

Cifar-10 Classification with MLP and CNN

袁无为 计预0

Abstract

这次作业中，我使用了 MLP 和 CNN 两种模型进行了图像分类的任务，探讨了 Dropout 和 Batch Normalization 两种方法的效果，并在 CIFAR-10 数据集上进行了一些测试，其中 MLP 模型能达到 57% 的正确率，CNN 的正确率能达到 77%。

1. Introduction

在这次作业中，我在测试了 MLP 模型和 CNN 模型在图像分类任务上的表现，并在基础的模型上添加了两方法：Dropout 和 Batch Normalization。其中 Dropout 能够减轻过拟合的程度，BN 能够提高正确率和减少过拟合的程度。

2. Approach

2.1 Basic Structure

MLP 模型的结构为 `input - Linear - BN - ReLU - Dropout - Linear - Softmax - CrossEntropyLoss`。其中隐藏层的节点数为 1000。

CNN 模型的结构为 `input - Conv - BN - ReLU - Dropout - MaxPool - Conv - BN - ReLU - Dropout - MaxPool - Linear - Softmax - CrossEntropyLoss`。其中第一个卷积层的输出的通道数为 64，第二个卷积层的输出的通道数为 128，卷积核大小都为 3×3 ，卷积为 same 卷积。MaxPooling 层的 PoolingSize 为 2。

在训练时，往 `model.forward` 中传入的参数为 `is_train = True, reuse = tf.AUTO_REUSE`。在测试时，传入的参数为 `is_train = False, reuse = tf.AUTO_REUSE`。当 `is_train == True` 时，BN 层会根据当前 MiniBatch 的输出计算平均值和方差，Dropout 层会随机丢弃一些节点。当 `is_train == False` 时，BN 层会使用训练时的平均值和方差，Dropout 层不会丢弃任何节点。`reuse = tf.AUTO_REUSE` 表示 TensorFlow 会自动的复用之前的参数。

2.2 Dropout

为了预防过拟合，在训练过程中，我们会随机丢弃一些节点，即在正向传播时把该节点的输出设为 0。在 $\text{drop rate} = p$ 时，会独立的以 p 的概率丢弃每个点。为了让输出的规模 and 没有 dropout 时接近，我们会把输出的权重乘上 $\frac{1}{1-p}$ 。在测试时不会丢弃任何节点。

我直接使用了 `tf.layers.dropout` 函数。

2.3 Batch Normalization

为了解决 internal covariate shift，我们会在每层中把每一维在 MiniBatch 中做一次归一化，并加一点偏移。具体来说，令一个 MiniBatch 中某一层的输出为 $(x^{(1)}, x^{(2)}, \dots, x^{(N)})$ ，变换之后的结果为

$$x_i^{(k)} = \gamma \cdot \frac{x_i^{(k)} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta$$

, 其中 μ_i, σ_i 为该神经元的输出在整个 MiniBatch 上的平均值和标准差。

有时候我们会每次给出一个输入进行测试, 这时候无法直接使用整个 MiniBatch 的平均值和方差, 所以需要维护训练过程中的整体平均值和方差。记 t 时刻维护的平均值为 μ_t , 第 t 次测试的平均值为 a_t , 令 $\mu_t = momentum \times \mu_{t-1} + (1 - momentum) \times a_t$ 。方差的维护方法相同。测试时会使用维护的平均值和标准差代替当前 MiniBatch 上的平均值和标准差。

我直接使用了 `tf.layers.batch_normalization` 函数。Momentum 为默认的 0.99。该函数会对输入的最后一维做 Batch Normalization。

