lab2 实验报告

- 刘兆宸
- PB21061373

实验目的

使用 pytorch 或者 tensorflow 实现卷积神经网络 CNN，在 CIFAR-10 数据集上进行图片分类。
研究 dropout、normalization、learning rate decay 等模型训练技巧，以及卷积核大小、网络深度等超参数对分类性能的影响。

实验过程

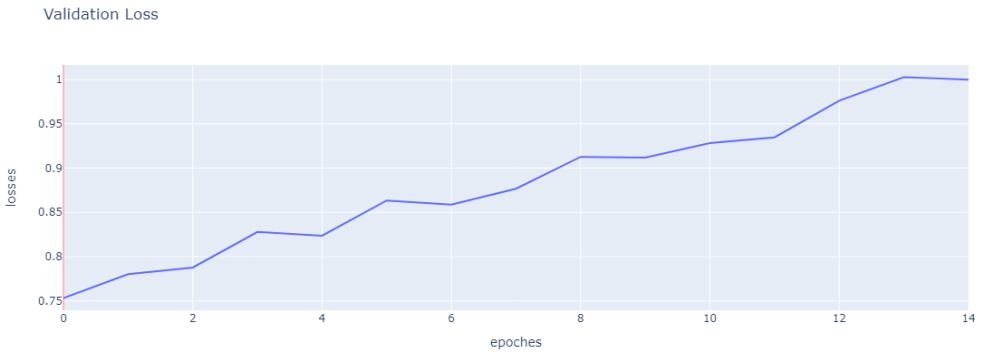
框架分为几步:先对CIFAR-10数据集进行预处理,然后搭建卷积神经网络，训练模型，最后对模型进行评估。

首先我用了一个简单的LeNet5的裁剪版，包含两个卷积层和一个全连接层，用于对 CIFAR-10 数据集进行分类。这个网络的结构如下:

```
>>> summary(model, input_size=(C, H, W)=(3, 32,
32)),batch_size=8
----------------------------------------------------------------
-
        Layer (type)               Output Shape         Param #
================================================================
=
            Conv2d-1          [-1, 6, 29, 29]             294
         MaxPool2d-2          [-1, 6, 14, 14]               0
            Conv2d-3         [-1, 16, 11, 11]           1,552
         MaxPool2d-4           [-1, 16, 5, 5]               0
            Linear-5                 [-1, 64]          25,664
            Linear-6                 [-1, 32]           2,080
            Linear-7                 [-1, 10]             330
================================================================
=
```

> *对于nn.conv2d,输出的尺寸计算公式为:*
> $O = \frac{W-K+2P}{S} + 1$
> *其中O为输出尺寸,W为输入尺寸,K为卷积核尺寸,P为padding,S为stride,如果按默认值,padding=0,stride=1,公式简化为:*
> $O = W - K + 1$

然后我对这个网络进行了训练，但是结果一般，收敛时,Validation-Acc只有0.6。然后我调整了batch_size=32,64进行测试,准确率有所提升(0.63,0.69),于是实验采用batch_size=64。
然而loss却随着epoches增加而增加,很显然过拟合了,如下图:



对于该数据集(size:40000 $\times$ 3 $\times$ 32 $\times$ 32),LeNet5模型无法进一步提取出有效的特征,
所以我们考虑增加模型的深度,比如增加卷积层和全连接层。

