

# 人工神经网络 HW2 Cifar-10 Classification with MLP and CNN

冯卓尔

计 86 2017011998

fengzhuoer-thu@outlook.com

**摘要：**本文主要记录了 Cifar-10 Classification with MLP and CNN 作业中的一些实验，测试了 MLP 与 CNN 激活函数在本任务中的优劣，BN 层与 Drop-out 层在相应网络中的性能，以及测试了各项参数对任务的准确率与 loss 中的影响。

**关键字：**多层感知机（MLP） 卷积神经网络（CNN） 数字图像识别 神经网络

## 一、网络基本参数介绍

### 1.1 MLP 全连接网络参数

层	1	2	3	4	5
类型	Linear	BN	ReLU	Dropout	Linear
输入节点数	784	256	256	256	256
输出节点数	256	256	256	256	10

Learning rate = 0.001

Decay=0.9995

### 1.2 CNN 网络参数

编号	类型	输入	输出	kernel (卷积、池化)	Dropout
1	Conv2d	Self.x_(32*32*3)	Conv1	40*8	
2	bn	Conv1	Bn1		
3	Relu	Bn1	Relu1		
4	Dropout	Relu1	Dp1		0.5
5	Maxpool2d	Dp1	Pool1	2*2	
6	Conv2d	Pool1	Conv2	40*8	
7	Bn	Conv2	Bn2		
8	Relu	Bn2	Relu2		
9	Dropout	Relu2	Dp2		0.5

10	Maxpool2d	Dp2	Pool2	2*2	
11	linear	Pool2.reshape(-1, 160)	10		

Learning rate = 0.001

Decay=0.9995

### 1.3 运行环境

Google Cloud n1-standard-4 Instance (4 vCPU, 15 GB RAM) with 1 NVIDIA Tesla P100 GPU

## 二、实验结果与讨论

### 2.1 如何填写网络参数设置

对于训练过程，`in_train=True`，`resue = False`；对于 validation 过程，由于需要用到与训练过程的同一组参数，因此不需要进行再次训练改变参数，`in_train=False`，`resue = True`。

在 BN 层与 drop out 层中，封装的一些函数在 `is_train` 与否的状态下表现不同，因而每次进行 forward 时需要知道当前网络的 `is_train` 状态，即每次传参都需要有 `is_train`，`is_train` 来控制 BN 层与 drop out 层的行为。

在网络的 `__init__` 函数中，forward 会被调用两次。如果不使用 `variable_scope` 功能，那么通过 `self.x_` 会建立两张绝缘的相同网络，它们的参数不共享（但是事实上两次作用的参数应当共享）。`Reuse=true` 时第二次调用 forward 时 model 就不需要重新建立一个新的网络，而是使用一个名为 'model' 的已建立网络中的参数进行第二次调用。如果 `reuse=false`，同时建立两个重名为 'model' 的网络会引发报错。

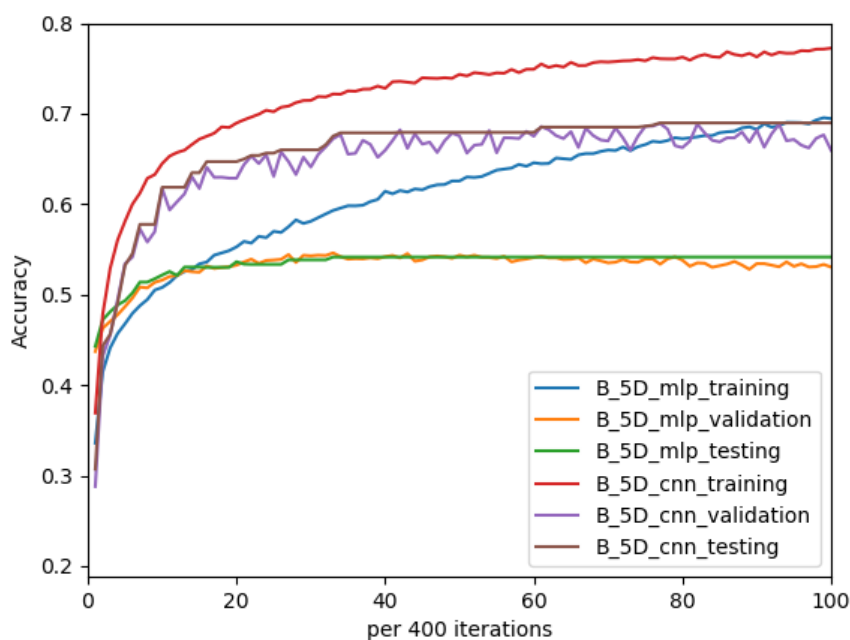
之所以 train 与 test 需要分开执行，是因为在 train 过程中，BN 层与 Drop-out 是为了降低网络对训练集的依赖而做的防止过拟合的措施，而测试时并不需要，因而可以通过 `is_train` 来控制网络训练的逻辑，使训练的时候 BN 与 drop-out 层发挥应有的作用，而测试的时候不发挥。