3. Experiments

3.1 Datasets

使用了 Cifar-10 数据集。数据集包含训练集和测试集。

3.2 Implementation Details

不同模型之间的对比

搭建 MLP 和 CNN 模型, 其中两个模型含有 BN 层, 另外两个不含 BN 层。

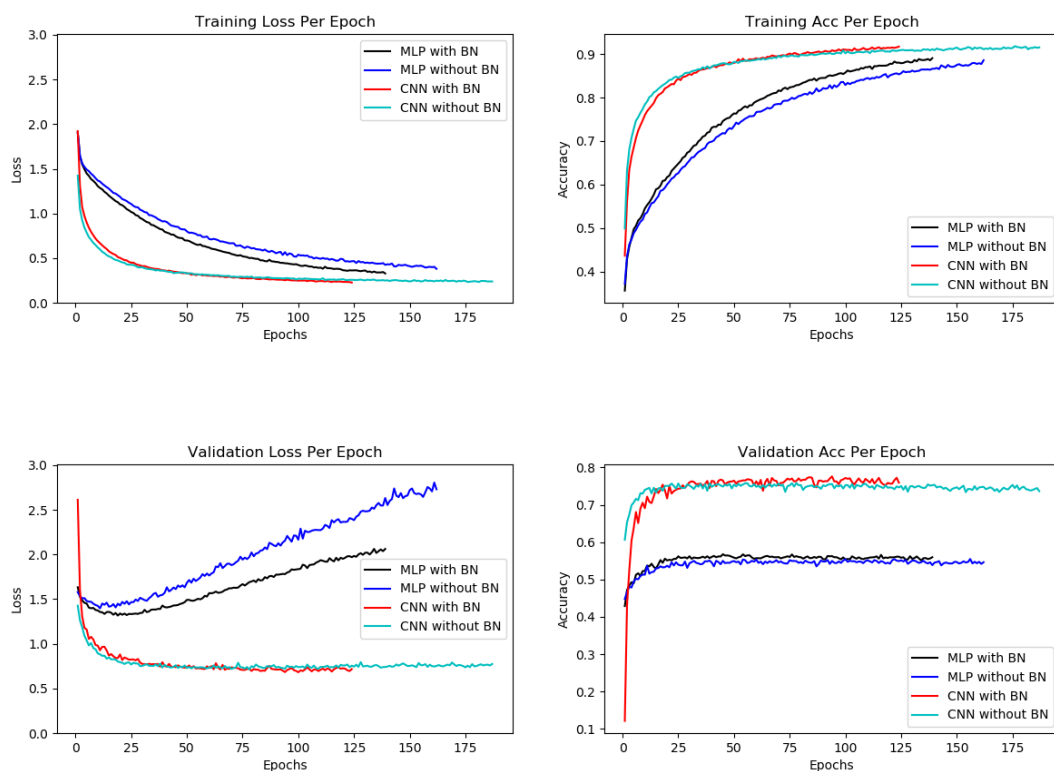
Batch Normalization 的作用

搭建 MLP 和 CNN 模型, 其中两个模型含有 BN 层, 另外两个不含 BN 层。另外还搭建了 Drop Rate 不同的 MLP 模型。

Drop Rate 的影响

搭建两个模型 (MLP with BN, MLP without BN), 设置不同的 Drop Rate。

