

Cifar-10 Classification with MLP and CNN

张益铭 车13 2021010552

self.training

`self.training` 是一个 `bool` 类型的变量，用于指示当前模型是否处于训练状态。当设置 `model.train()` 时，`self.training` 会被设置为 `True`，而当设置 `model.eval()` 时，`self.training` 会被设置为 `False`。

本次实验中，`self.training` 的作用主要体现在以下方面

- `BatchNorm` 层：训练时，该层会根据当前batch的均值和方差进行归一化，而测试时，由于无法得到当前batch的均值和方差，因此需要使用之前训练得到的均值和方差进行归一化，即移动平均。
- `Dropout` 层：训练时，该层会随机丢弃一部分神经元，并给剩余神经元乘上 $\frac{1}{1-p}$ ，而测试时，该层不会进行任何操作。

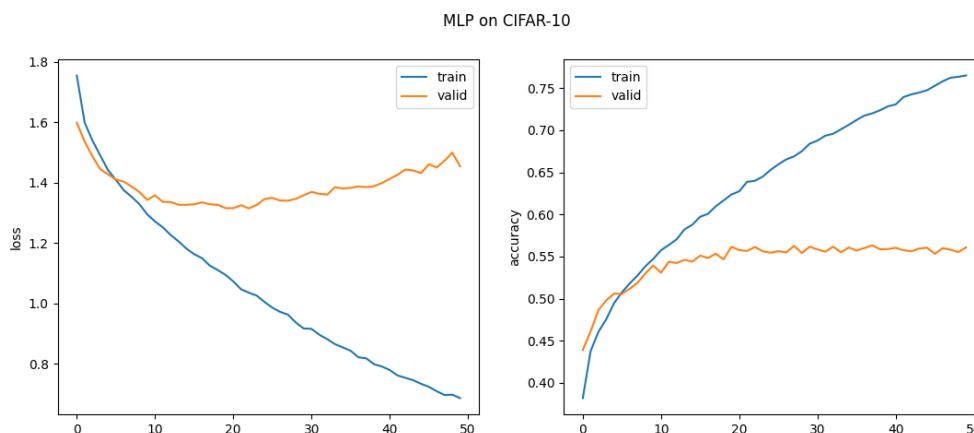
具体实验

MLP

效果最好的MLP模型如下，`num_epochs=50`，其余超参为默认值。

```
1 self.fc1 = nn.Linear(3 * 32 * 32, 1024)
2 self.bn = BatchNorm1d(1024)
3 self.relu = nn.ReLU()
4 self.dropout = Dropout(drop_rate)
5 self.fc2 = nn.Linear(1024, 10)
```

准确率以及loss曲线如下：



CNN

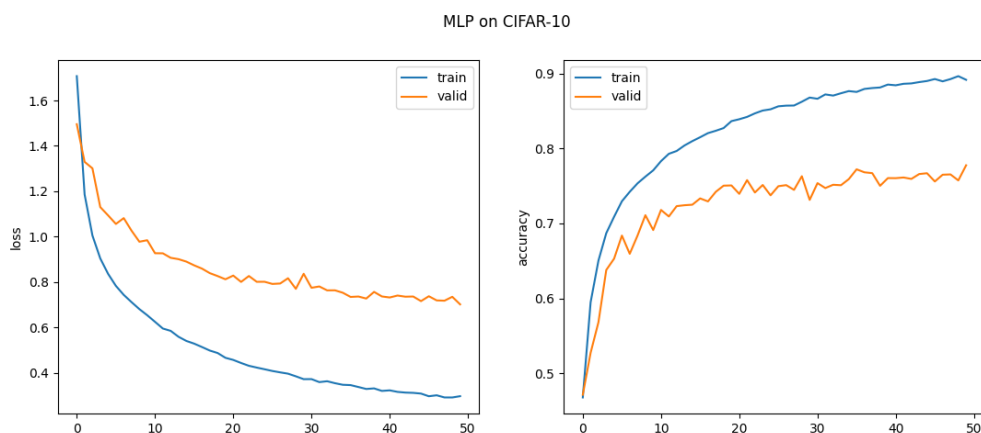
效果最好的CNN模型如下，`num_epochs=50`，其余超参为默认值。

```

1 self.conv1 = nn.Conv2d(3, 64, 5, padding=2)
2 self.bn1 = BatchNorm2d(64)
3 self.relu1 = nn.ReLU()
4 self.dropout1 = Dropout(drop_rate)
5 self.pool1 = nn.MaxPool2d(2, stride=2)
6
7 self.conv2 = nn.Conv2d(64, 128, 3, padding=1)
8 self.bn2 = BatchNorm2d(128)
9 self.relu2 = nn.ReLU()
10 self.dropout2 = Dropout(drop_rate)
11 self.pool2 = nn.MaxPool2d(2, stride=2)
12
13 self.fc = nn.Linear(128 * 8 * 8, 10)