### 2.2 training loss/acc 与 validation loss /acc 的绘制

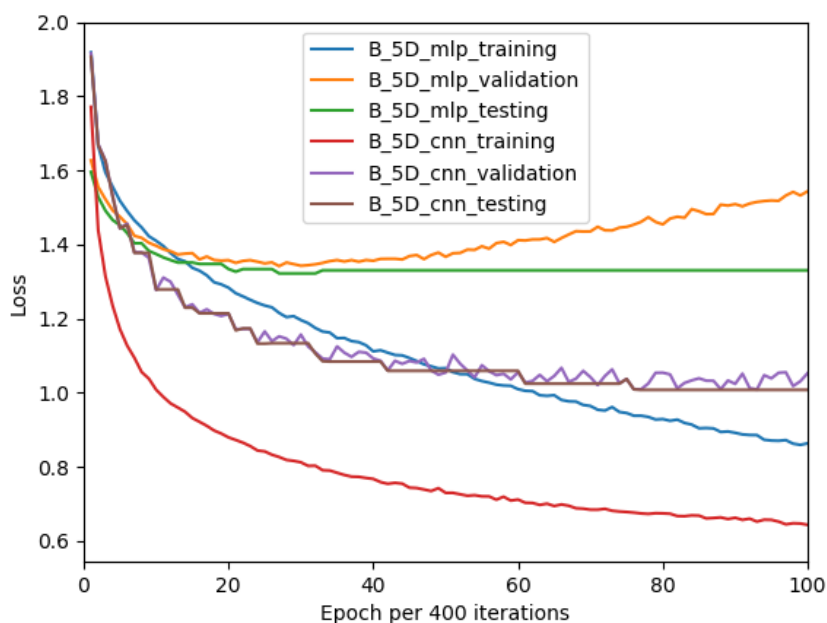
从图中我们可以观察到，两个模型的曲线出现以下特征。

随着迭代的增加，training acc 不断增加，loss 不断减少；但是 validation 和 testing acc 到一定时候会停止增长甚至会减少，loss 到一定时候会停止增长甚至会增加

（B\_5D\_mlp\_validation loss）。出现这种现象的原因应该是过拟合。在这个任务中，由于 mlp 的感知能力有限，在感知范围内的训练完成后会过度，对于未感知到的特征会忽略，因而 mlp 模型过拟合现象更加明显。使过拟合现象更加明显的做法可以将 dropout-rate 调整至接近 0 的小数，反之增大 dropout rate (<1) 能够缓解过拟合现象。