3.3 Quantitative Results

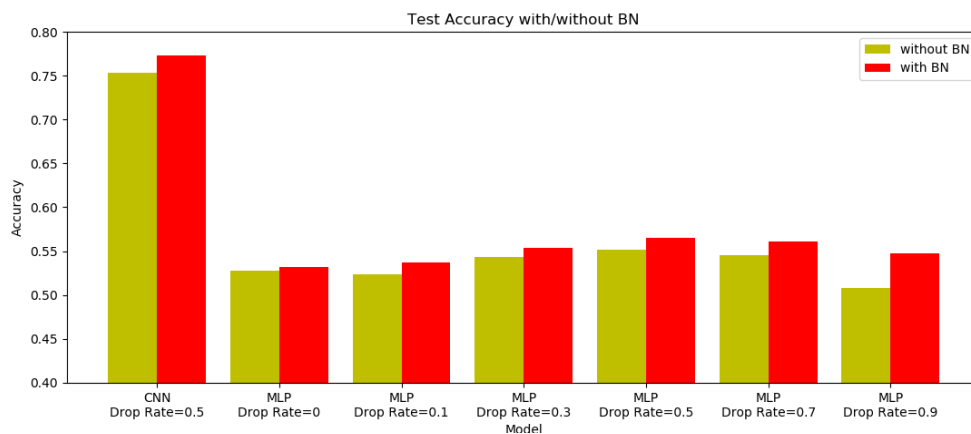


MLP and CNN models

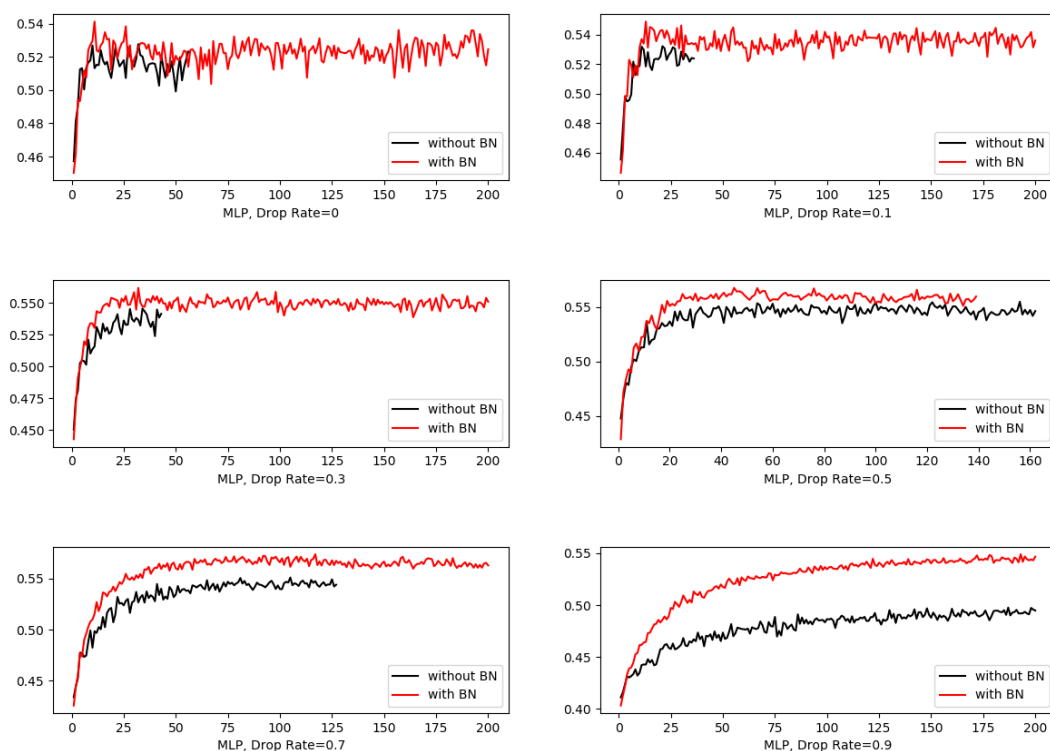
以下的正确率为验证集上的正确率达到最高时在测试集上的正确率。

模型	每个 epoch 所需时间	正确率
MLP, with BN, Drop Rate=0.5	25 Second	56.5%
MLP, without BN, Drop Rate=0.5	22 Second	55.1%
CNN, with BN, Drop Rate=0.5	824 Second	77.3%
CNN, without BN, Drop Rate=0.5	301 Second	75.3%

