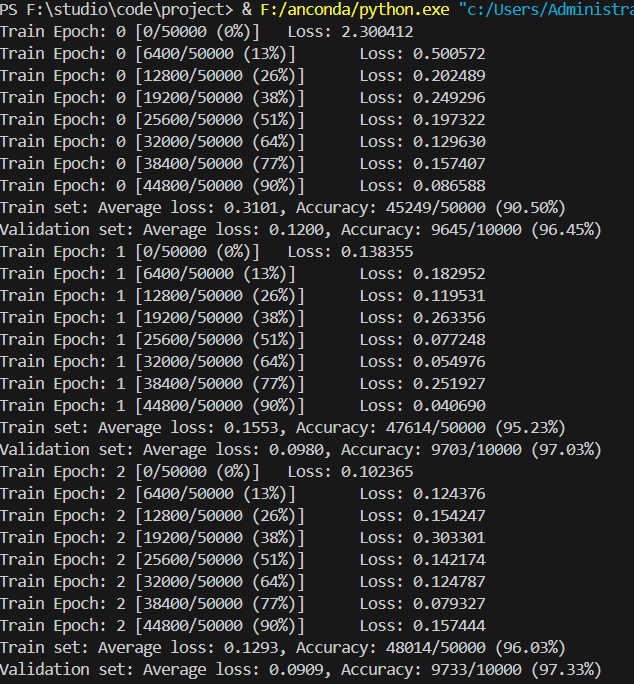
附：强制要求epoch=10正确率：

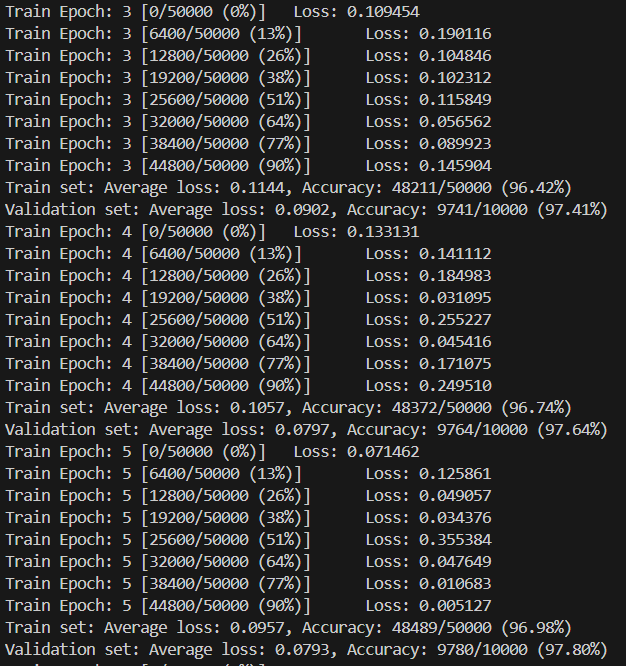
1. **torch\_mnist.py**
2. 代码改动与效果：
3. 源代码：