图一 MLP/CNN 网络 training, validation, testing 准确率



图二 MLP/CNN 网络 training, validation, testing loss

### 2.3 MLP 与 CNN 模型在处理 cifa-10 问题上的区别

在 2.2 中的图线中可以看出，CNN 的 testing acc 显著比 MLP 的高，CNN 的 testing loss 比 MLP 的低。

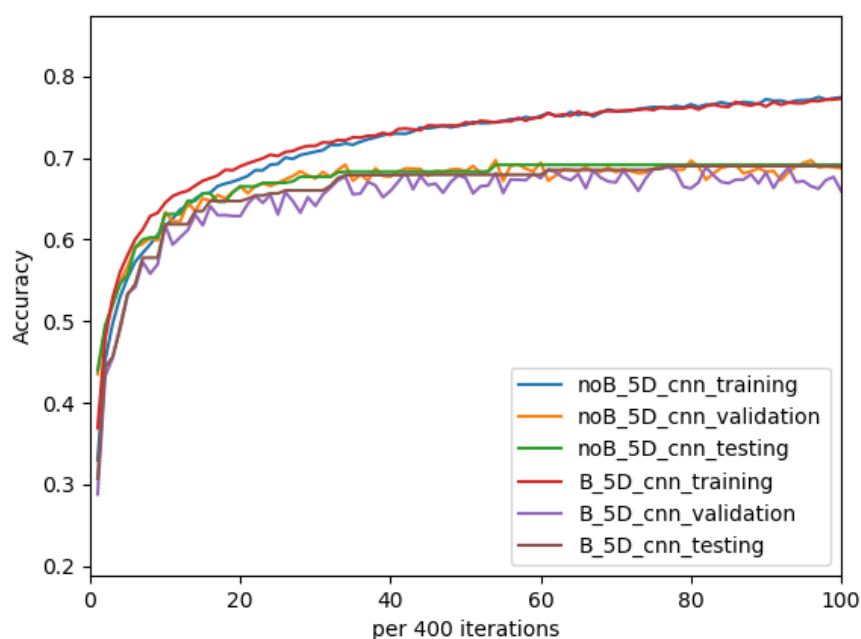
另外，MLP 的 validation loss 较 training loss 显著要高，在 testing loss 停止减小后，MLP 网络的 training loss 仍然在减小，但是 validation loss 反而变大了（40 个 epoch 之后），过拟合现象比较严重；而同时期的 CNN 网络中并未发现类似的过拟合现象。

网络	MLP	CNN
Training_acc	0.6946	<b>0.7725</b>
Training_loss	0.8632	<b>0.6434</b>
Validation_acc	0.5305	<b>0.6596</b>
Validation_loss	1.5434	<b>1.0526</b>
Testing_acc	0.5416	<b>0.6899</b>
Testing_loss	1.3302	<b>1.0077</b>

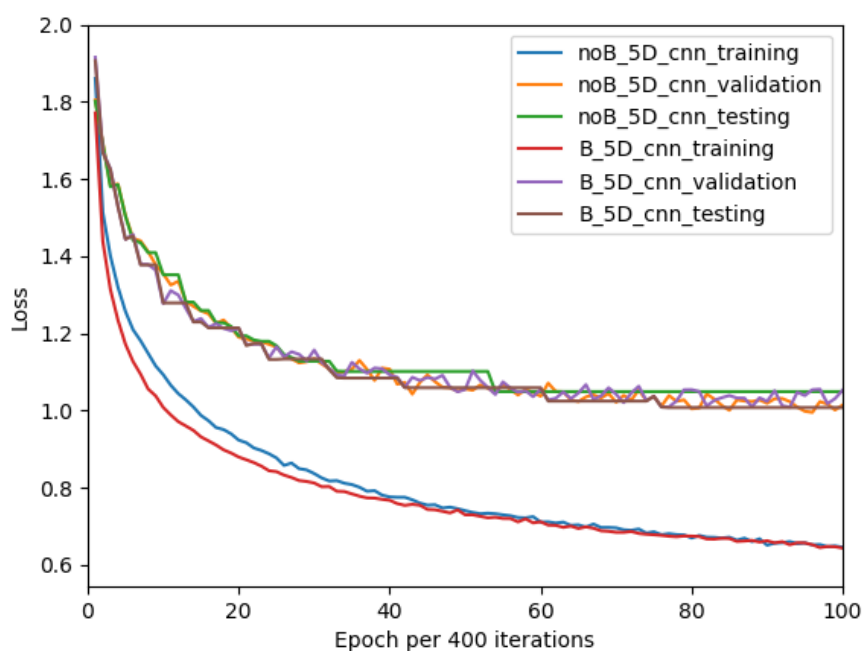
## 2.4 BN 层对 MLP 与 CNN 模型在处理 Cifar-10 问题上的性能影响

为了研究这个问题，我删去了 CNN 中的两个 BN 层，MLP 中的唯一 BN 层，并且将 drop rate 调节至 CNN 网络性能最优的 0.5，进行了实验。将其结果与带有 BN 层的结果一起绘制于下图中。

**CNN:** 可以看出，从收敛后的结果上看，BN 层似乎不能明显提升 CNN 网络的 testing acc，但是在 loss 曲线中，BN 层的存在下 CNN 网络的收敛速度提升了一点点。在得到近乎相同的结果前提下，收敛速度的提升对于一个网络的性能也有很大的提升。