可以看出，CNN 模型的正确率和每个 Epoch 的训练时间都远大于 MLP 模型。

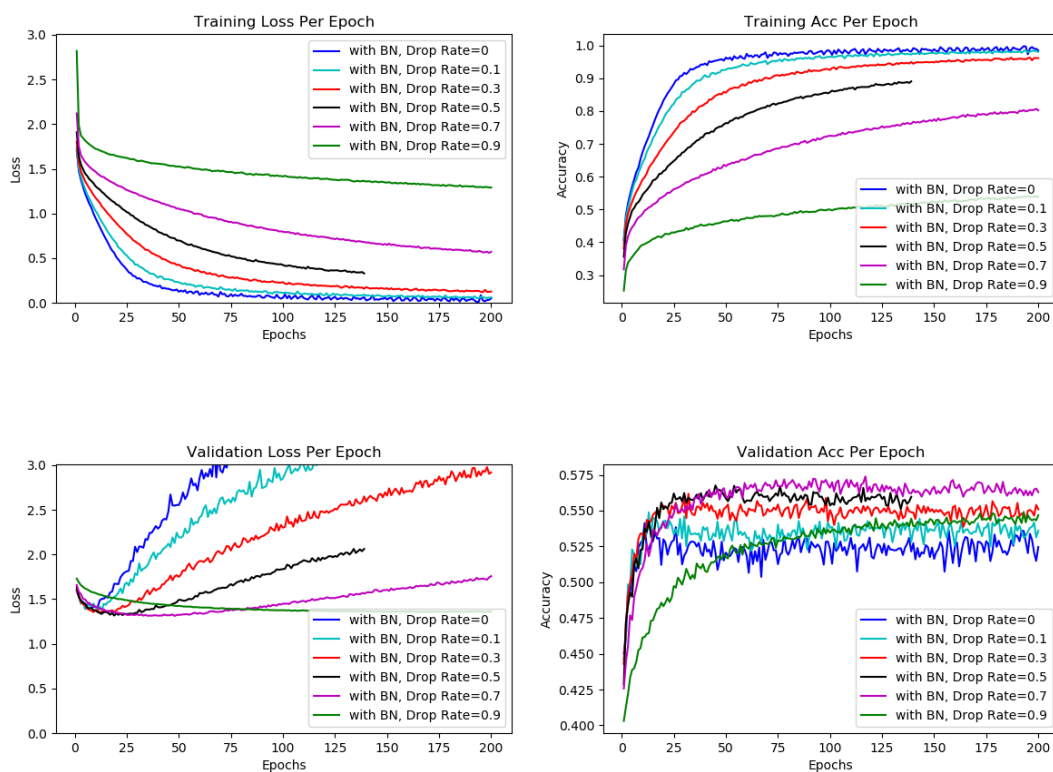


各个包含/不包含 BN 的模型的对比

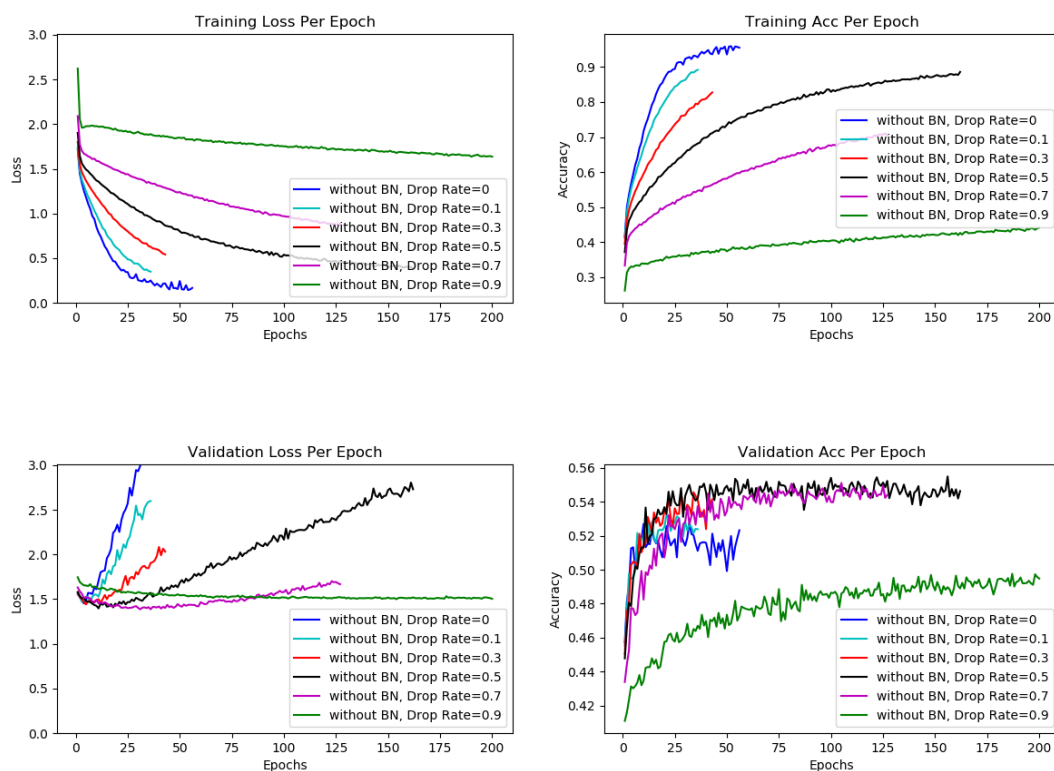


各个包含/不包含 BN 的模型的对比