下面我们先研究模型结构对性能的影响，然后再研究训练技巧。

模型结构对性能的影响

原来的LeNet5模型结构由2层卷积层和3层全连接层组成。

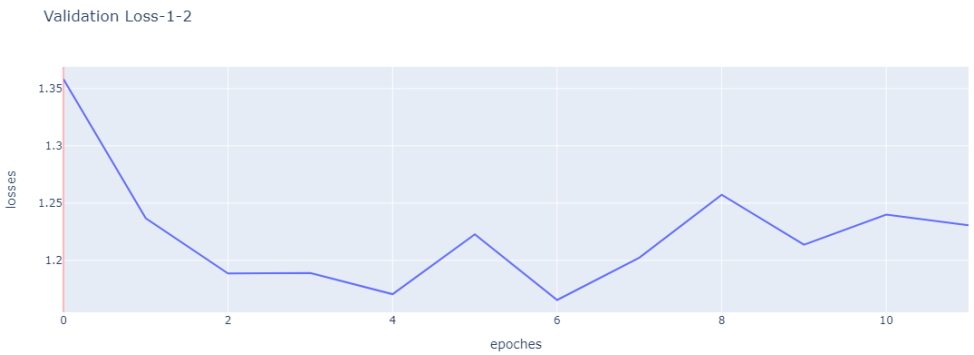
模型层数的影响

- 1层卷积层，2层全连接层

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 6, 28, 28]             456
         MaxPool2d-2          [-1, 6, 14, 14]               0
            Linear-3                [-1, 64]          75,328
            Linear-4                [-1, 10]             650

Train Accuracy : 0.7144
Val Accuracy : 0.5968
```

loss如下图所示:



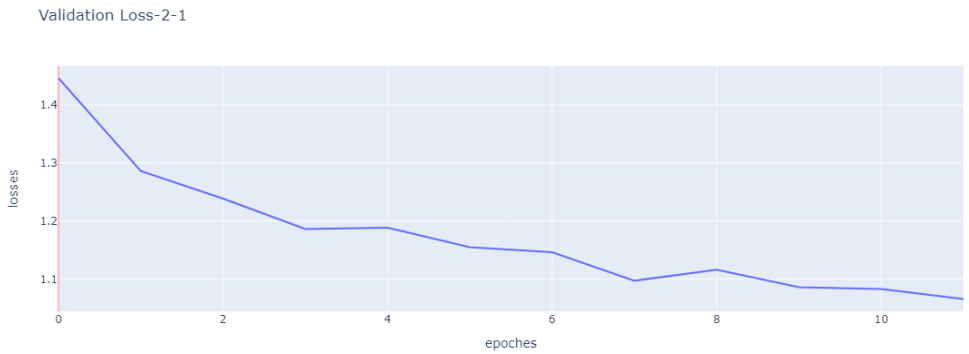
观察Loss曲线发现几个epoch之后收敛不稳定,并且性能不如之前那个的模型，说明模型太简单了，无法提取出有效的特征。

此处模型性能有所提升,但是还是不够理想。并且参数量太大,我们将卷积层增加为2层,如下:

- 2层卷积层，1层全连接层

```
----------------------------------------------------------------
-
        Layer (type)               Output Shape         Param #
================================================================
=
            Conv2d-1           [-1, 6, 28, 28]             456
         MaxPool2d-2           [-1, 6, 14, 14]               0
            Conv2d-3          [-1, 16, 10, 10]           2,416
         MaxPool2d-4            [-1, 16, 5, 5]               0
            Linear-5                  [-1, 10]           4,010
================================================================
=
Train Accuracy : 0.6601
Val Accuracy : 0.6345
```

loss如下图所示:



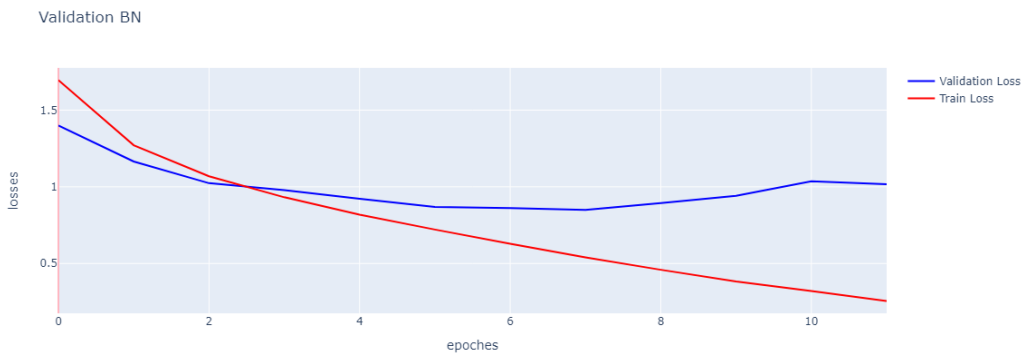
这张图显示了模型的性能有所提升，并且没有出现明显的过拟合现象。说明多层的卷积层可能会提取出更多的特征，从而提高模型的性能。

- 2层卷积层，3层全连接层
  一开始的LeNet5模型结构由2层卷积层和3层全连接层组成,不再赘述。
- 4层卷积层，3层全连接层

上述实验表现都不太好,我们考虑用更深的网络来训练,我先参考了AlexNet的结构,
AlexNet 用了5层卷积层提取特征和3层全连接层用来分类,但是明显感觉网络尺寸对于CIFAR-10数据集太大了,所以我对其进行了裁剪
(对第一层卷积核裁剪，并减少各层的通道数,线性层减小宽度),如下:

```
----------------------------------------------------------------
-
        Layer (type)               Output Shape         Param #
================================================================
=
            Conv2d-1          [-1, 32, 15, 15]           4,736
            Conv2d-4            [-1, 96, 7, 7]          76,896
            Conv2d-6           [-1, 192, 7, 7]         166,080
            Conv2d-8           [-1, 128, 7, 7]         221,312
           Linear-13                 [-1, 512]         590,336
           Linear-16                 [-1, 512]         262,656
           Linear-18                  [-1, 10]           5,130
================================================================
=
Train Accuracy EPOCH 12: 0.8642
Val Accuracy EPOCH 12: 0.6859
```

loss如下图所示:



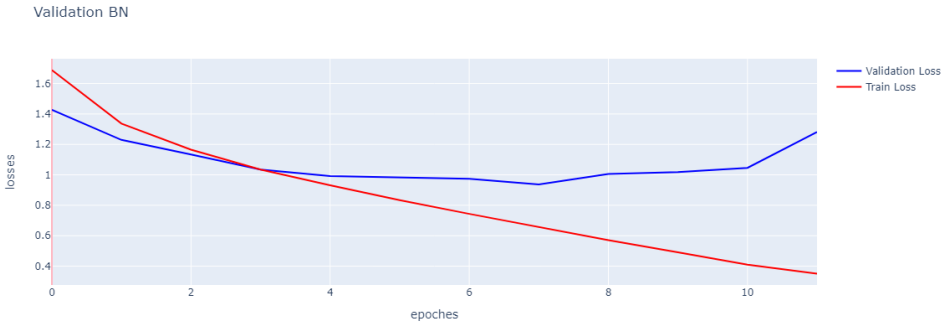
可以发现，验证集和测试集的loss存在较大的差距，说明模型存在过拟合现象,可以考虑在我们后续的Tricks中加入dropout,BN等技巧。

这个模型深度的结构貌似感觉已经还可以了，我们再研究一下卷积核尺寸的影响。

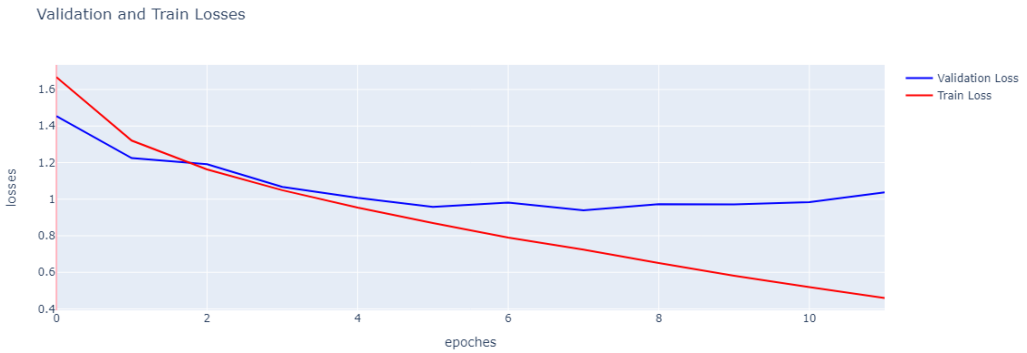
卷积核尺寸的影响

上述实验中，卷积核的尺寸是(7,5,3,3),下面考虑改变各层卷积核尺寸(尺寸一般是奇数，这样可以保证卷积核有一个中心点),下面考虑卷积核尺寸的影响(3x3和7x7)。

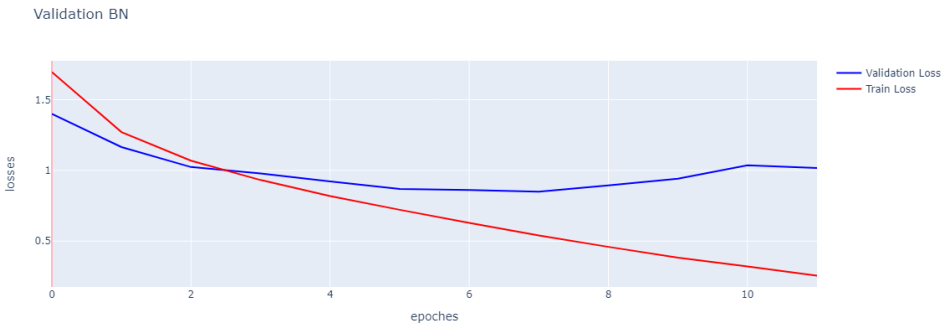
- (9,5,3,3)



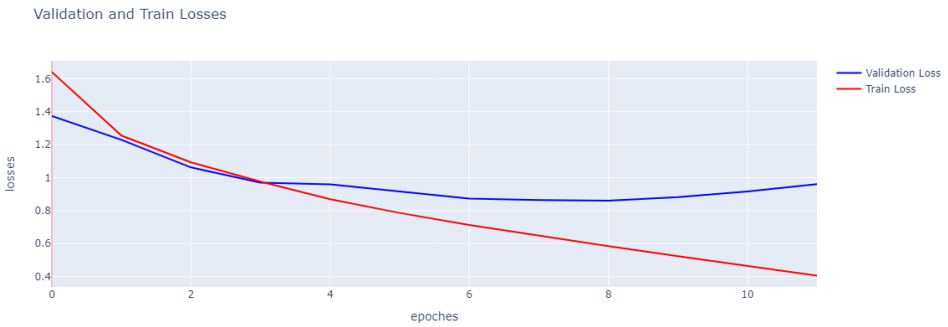
-(9 5 3 1)



- (7,5,3,3)



- (5,5,3,1)