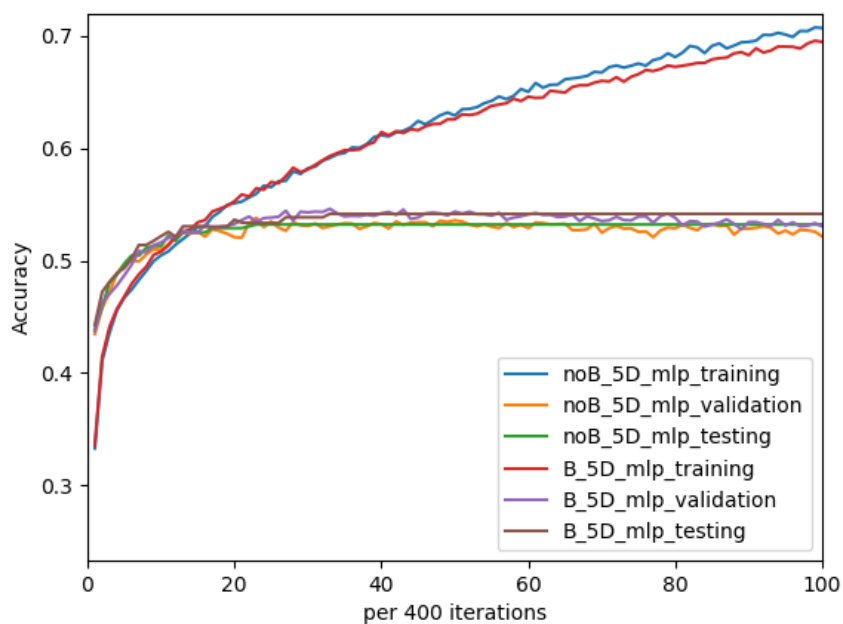
图三 CNN 网络有无 BN 层的 training, validation, testing acc 对比



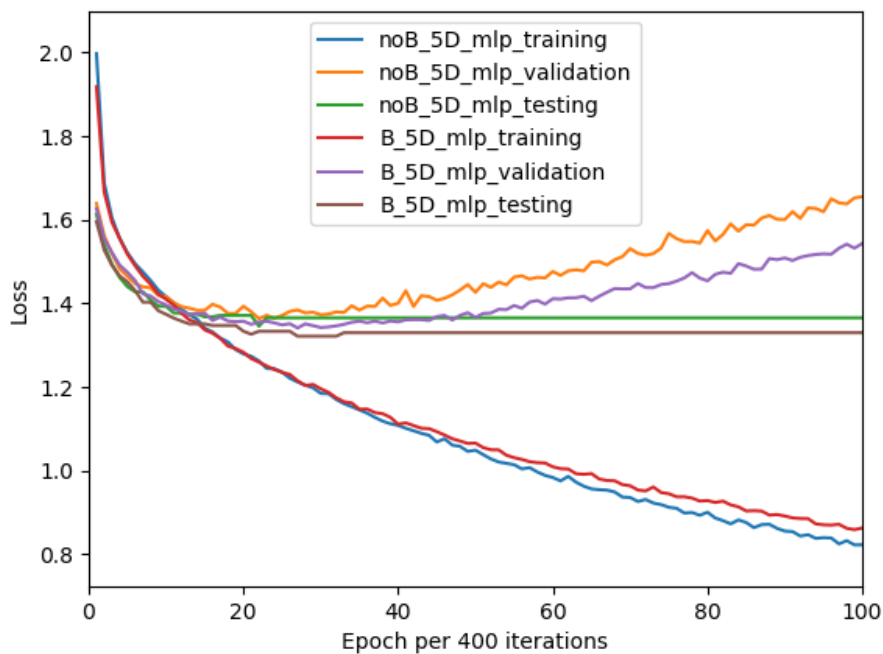
图四 CNN 网络有无 BN 层的 training, validation, testing loss 对比

网络	CNN+BN	CNN
Training_acc	<b>0.7725</b>	0.7704
Training_loss	<b>0.6434</b>	0.6459
Validation_acc	0.6596	<b>0.6873</b>
Validation_loss	1.0526	<b>1.0140</b>
Testing_acc	0.6899	<b>0.6915</b>
Testing_loss	<b>1.0077</b>	1.0490