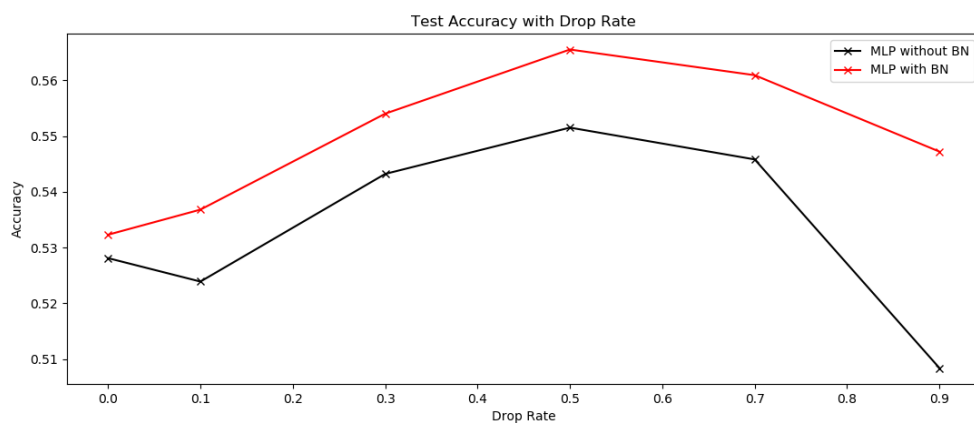
通过对比，可以发现：含有 Batch Normalization 层的网络无论在 Drop Rate 是多少时都有较高的正确率，且正确率差别比较小。那么可以得到结论：Batch Normalization 可以提高模型的正确率，增加收敛速度，但是会增加每个 Epoch 的训练时间。



