

# Deep Learning HW2 Report

唐建宇

2017012221

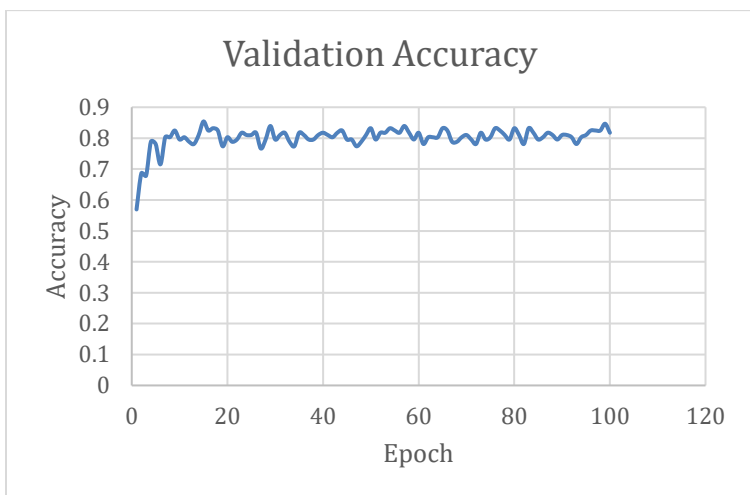
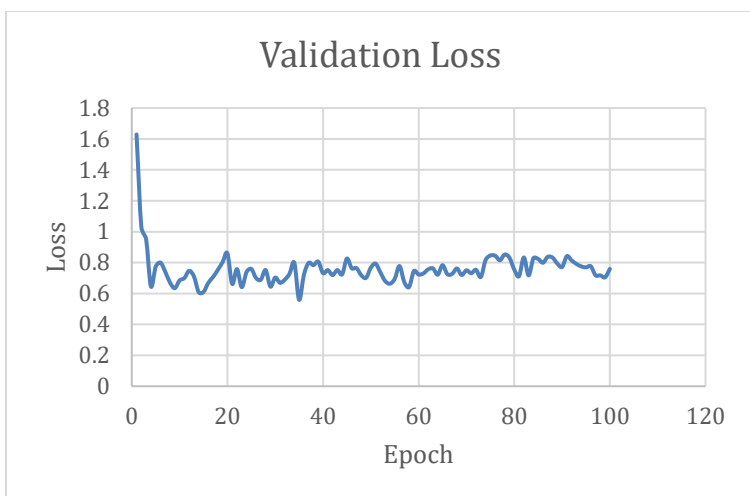
## 1. Model A

### 实验结果

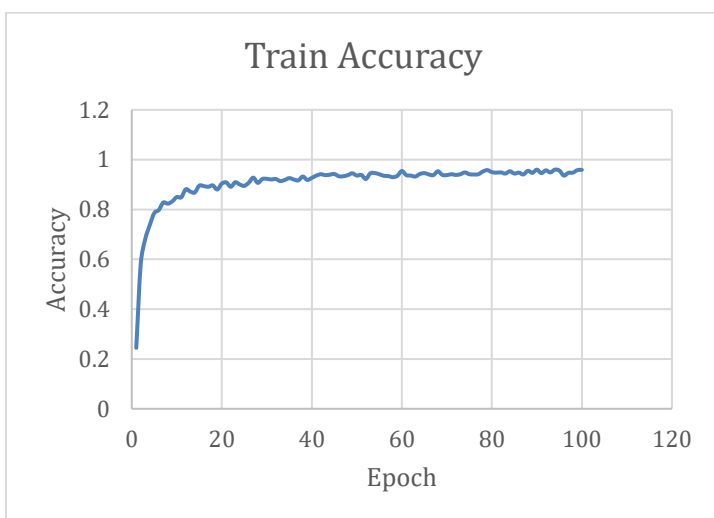
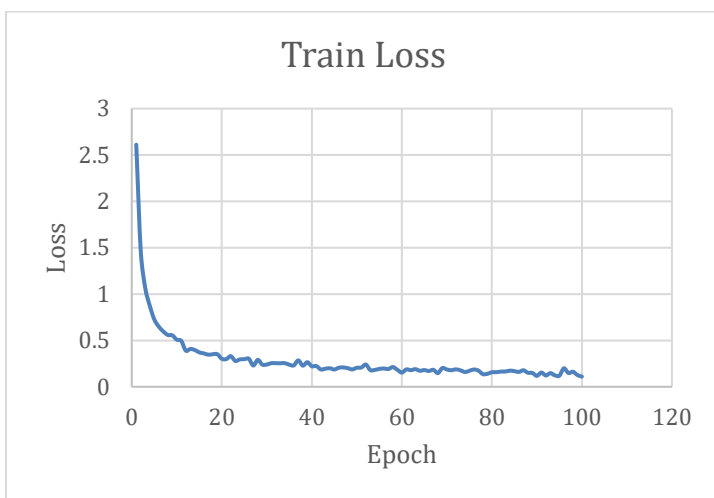
参数配置: batch\_size 48, learning rate 0.005, epoch 100, CE Loss