```
Train Accuracy : 0.8552
Val Accuracy : 0.7029
```

由上述实验看出,卷积核尺寸为(5,5,3,1)时,loss最小，性能最好。说明卷积核尺寸的选择对模型的性能有很大的影响。
我们接下来暂时以该模型为基础进行训练。

Tricks

对于模型的训练的一些tricks,首先最开始我们已经提到了batch_size的选择,然后我们再来看看其他的tricks。

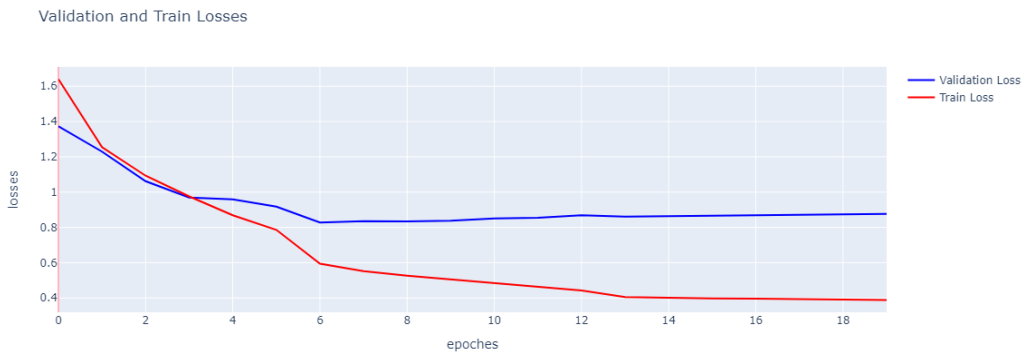
在数据提取部分我们使用了 `transforms.Normalize( [0.485, 0.456, 0.406], [0.229, 0.224, 0.225])` 对数据进行了归一化处理。
因为数据集较为简单我们没有使用其他数据增强技巧。

> *这个数据是pytorch统计的ImageNet的数据集的均值和方差以作为归一化的标准*

接下来我们主要讨论学习率衰减，dropout，batch normalization等技巧。
注意到我们的模型存在过拟合现象(验证集比训练集loss大不少)，所以可以主要着手减少过拟合现象。

学习率衰减

学习率衰减是一种常用的训练技巧，可以提高模型的性能。在训练过程中，随着训练的进行，模型的性能会逐渐收敛，这时候我们可以逐渐减小学习率，以提高模型的性能。在这里我们使用了 `torch.optim.lr_scheduler.StepLR` 来实现学习率的衰减。我们设置了每8个epoch衰减一次，衰减因子为0.1。实验结果如下:




```
Train Accuracy EPOCH 14: 0.8643
Val Accuracy EPOCH 14: 0.7487
```

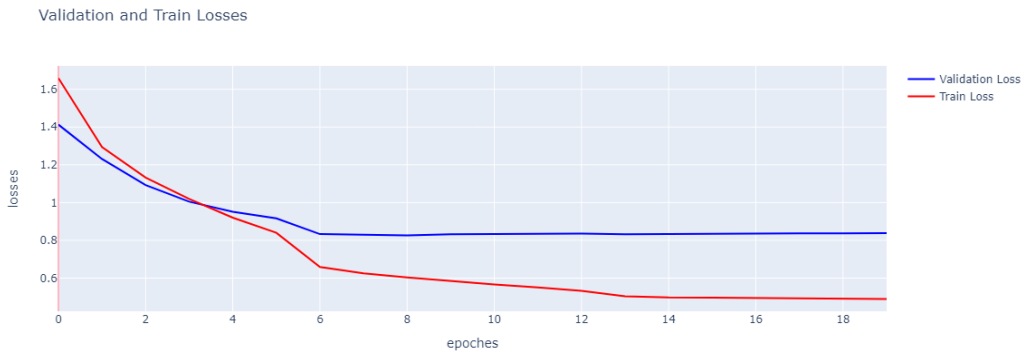
性能略微提升并且,且模型收敛更稳定了,loss也略微变小了。

Dropout

Dropout是一种常用的正则化技巧，可以提高模型的泛化能力。在训练过程中，我们随机的将一些神经元的输出设置为0，这样可以减少模型的过拟合现象。在这里我们使用了 `nn.Dropout` 来实现Dropout。我们设置了Dropout的概率为0.15。实验结果如下：