MLP models with BN

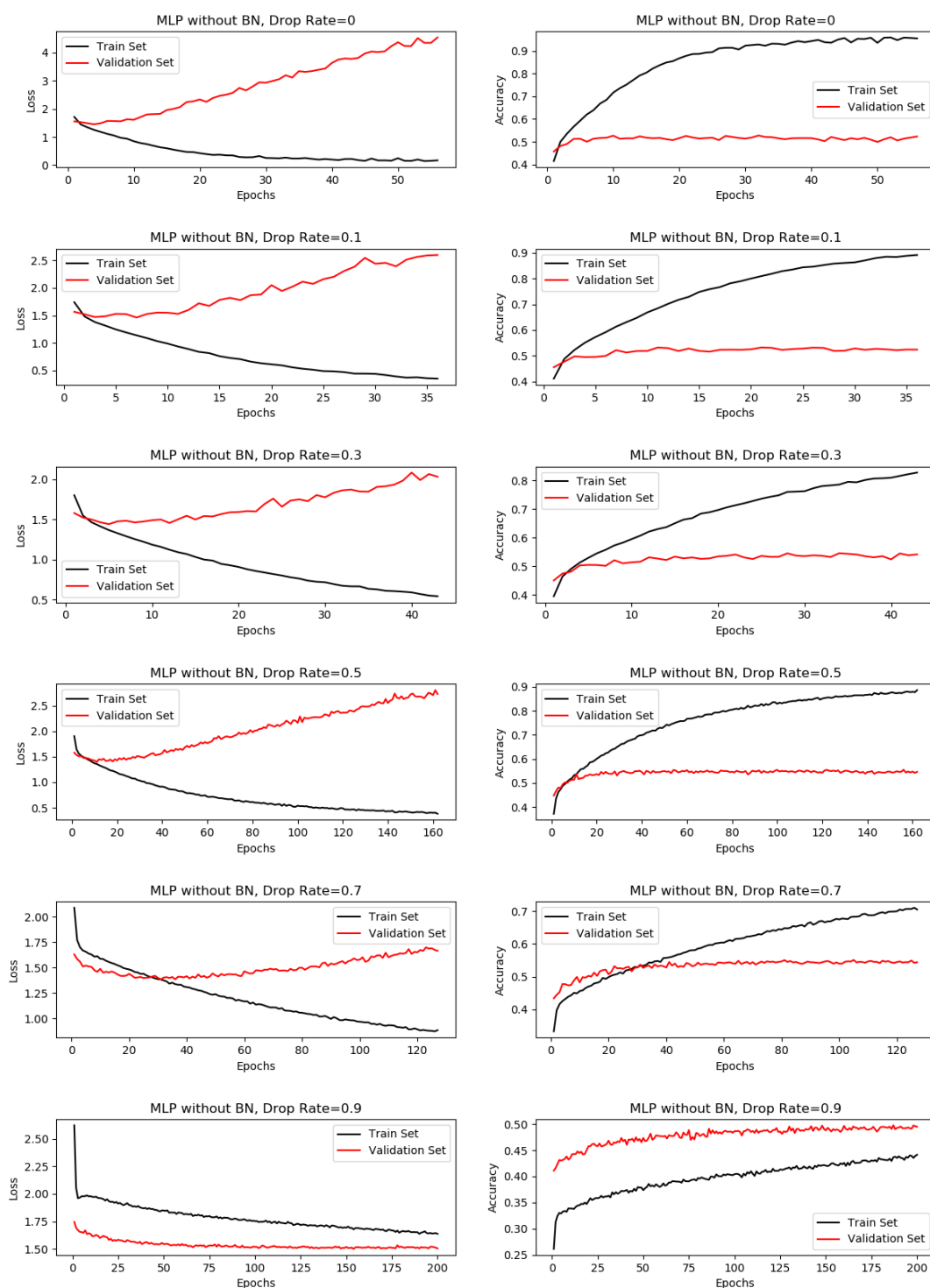


MLP models without BN



Accuracy with DropRate

模型	Drop Rate	Accuracy, with BN	Accuracy, without BN
MLP	0	53.2%	52.8%
MLP	0.1	53.7%	52.4%
MLP	0.3	55.4%	54.3%
MLP	0.5	56.5%	55.1%
MLP	0.7	56.1%	54.6%
MLP	0.9	54.7%	50.8%



Dropout

观察发现：当 Drop Rate 过低时，会有严重的过拟合现象，且正确率较低。当 Drop Rate 过高时，训练速度会变慢，且正确率较低。当 Drop Rate 接近 0.5 时，正确率最高。

4. Conclusion

在这次作业中，我对 CNN、Dropout 以及 Batch Normalization 进行了一些研究。我发现 CNN 能够大幅度提升图像分类的正确率，Dropout 能够提升正确率，Batch Normalization 能够提升正确率和收敛速度。

遗憾的是，这几个模型的正确率仍然不能令人非常满意，我们仍然需要去寻找一些正确率更高的模型。

5. Others

Q: Explain why training loss and validation loss are different. How does the difference help you tuning hyper-parameters?

在没有 Dropout 时，网络是对训练集进行优化的，但是训练集和验证集并不完全相同，因此会导致过拟合，学习到训练集的特征而不是整个问题的特征，失去部分泛化能力。表现为训练集上的正确率远大于验证集上的正确率。

在 Drop Rate 比较大时，训练时难以用极少的参数表示出所有数据的特征，会导致训练集上的正确率小于验证集上的正确率。

因此，当过拟合现象比较严重时，应该增大 Drop Rate。当训练集上的正确率一直小于验证集上的正确率时，应该减小 Drop Rate。

References

[TensorFlow API Documents](#)

[Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift](#)