



南開大學  
Nankai University

计算机学院  
深度学习实验报告

## MLP 实验报告

姓名：徐文斌

学号：2010234

专业：计算机科学与技术

2023 年 6 月 18 日

# 目录

<b>1</b>	<b>原始版本 MLP</b>	<b>2</b>
1.1	网络结构 .....	2
1.2	损失和准确度曲线 .....	2
<b>2</b>	<b>网络结构改进</b>	<b>3</b>
2.1	增加网络深度 .....	3
2.2	增加网络宽度 .....	4
2.3	调节训练参数 .....	5
2.4	综合调整模型 .....	6
<b>3</b>	<b>ViP 网络复现</b>	<b>6</b>

# 1 原始版本 MLP

## 1.1 网络结构

使用 print 函数打印网络结构如下所示：

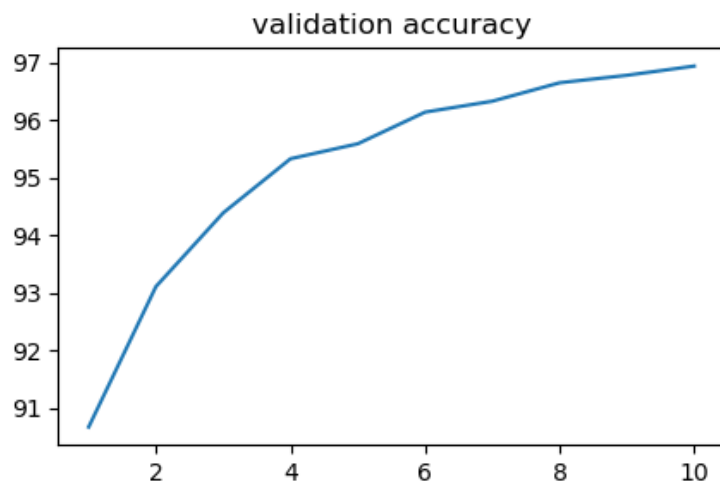
```
1 Net(  
2   (fc1): Linear(in_features=784, out_features=100, bias=True)  
3   (fc1_drop): Dropout(p=0.2, inplace=False)  
4   (fc2): Linear(in_features=100, out_features=80, bias=True)  
5   (fc2_drop): Dropout(p=0.2, inplace=False)  
6   (fc3): Linear(in_features=80, out_features=10, bias=True)  
7 )
```

第一层是一个全连接层，将  $28 \times 28$  的图片映射到 100 维。为了防止过拟合，在训练过程中使用 dropout 随机忽略 20% 的神经元。接下来一个输入 100 维输出 80 维的全连接层，同样使用 dropout 随机忽略 20% 的神经元；最后再通过一个 80 维到 10 维的全连接层，该层输出的结果经过 softmax 便可以预测出图像的数字。