**MLP:** 与 CNN 类似, BN 层没有显著对模型的 acc、loss 结果有所影响, 但是 BN 层加快了网络的收敛速度。与 CNN 不同的是, 由于 MLP 模型的过拟合现象比较严重, BN 层的存在还一定程度上减缓了过拟合趋势 (两者的 validation loss 曲线)。



图五 MLP网络有无BN层的 training, validation, testing acc 对比



图六 MLP网络有无BN层的 training, validation, testing loss 对比

网络	MLP+BN	MLP
Training_acc	0.6946	<b>0.7069</b>

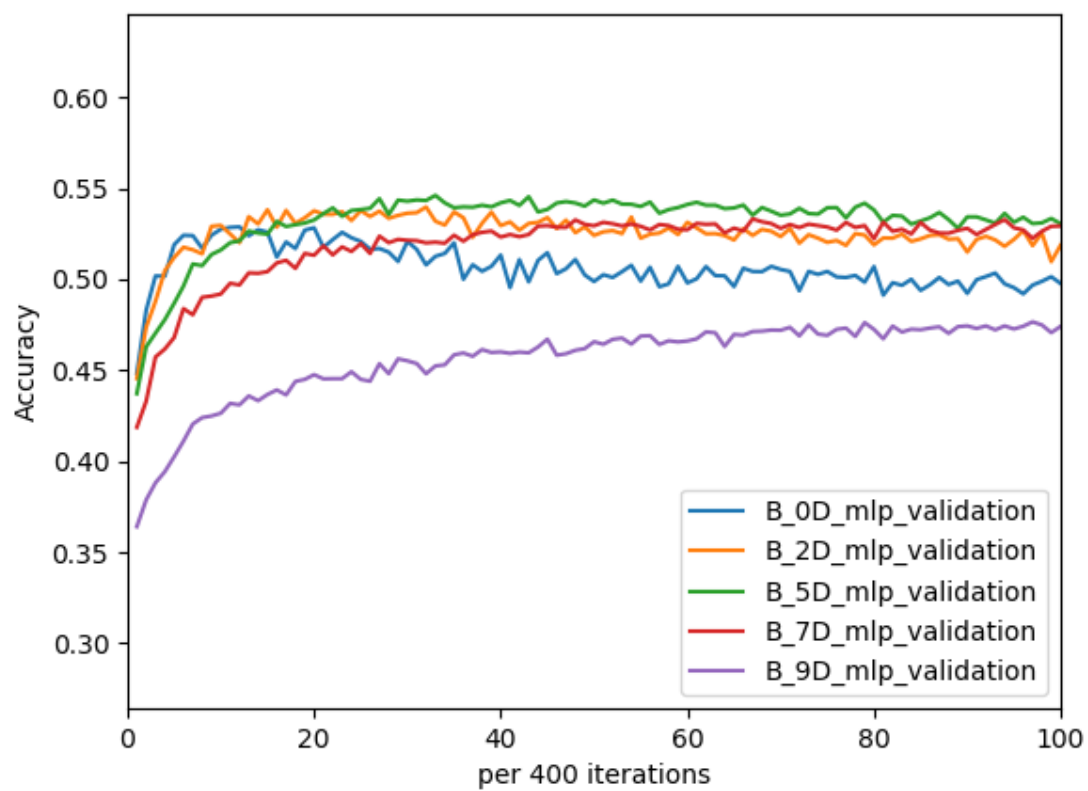
Training_loss	0.8632	<b>0.8229</b>
Validation_acc	<b>0.5305</b>	0.5216
Validation_loss	<b>1.5434</b>	1.6552
Testing_acc	<b>0.5416</b>	0.5321
Testing_loss	<b>1.3302</b>	1.3657

但是 10 月 22 日 lecture 中博士学长讲述，BN 层目前在 CNN 中的作用还是一个未知数，它的理论预测作用似乎与实际作用并不相符。具体 BN 在网络中的作用除了上述可能，其余有待商榷。