```

准确率以及loss曲线如下：



Loss

训练和验证的loss不同的原因有以下几点：

1. 过拟合。训练集 loss 较低，而验证集 loss 较高，并且随着训练进行，验证集 loss 持续上升（从 MLP 的 loss 曲线可以看出）
2. 欠拟合。模型在训练集和验证集上表现都不好，说明模型复杂度不足，无法有效捕捉数据中的模式。表现为训练集和验证集的 loss 都很高，并且随着训练进行，loss 几乎不再下降
3. 模型训练不稳定。可能是由于学习率过高，导致模型在训练时波动过大，无法有效收敛。表现为训练集的 loss 大幅波动，而验证集的 loss 可能随机变化或保持高位

调整超参的方法：

1. 如果验证集 loss 比训练集高很多且持续上升，模型可能过拟合，可以考虑加入 regularization 或减小模型复杂度
2. 如果训练和验证集的 loss 都很高，模型可能欠拟合，可以增加模型的复杂度或增加训练的 epoch 数目
3. 加入 early stopping，通过监控验证集的表现，决定何时停止训练，避免过拟合

准确率

	Training	Validation	Test
MLP	0.7199749828875065	0.5632999897003174	0.5613999861478806

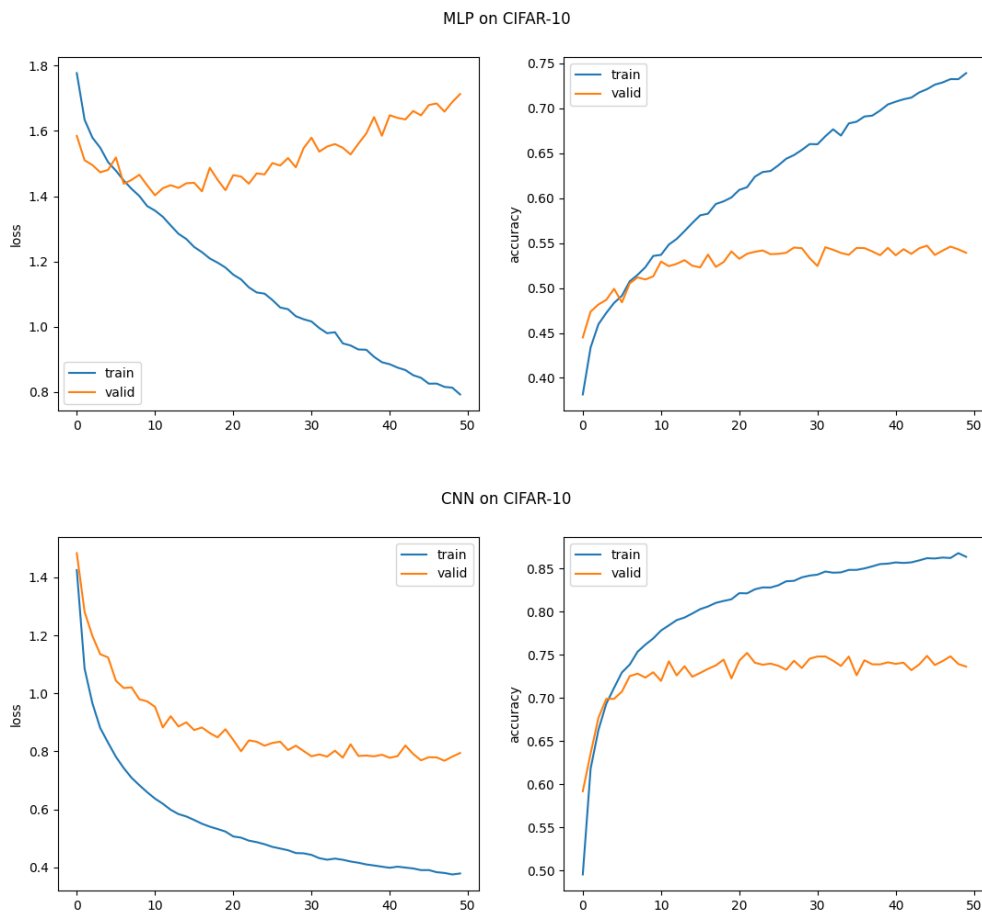
	Training	Validation	Test
CNN	0.89149998113513	0.7774999785423279	0.7697999811172486