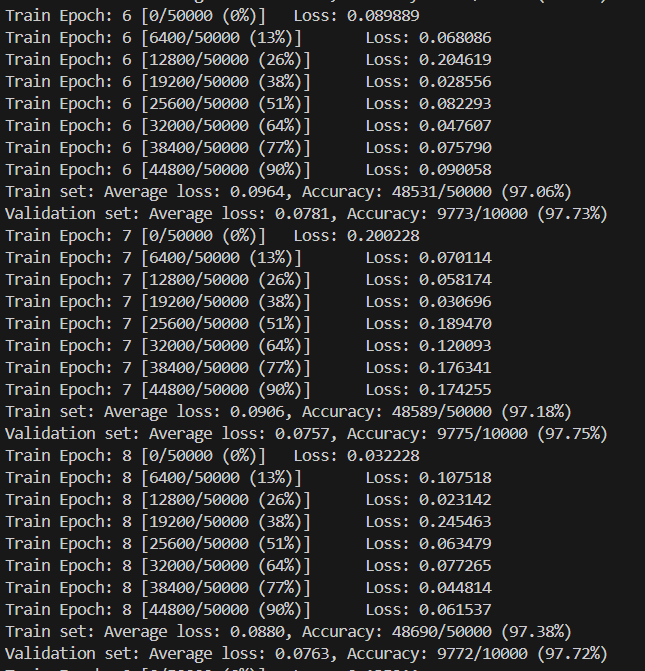
****

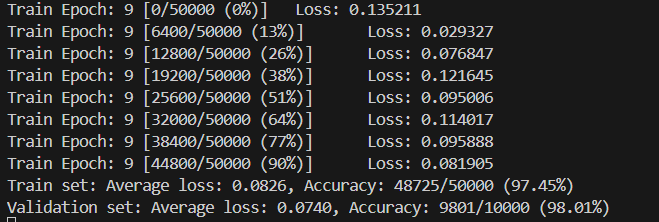
1. 加入dropout层，改为Adam优化，效果好了很多：





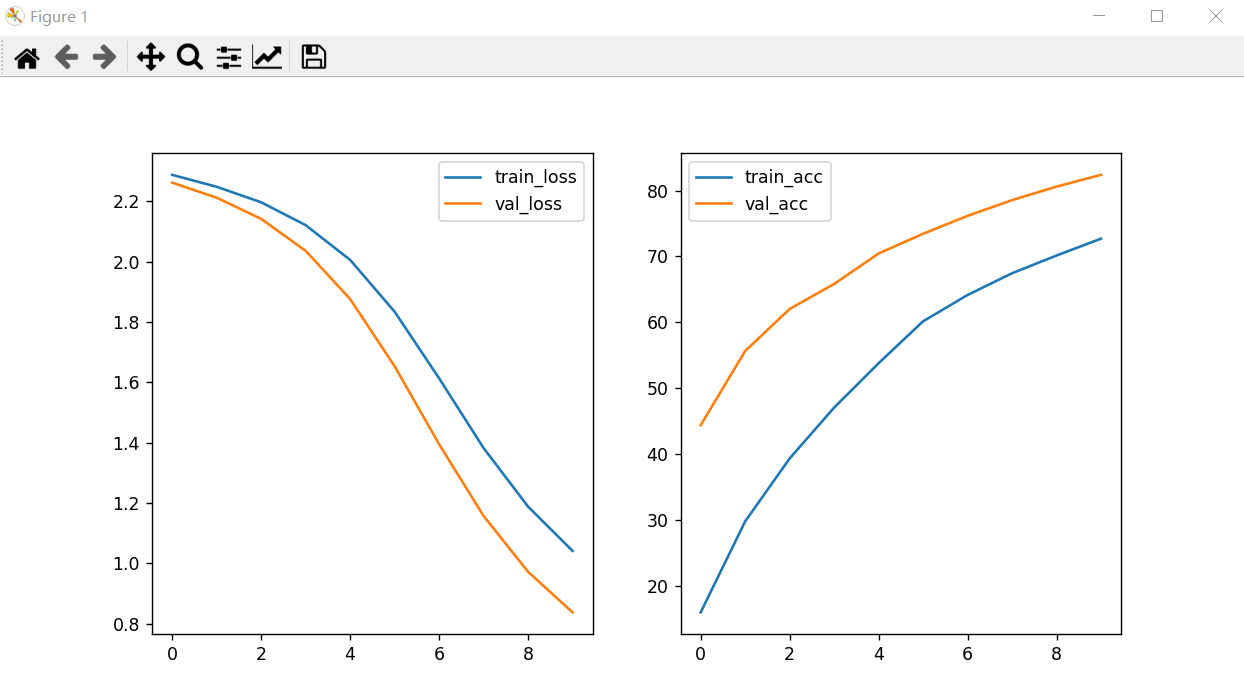








1. b)中Adam改为SGD效果差：s





1. 多一层+Adam+dropout，效果竟然不好，过拟合导致的：