## 2.5 droping-rate 层对 MLP 与 CNN 模型在处理 cifar-10 问题上的性能影响

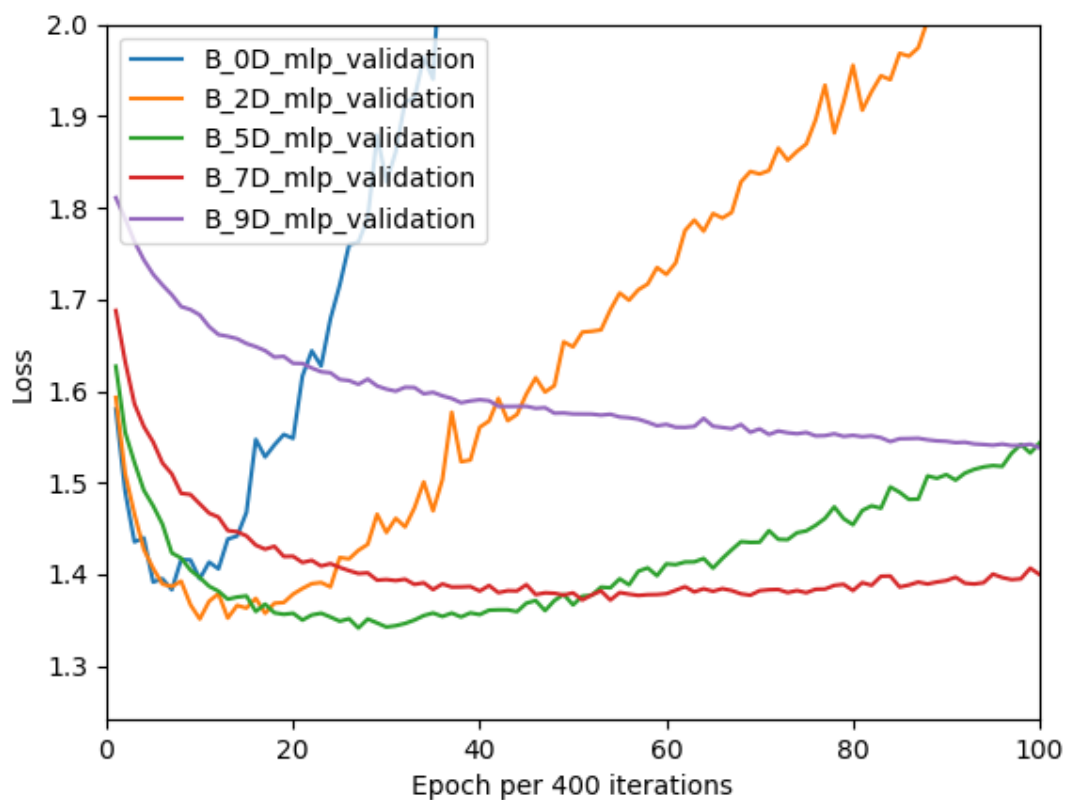
为了研究这个问题，我设置了五组 drop-rate 值，分别是{0, 0.2, 0.5, 0.7, 0.9}，分别记为 0D, 2D, 5D, 7D, 9D。其中，0D 是不加 drop-out 层的结果。对于 MLP 模型与 CNN 模型（均含有 BN），我分别进行了实验，获得了以下的 loss 与 acc 曲线。

**MLP:** 从图中可以看出，当 drop-rate 大致在 0.5 左右的时候，MLP 网络的 validation acc 能够达到比较高的程度，同时它的 loss 也是同类中相比最低的。如果 drop-rate 过大，MLP 的感知能力会有明显受阻，acc 显著减低，loss 显著变大；如果不添加 drop-rate，过拟合会限制模型的性能提升。其原因或许是由于我们建立的单隐藏层 MLP 的感知能力有限，比较容易出现过拟合问题，而 drop-out 层能够通过减低 MLP 的感知“强度”，即削弱 mlp 对于线性全连接特征的感知，来避免过拟合，从而提高测试的 acc 与降低 loss。