从上表可见，CNN模型的准确率明显高于MLP模型，可能的原因有以下几点：

1. MLP将输入数据展平成一个长向量，忽略了输入数据的空间结构，无法捕捉到像素之间的空间关系。而CNN的卷积核在图像上滑动，处理局部的像素块，可以有效捕捉到局部特征。CNN在图像分类任务中能够捕捉图像的空间依赖性，这使得它比MLP更适合处理图像数据
2. 在MLP中，每一层的神经元与前一层的所有神经元是全连接的，这导致参数量非常庞大。由于每个像素都需要一个独立的权重，导致模型容易过拟合，并且计算效率低。而CNN卷积核在整个图像上共享相同的参数，只与局部像素块进行计算，减少了计算成本，增强了模型的泛化能力。相比之下，CNN的计算效率更高，并且不容易过拟合

Batch normalization

删除batch normalization层后，mlp与cnn的训练结果如下所示：

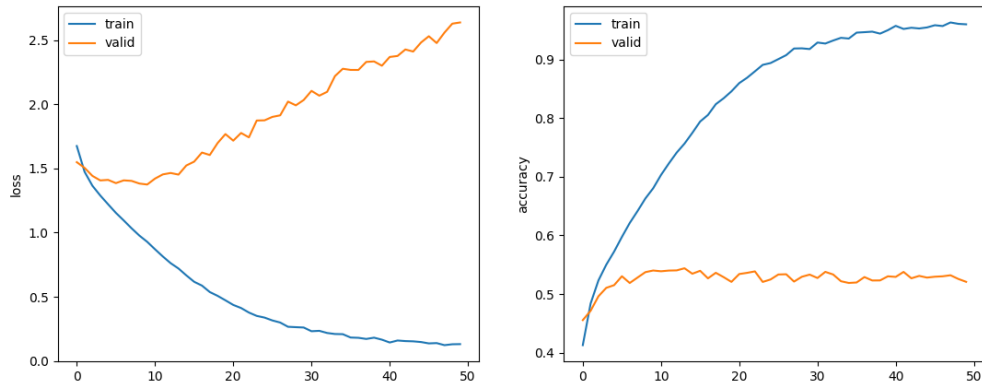


从上图可以看出，删除batch normalization层后，模型的loss曲线变得更加波动，且验证集的准确率下降。batch normalization层的作用是对每个batch的数据进行归一化，使得模型更容易收敛，提高了模型的泛化能力。因此，删除batch normalization层后，模型的训练效果变差。

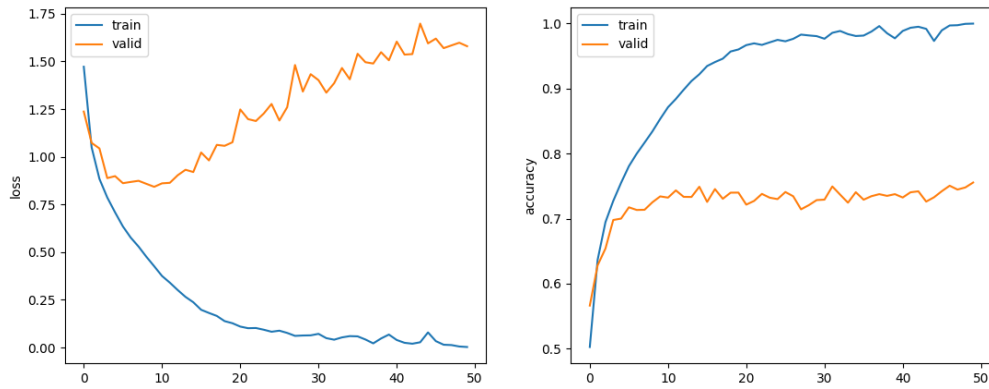
Dropout

设置 `--drop_rate=0`，mlp与cnn的训练结果如下所示：

MLP on CIFAR-10



CNN on CIFAR-10



从上图可以看出，设置 `drop_rate=0` 后，验证集的loss曲线上升更加明显，且验证集的准确率下降，出现了较为严重的过拟合。dropout层的作用是随机丢弃一部分神经元，使得模型更加健壮，同样提高了模型的泛化能力。因此，删除dropout层后，模型出现过拟合，训练效果变差。