

# 卷积神经网络实验报告

姓名：靳乐卿

学号：2012159

实验要求：

- 掌握前馈神经网络（FFN）的基本原理
- 学会使用 PyTorch 搭建简单的 FFN 实现 MNIST 数据集分类
- 掌握如何改进网络结构、调试参数以提升网络识别性能

报告内容：

- 运行原始版本 MLP，查看网络结构、损失和准确度曲线
- 尝试调节 MLP 的全连接层参数（深度、宽度等）、优化器参数等，以提高准确度
- 分析与总结格式不限
- 挑选 MLP-Mixer，ResMLP，Vision Permutator 中的一种进行实现（加分项）

作业提交：

- 期末前将报告和代码（可将 jupyter notebook 里代码复制到一个 xxx.py 文件中）打包（学号+姓名.zip），提交方式另行通知
- 实验报告内容应工整
- 应包含实验心得（**重点阐述部分**）：描述实验过程中对参数和网络结构进行调整时对实验结果的影响。

## 原始版本 MLP 运行

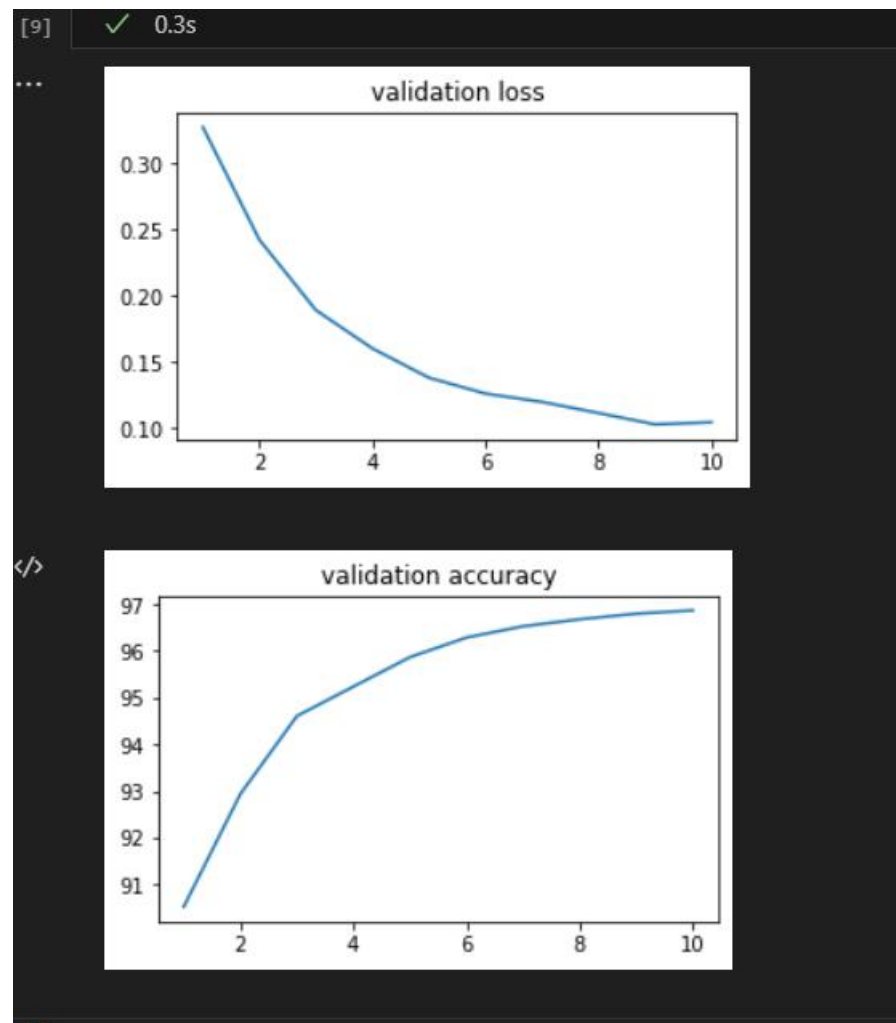
网络结构：

```
[5] ✓ 1.1s

... Net(
  (fc1): Linear(in_features=784, out_features=100, bias=True)
  (fc1_drop): Dropout(p=0.2, inplace=False)
  (fc2): Linear(in_features=100, out_features=80, bias=True)
  (fc2_drop): Dropout(p=0.2, inplace=False)
  (fc3): Linear(in_features=80, out_features=10, bias=True)
)
```

结构分析：第一层是一个全连接层，把 28\*28 的图片映射到维度为 100 的空间上，然后接下来为了防止过拟合，以 20% 的几率随机使一些神经元失效，接下来是一个 100 维到 80 维的全连接，接着仍然是一个 dropout 层，防止过拟合。最后是一个 80 维到 10 维的映射，这里映射到 0 到 9 的十个数字上