```
Train Accuracy EPOCH 20: 0.8290
Val Accuracy EPOCH 20: 0.7317
```

loss如下图所示:



我们发现ACC和loss都略有变差,但是训练集和验证集的loss差距变小了,说明模型的泛化能力有所提升。因此我们可以增加迭代次数进一步训练模型。

Batch Normalization

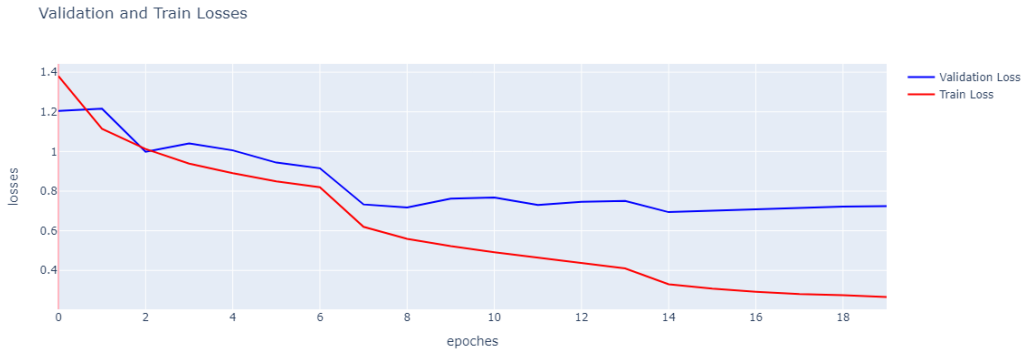
Batch Normalization是一种常用的正则化技巧，可以提高模型的泛化能力，加速模型的收敛。
在这里我们使用了 `nn.BatchNorm2d` ,参考了resnet的结构,在每个卷积层后面都加入了Batch Normalization。

按照 `conv->bn->relu->pool` 的顺序建模。

实验结果如下：

```
Train Accuracy EPOCH 20: 0.9184
Val Accuracy EPOCH 20: 0.7672
```

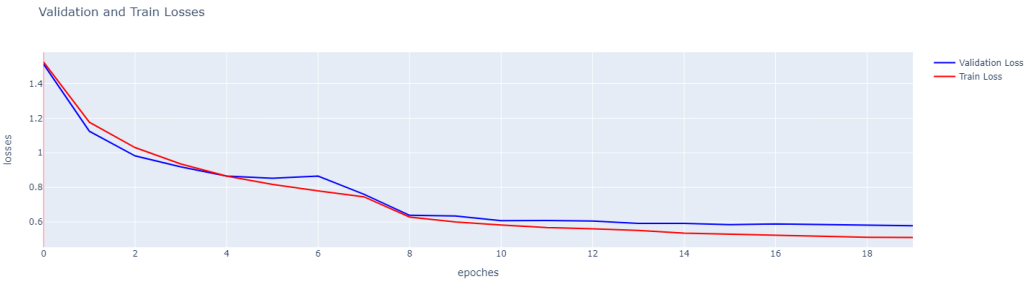
loss如下图所示:



可以发现模型的性能有所提升,并且收敛速度更快了。说明Batch Normalization对模型的性能提升还是比较稳定的

总的来说,感觉网络结构对性能的影响更大,而tricks对性能的影响相对较小,但是tricks的优势在于能提高模型的泛化能力(减小过拟合方面),加速模型的收敛。

然而此时验证集还没有达到预想的80%的准确率(因为没有用res块,本质还算比较浅的网络),于是又仔细调整参数(还有正则化技巧比如weight_decay调大一点),加入了数据增强方法等,但是效果不是很明显,所以我们考虑使用更深的网络结构。
这里我们参考了VGGNet的结构，思路是用多层的卷积核减小特征图大小，增加通道数目,这里我使用了类似VGGNet的8层卷积层和3层全连接层的结构，最终在数据增强的验证集上达到了近80%的准确率。



Test

最后拿我们的模型在测试集上进行测试，得到了如下结果:

```
Test Loss: 0.4359
Test Accuracy: 0.8519
```

收获

本次实验加深了我对卷积神经网络的理解，对于卷称神经网络的训练技巧有了更深的认识，积累了根据loss曲线进行评估调整模型参数以及策略的经验。
并且对 `batch_size` 、 `lr_decay` 、 `dropout` 、 `batch-normalization` 等超参数的调整有了更深的认识。

此外通过对几个经典模型(LeNet5,AlexNet,VGGNet,resNet)的实现，对这几个模型的结构有了更深的理解。