**深度学习实验报告**

实验1：深度学习框架熟悉

姓名：姚舜宇

学号：1190202107

班级：1903602

**实验环境：**

Windows11

Python3.8

PyTorch1.10.2

**实验内容：**

使用PyTorch实现MLP

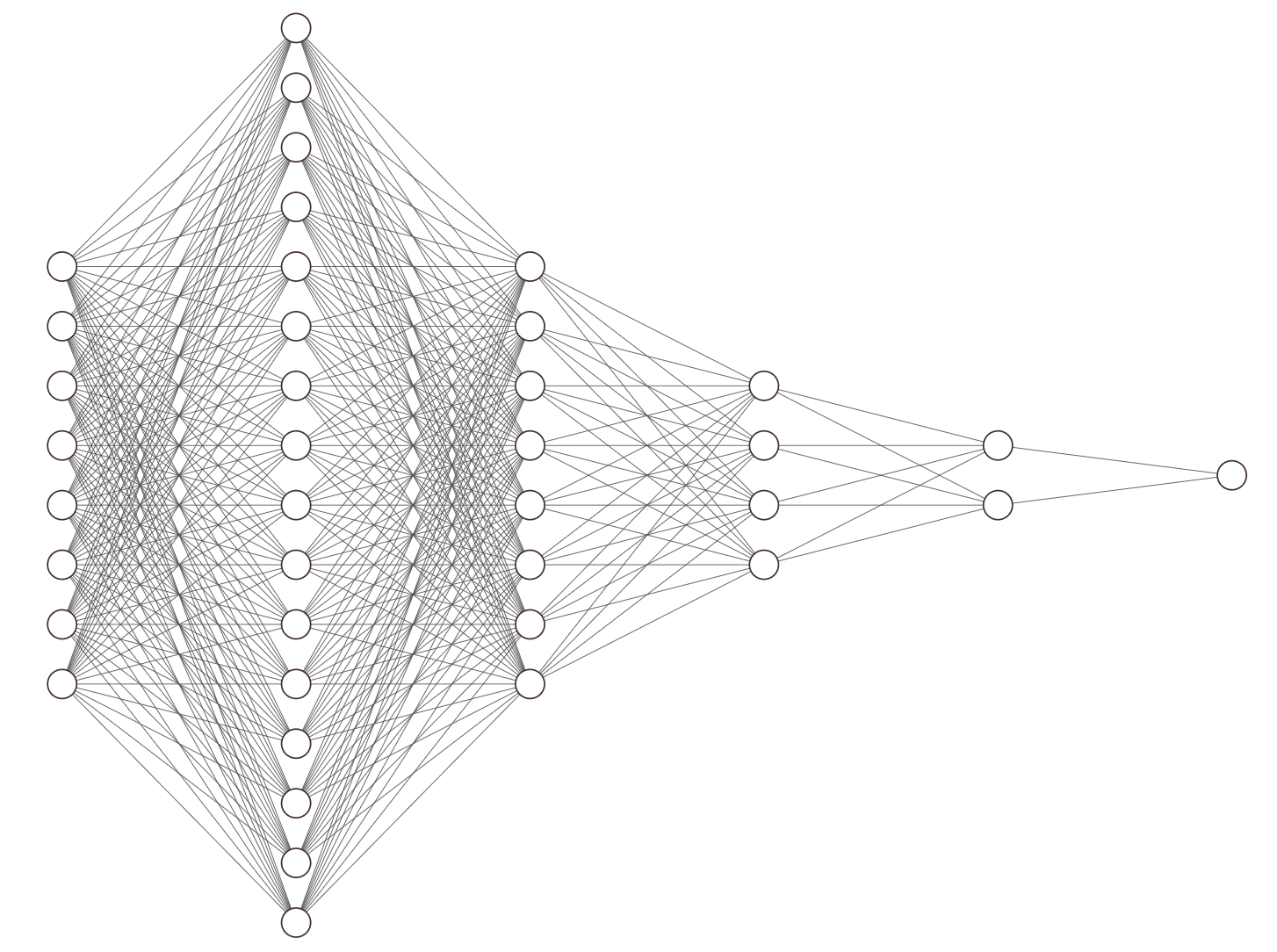
1. 数据读取

自定义dataset类依次读取数据。

考虑到原数据集的格式，使用二进制的方式读取文件，并用\_\_getitem()\_\_中的item参数决定数据集文件的字节偏移量读取指定位置的数据和label。对于图像数据，进行归一化。