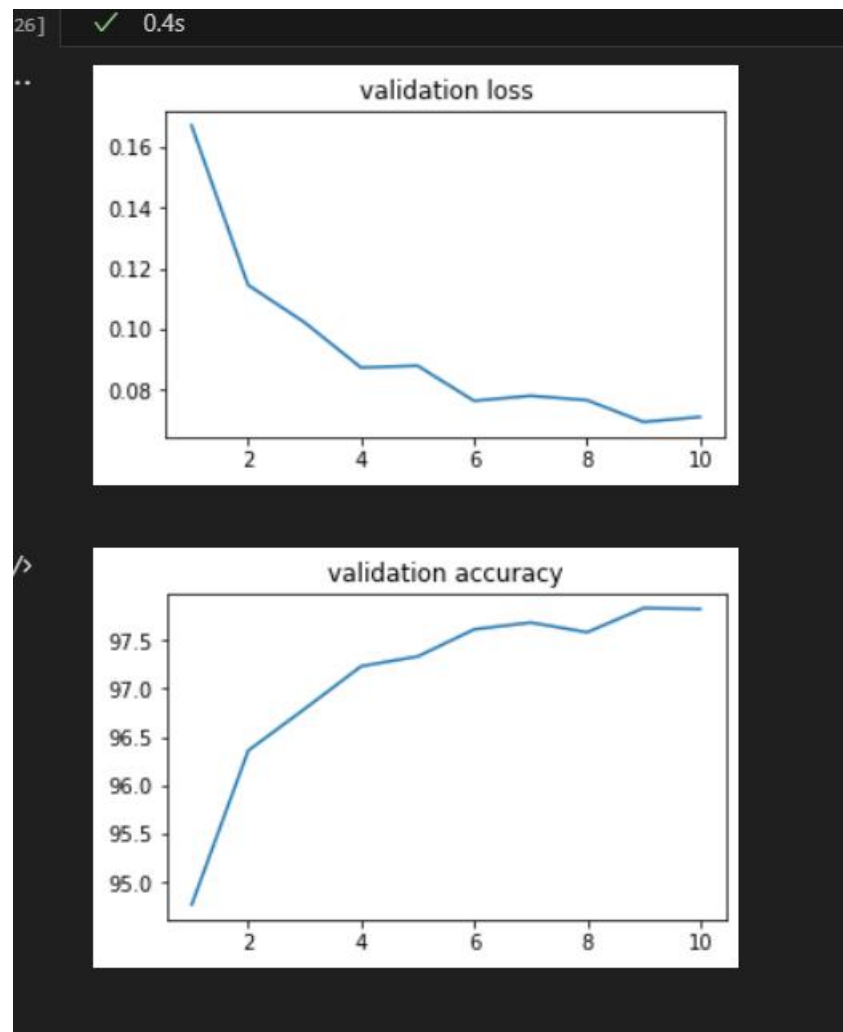
## 损失和精确度曲线



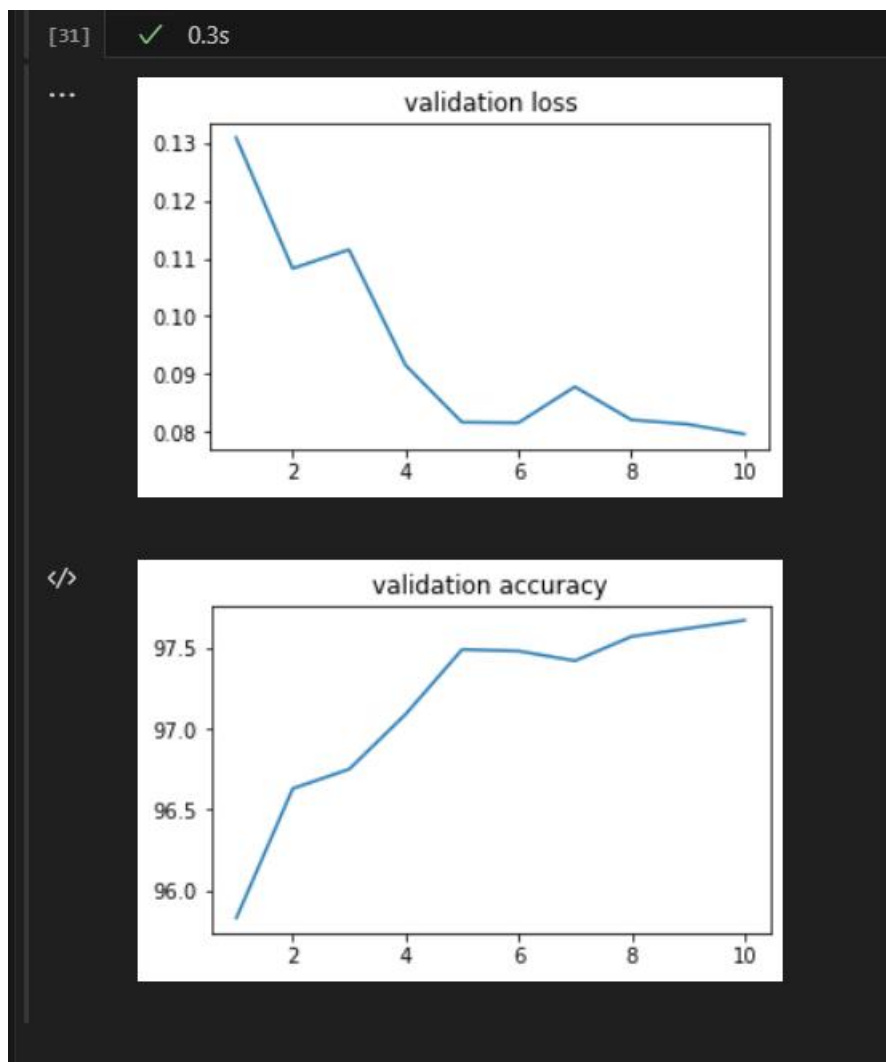
**调节全连接层、优化器参数，以提高准确度**

调整学习率

尝试提高学习率到 0.05，结果如下：



继续提高学习率到 0.1

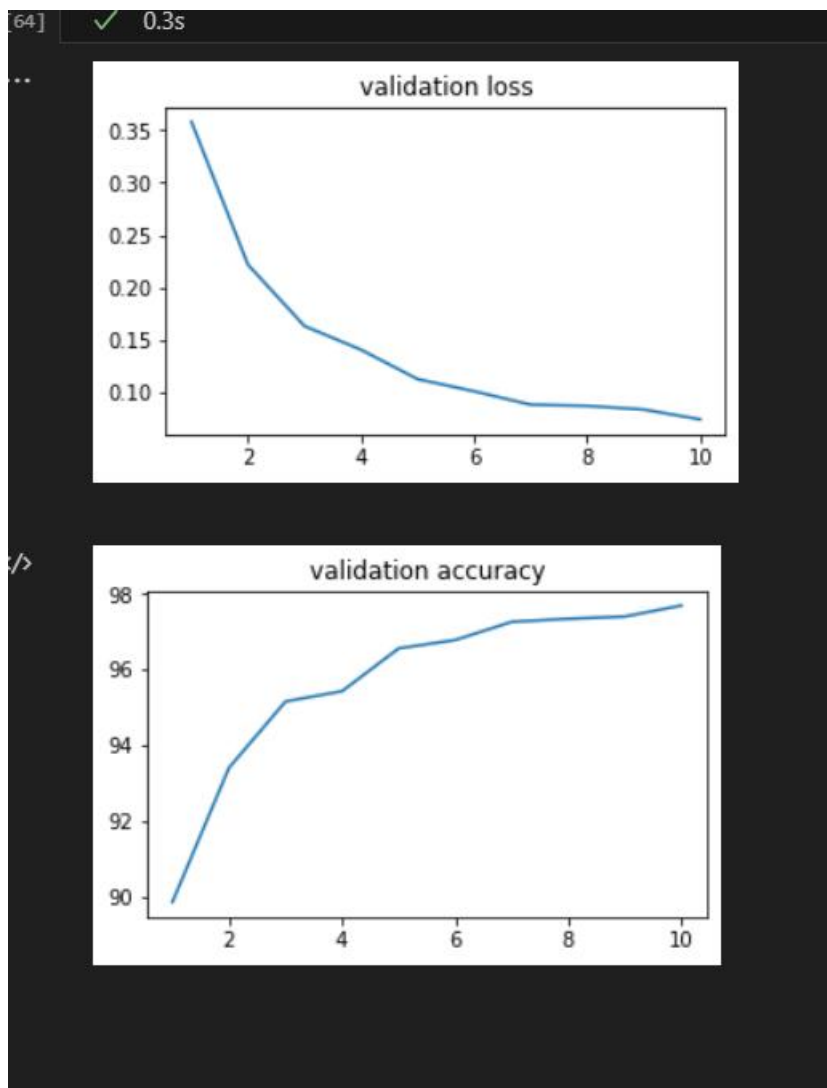


## 调整网络参数

调整网络结构如下（学习率恢复至 0.01）

```
[59] ✓ 0.0s
... Net(
  (fc1): Linear(in_features=784, out_features=392, bias=True)
  (fc1_drop): Dropout(p=0.2, inplace=False)
  (fc2): Linear(in_features=392, out_features=196, bias=True)
  (fc2_drop): Dropout(p=0.2, inplace=False)
  (fc3): Linear(in_features=196, out_features=98, bias=True)
  (fc3_drop): Dropout(p=0.2, inplace=False)
  (fc4): Linear(in_features=98, out_features=10, bias=True)
)
```

结果如下



## 分析与总结

我们可以看到随着学习率的升高，他的拟合速度变快，在训练 10 轮的情况下准确率会提升，但是同时的，其变化也更为陡峭，此外我们可以看到学习率在 0.05 的时候已经可以看到反而有训练多一轮反而准确率下降，同时 loss 上升，这代表着过大的学习率也会让学习过程中错过拟合函数的极值点，导致无法得到最优效果，所以在训练过程中，往往开始设置较大的学习率，在后面慢慢降低。

关于网络结构的调整，我将前面的维度提高，以期望可以在一开始可以多学习图片的特征，此外增加了一层额外的全连接层来增加网络的表达能力，实践表明，控制其他参数不变的情况下，确实比原始的网络效果有所提升。