图七 MLP网络不同 drop rate 条件下 validation acc 对比

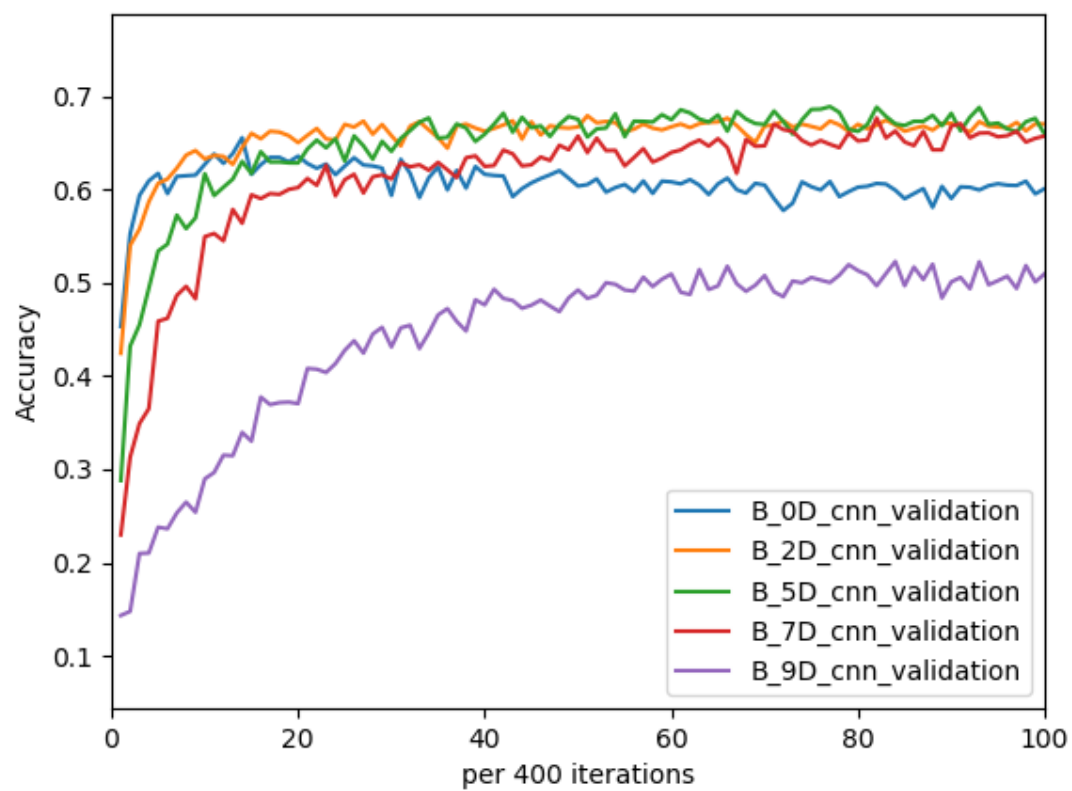




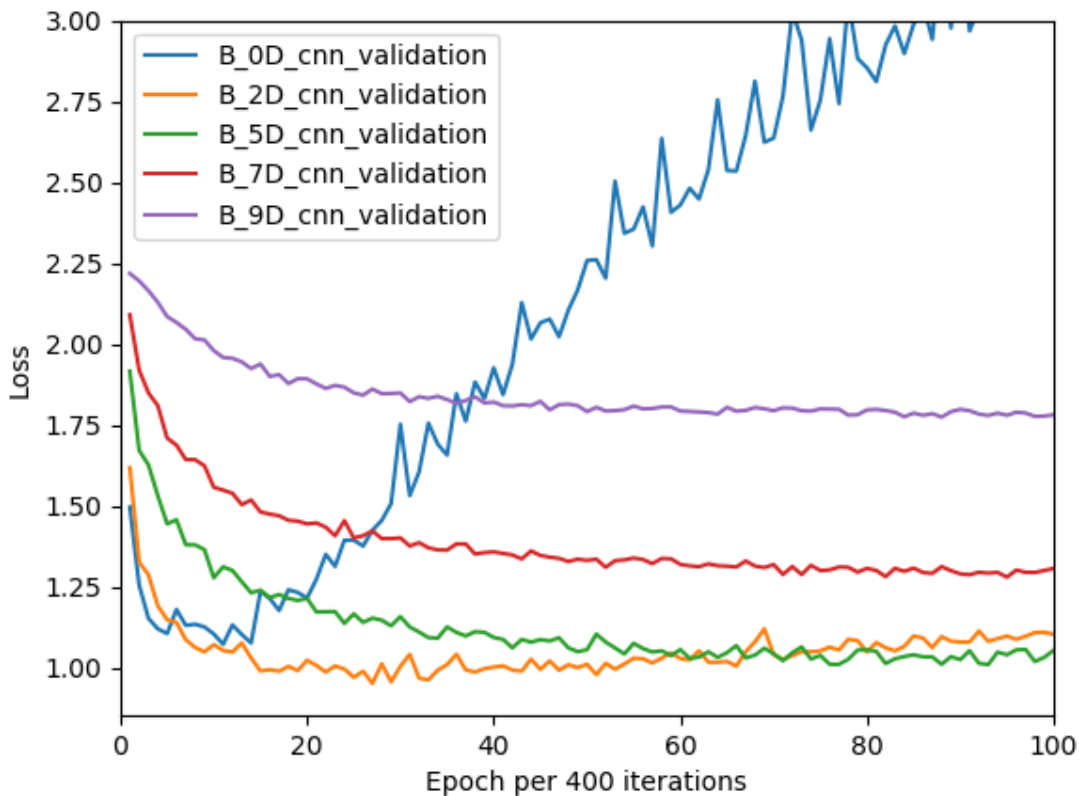
图八 MLP 网络不同 drop rate 条件下 validation loss 对比