## 1.2 损失和准确度曲线

训练网络，得到每轮的损失和准确度曲线如下，在经过 10 轮训练后，在验证集上达到了 97.13% 的准确率。





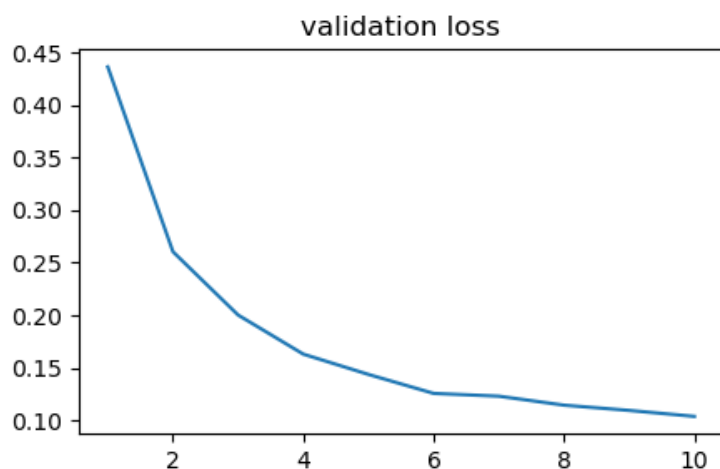
## 2 网络结构改进

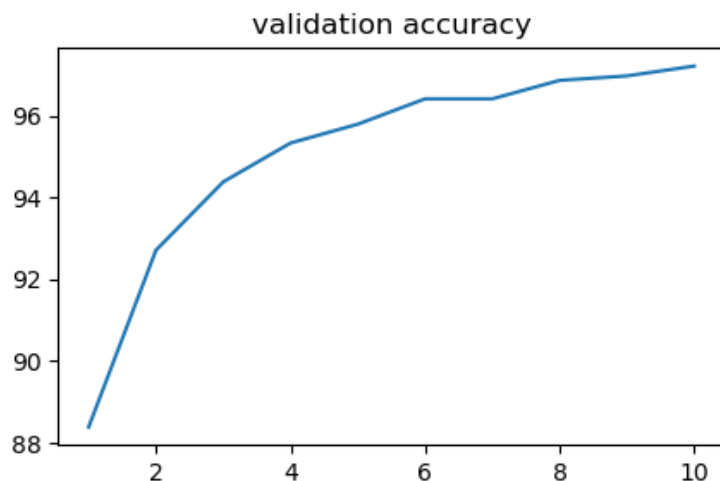
### 2.1 增加网络深度

首先我们尝试增加网络深度，网络结构如下，我们在原始网络的基础上额外增加了一个全连接层。

```
1 Net(  
2     (fc1): Linear(in_features=784, out_features=100, bias=True)  
3     (fc1_drop): Dropout(p=0.2, inplace=False)  
4     (fc2): Linear(in_features=100, out_features=50, bias=True)  
5     (fc2_drop): Dropout(p=0.2, inplace=False)  
6     (fc3): Linear(in_features=50, out_features=25, bias=True)  
7     (fc3_drop): Dropout(p=0.2, inplace=False)  
8     (fc4): Linear(in_features=25, out_features=10, bias=True)  
9 )
```

训练网络，得到每轮的损失和准确度曲线如下，在经过 10 轮训练后，在验证集上达到了 97.21% 的准确率。



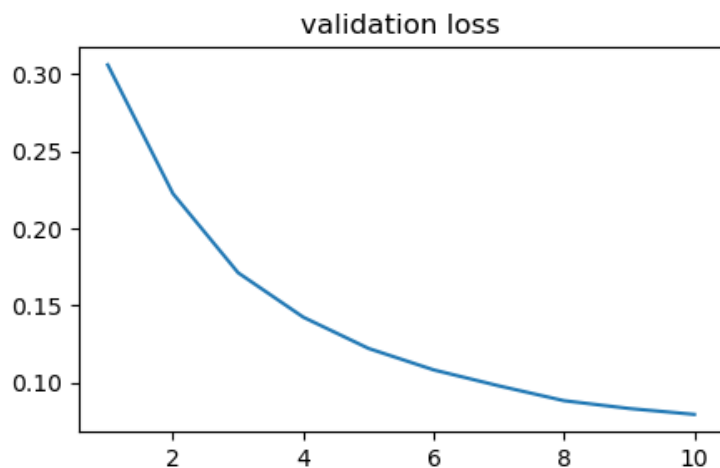


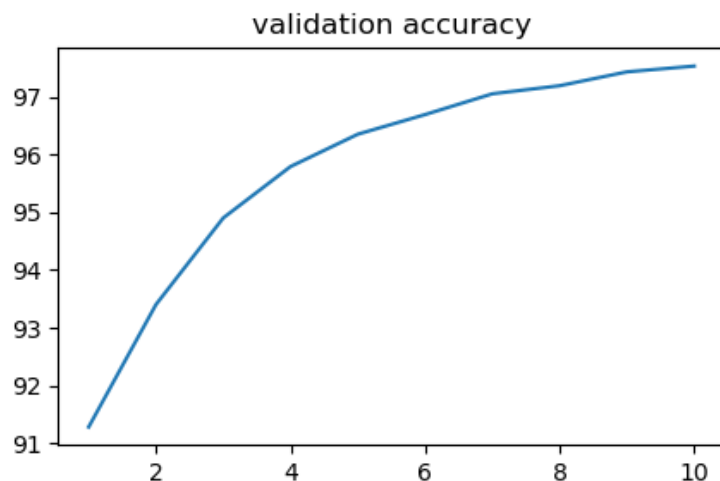
## 2.2 增加网络宽度

接下来我们尝试增加网络宽度，保持原始网络深度不变，我们增加原始网络全连接层的输出宽度。网络结构如下。

```
1 Net(  
2     (fc1): Linear(in_features=784, out_features=512, bias=True)  
3     (fc1_drop): Dropout(p=0.2, inplace=False)  
4     (fc2): Linear(in_features=512, out_features=256, bias=True)  
5     (fc2_drop): Dropout(p=0.2, inplace=False)  
6     (fc3): Linear(in_features=256, out_features=10, bias=True)  
7 )
```