1. 搭建网络结构

原数据集中的图像数据维度为28\*28，输入神经元个数即为784。搭建MLP结构如下：



（由于神经元数量过多，图中仅表示比例。）

|  |
| --- |
| self.mlp = nn.Sequential(  nn.Linear(in\_nums, in\_nums \* 2),  nn.Dropout(drop),  nn.ReLU(),  nn.Linear(in\_nums \* 2, in\_nums),  nn.Dropout(drop),  nn.ReLU(),  nn.Linear(in\_nums, in\_nums // 2),  nn.Dropout(drop),  nn.ReLU(),  nn.Linear(in\_nums // 2, in\_nums // 4),  nn.Dropout(drop),  nn.ReLU(),  nn.Linear(in\_nums // 4, out\_nums)  ) |

使用5个线性层，神经元数量分别为：784， 1568， 784， 392， 196， 1。

每层之间使用dropout防止过拟合和ReLU激活函数引入非线性因子。

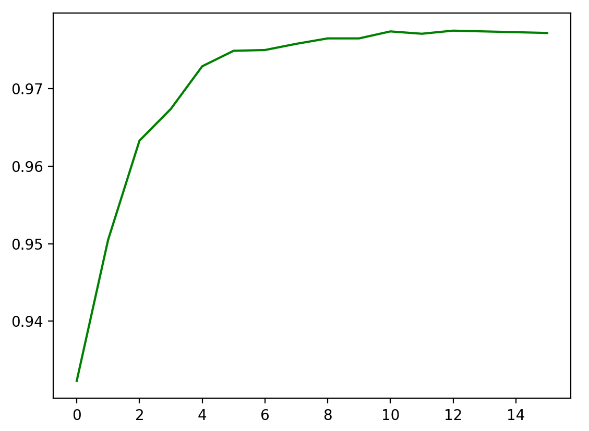
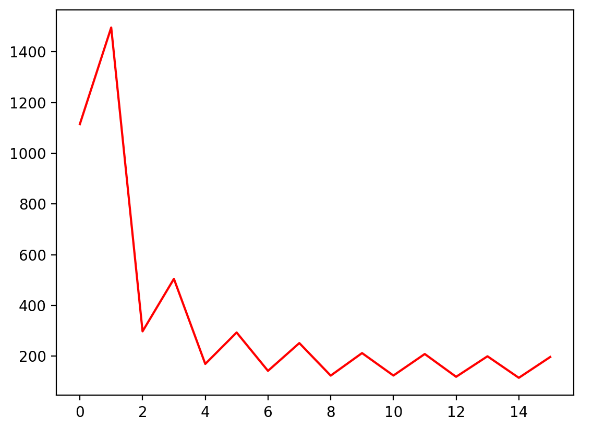
1. 定义优化器

使用Adam优化器，初始学习率设置为1e-4，weight decay设置为2e-5，学习率调整方式使用MultiStepLR，milestone设置为2，4，6，gamma为0.2，共训练epoch数为8次。

|  |
| --- |
| optimizer = optim.Adam(filter(lambda p: p.requires\_grad, model.parameters()), lr=opt.lr\_initial,  betas=(0.9, 0.999), eps=1e-8, weight\_decay=opt.weight\_decay)  scheduler = torch.optim.lr\_scheduler.MultiStepLR(optimizer, milestones=[2, 4, 6], gamma=0.2) |

1. 定义损失函数并训练

使用交叉熵损失函数进行训练。每个epoch进行两个测试并且记录。训练loss和测试准确率图如下：

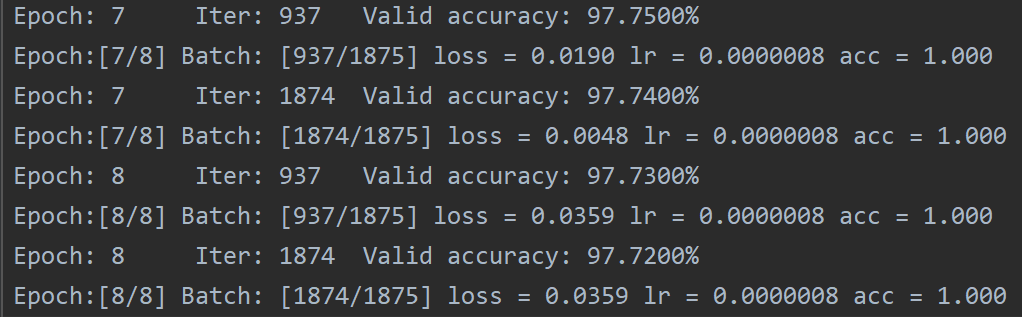


最终在验证集上的准确率为97.73%。

**实验结果与分析：**

在验证集上的准确率为97.73%。

根据训练过程中的log，发现模型在最后有一定程度上过拟合的现象，在训练集上的准确率接近100%。



通过增加dropout层或者一些数据增强，可以缓解这一现象。

加载训练参数代码：

|  |
| --- |
| net = MLP().cuda()  net.load\_state\_dict(torch.load('./checkpoint/mlp\_mnist\_e8.pth')['state\_dict']) |