MLP 网络 drop rate	0	0.2	0.5	0.7	0.9
Training_acc	<b>0.9577</b>	0.8630	0.6946	0.5604	0.3944
Training_loss	<b>0.1264</b>	0.4004	0.8632	1.2275	1.6849
Validation_acc	0.4977	0.5187	<b>0.5305</b>	0.5293	0.4741
Validation_loss	3.2886	2.0836	1.5434	<b>1.3999</b>	1.5381
Testing_acc	0.5282	0.5402	<b>0.5416</b>	0.5333	0.4787
Testing_loss	1.3852	1.4320	<b>1.3302</b>	1.3580	1.5198

**CNN:** 对于 CNN 模型，其感知能力远强于 MLP 的感知能力，本身 CNN 出现过拟合的线性在较低的 drop rate 下就能够被消除，倘若 drop-rate 如果过大（如 0.9），CNN 丢失了许多感知能力，降低了其本身的性能，表现 CNN 的 validation acc 降低，loss 升高。与 MLP 类似，在 drop-rate 约等于 0.5 左右时，CNN 的网络性能达到最优。



图九 CNN 网络不同 drop rate 条件下 validation acc 对比



图十 CNN 网络不同 drop rate 条件下 validation loss 对比

CNN 网络 drop rate	0	0.2	0.5	0.7	0.9
Training_acc	<b>0.9821</b>	0.8768	0.7725	0.6964	0.5428
Training_loss	<b>0.0535</b>	0.3396	0.6434	0.8656	1.2836
Validation_acc	0.6011	<b>0.6704</b>	0.6596	0.6576	0.5097
Validation_loss	3.2527	1.1021	<b>1.0526</b>	1.3055	1.7805
Testing_acc	0.6437	0.6791	<b>0.6899</b>	0.6652	0.5241
Testing_loss	1.1039	<b>0.9695</b>	1.0077	1.2848	1.7724

## 2.6 training loss 与 validation loss 的区别原因

现象是 training loss 远低于 validation loss。原因是过拟合现象的存在，如 2.2 中讨论的，无论是感知能力特别有限的 MLP 还是 CNN，在 cifar-10 任务中其能够感知到的 sample 的特征是有限的，在训练中，训练得到的结果对于其感知到的特征是最优的，但是对于未感知到的特征，拟合得越好感知性越差。即便是用同样的 sample 去测试，模型的 training loss 也会低于 validation loss。另外，2.1 中也讨论了为何 training 与 test 需要分别进行，分别进行时因为 BN 与 drop-out 的性能不同，得出来的 loss 自然不

同：在 training 时，数据集被进行了归一化与 drop，在 testing 时并无如此操作，得出来的 loss 是不一样的。

调大 drop-out 值（0.5 左右为宜）能够帮助缓解过拟合现象。

## 2.7 CNN 卷积层、池化层对性能的影响

此部分实验在 GPU 上运行，但是由于没有及时下载数据，没法获得图像与结果，只能定性阐述。

在我的实验中，卷积层的 stride (x, y) 的第二个参数 y 有 2, 4, 8 等选择，以 2、4 为最佳；第一个参数 x 有 10, 20, 40, 80 等选择，实验结果中以 40 为最佳。卷积层反映的是 CNN 的感知能力，一个维度尽可能大（如 40）能够获取到二维的信息最多。

池化层有 (1, 1) (2, 2) (4, 4) 等多种选择，在实验中以 (1, 1) 为最佳，此时事实上没有池化的模糊操作。为了体现池化的意义，我选择了 (2, 2) 将 4 个像素取平均到一个值，加快了运算效率。