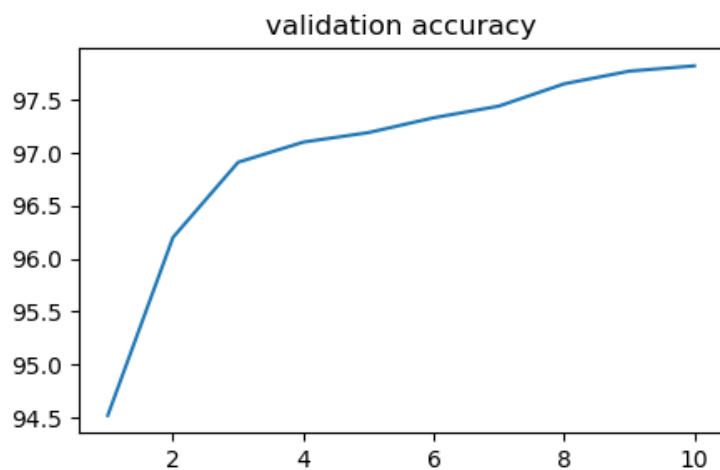
训练网络，得到每轮的损失和准确度曲线如下，在经过 10 轮训练后，在验证集上达到了 97.53% 的准确率。





### 2.3 调节训练参数

这里我们保持原始网络的结构不变，主要调节模型训练时的学习率。最终发现当学习率为 0.04 时，经过 10 轮的训练之后，模型在验证集上达到了 97.82% 的准确率。损失和准确度曲线如下。



## 2.4 综合调整模型

我们综合结合上述提到的方法来对网络进行优化，同时增加网络的宽度和深度，模型如下所示。

```

1 Net(
2   (fc1): Linear(in_features=784, out_features=512, bias=True)
3   (fc1_drop): Dropout(p=0.2, inplace=False)
4   (fc2): Linear(in_features=512, out_features=256, bias=True)
5   (fc2_drop): Dropout(p=0.2, inplace=False)
6   (fc3): Linear(in_features=256, out_features=128, bias=True)
7   (fc3_drop): Dropout(p=0.2, inplace=False)
8   (fc4): Linear(in_features=128, out_features=64, bias=True)
9   (fc4_drop): Dropout(p=0.2, inplace=False)
10  (fc5): Linear(in_features=64, out_features=10, bias=True)
11 )

```

接下来，调整学习率为 0.04 训练模型，经过 10 轮的训练之后，得到了 97.99% 的准确率。

所做的改进尝试以及模型准确率如下：

改进方式	准确率
原始网络	97.13%
增加深度	97.21%
增加宽度	97.53%
调节学习率	97.82%
综合改进	97.99%

## 3 ViP 网络复现

ViP 是一种新型 MLP 架构，此前的 MLP 模型沿展平的空间维度对空间信息进行编码。而 Vision Permutator 则使用线性投影分别沿高度和宽度维度（线形感受野）对特征表示进行编码。其主要创新点在于提出了一种 Permute-MLP 结构，该网络结构将特征在通道方向上直接采用正常的线性全连接层进行处理，在宽度和高度方向则先对特征进行了重新排列与整合，然后通过全连接层处理，最后将三个分支的结果进行加权求和。

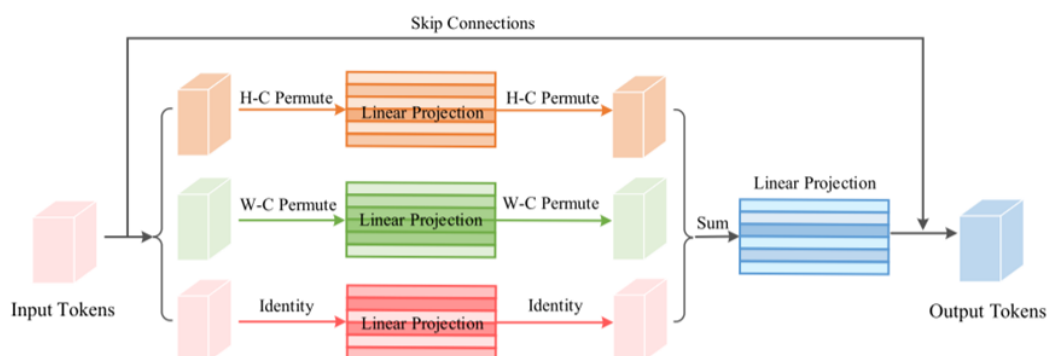


图 3.1: Hou Q, Jiang Z, Yuan L, et al. Vision permutator: A permutable mlp-like architecture for visual recognition, 2022

根据论文对 ViP 网络进行实现，运行得到损失和准确度曲线如下图所示。模型在验证集上达到了

98.09% 的准确率。

