

# 卷积神经网络实验报告

姓名： 黄昶玮

学号： 2112380

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）

实验内容：

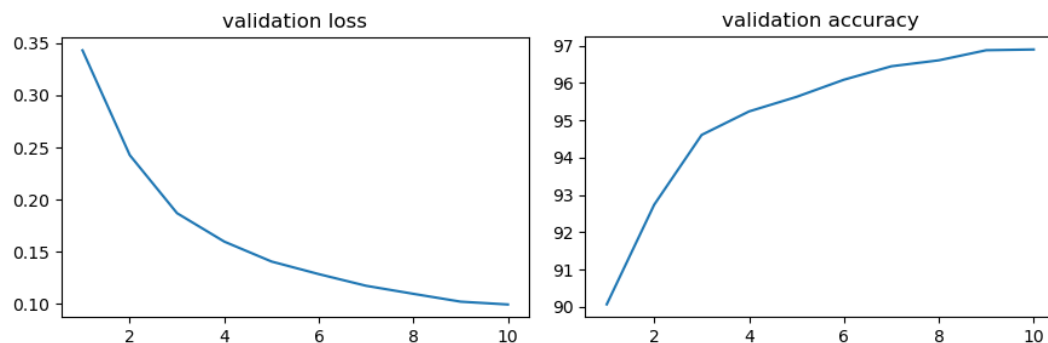
## 一．原始版本 MLP

网络结构：

```
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)  
)
```

初始的一个简单的全连接神经网络，包含了三个线性全连接层和两层 Dropout 层。第一层线性层将输入的 784 维特征向量映射到 100 维，第二层 Dropout 层在训练期间，以 20% 的概率随机将某些神经元的输出设置为零，从而防止过拟合，再之后将 100 维的输入特征向量映射到 80 维，经历了一个 dropout 层后又将 80 维的输入特征向量映射到 10 维，对应 mnist 数据集的 10 个类别

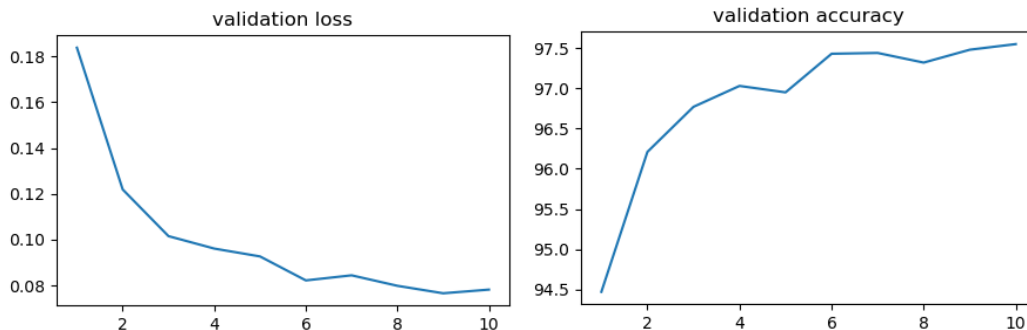
输出结果：



## 二. 网络调参

### 1. 修改学习率:

将 lr 改为 0.04



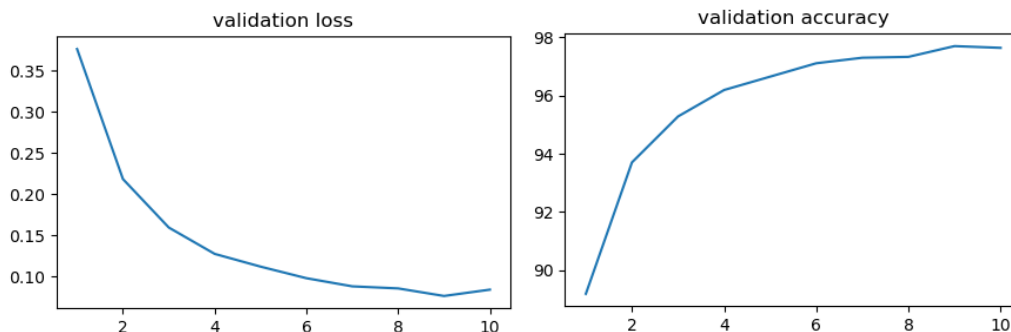
增大学习率后在初始训练过程中 loss 下降的速度明显增快但是可以看到在 6 个 epoch 后 loss 值有回升, 并没有继续下降, 同时验证集的准确率也下降。高学习率会导致梯度更新幅度过大, 从而使得权重更新时出现较大的震荡。这会导致梯度值非常大, 进而导致损失函数的值大幅波动甚至增大。但是最终还是逐渐下降。

### 2. 修改网络结构:

增加网络层数为如下

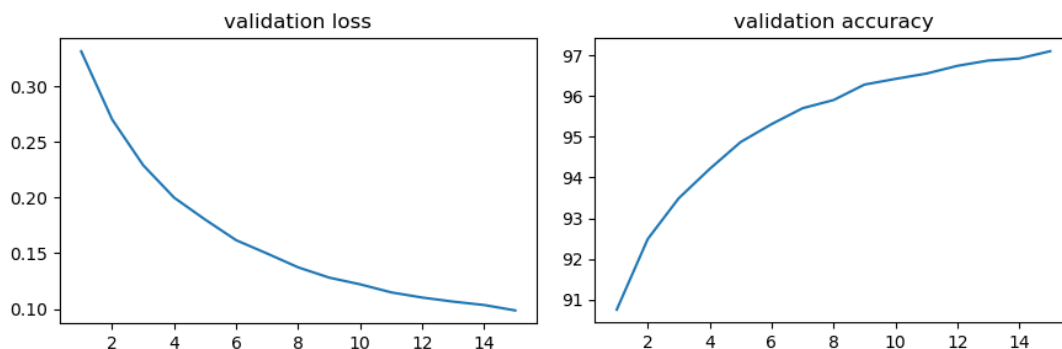
```
Net(  
  (fc1): Linear(in_features=784, out_features=400, bias=True)  
  (fc1_drop): Dropout(p=0.2, inplace=False)  
  (fc2): Linear(in_features=400, out_features=100, bias=True)  
  (fc2_drop): Dropout(p=0.2, inplace=False)  
  (fc3): Linear(in_features=100, out_features=80, bias=True)  
  (fc3_drop): Dropout(p=0.2, inplace=False)  
  (fc4): Linear(in_features=80, out_features=10, bias=True)  
)
```

结果:



减少网络层数为如下:

```
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=10, bias=True)  
)
```



可以看到在增加了网络层数的时候在训练了 9 轮之后 loss 值略有上升，有可能是因为参数过多产生了过拟合，而减少了网络层数之后，训练的 loss 值一直下降。在增加了训练轮数后依然可以达到和原网络同样的 loss 和准确率。

### 三．实验总结和心得

在此次的实验中我尝试了对学习率和网络结构的调整，得出结论：学习率的提高可以很快提高初始阶段的训练速度，但是在后期模型可能无法收敛到一个稳定的状态，而是不断在最优解周围振荡。实际训练过程中可以采取逐渐递减调整的学习率策略。同时网络的结构直接决定了网络的表达能力，过于复杂的网络可能会造成过拟合的结果，对于 mnist 这种简单数据集，使用最简单的一层线性层通过增加训练轮数也可以达到很好的训练结果。