验证集上的准确率为 84.67%。

验证集上的 loss 曲线和准确率曲线:



训练集上的 loss 曲线和准确率曲线：



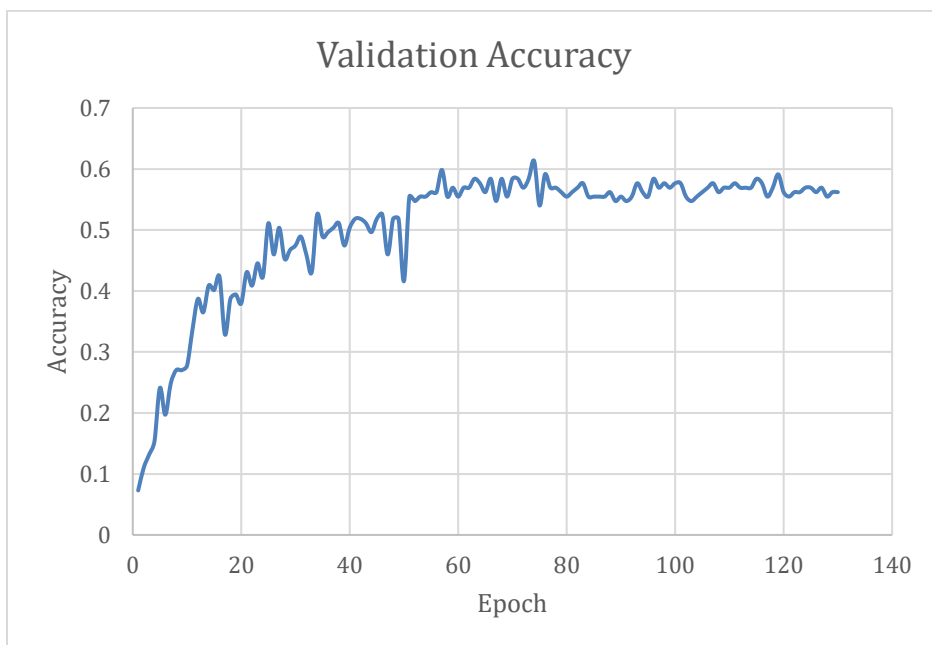
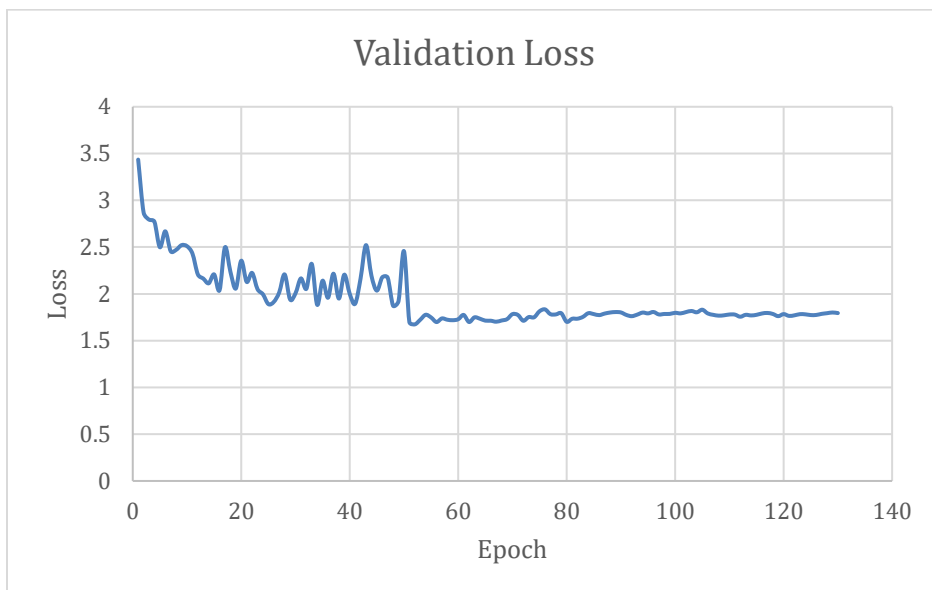
## 2. Model B

### 实验结果

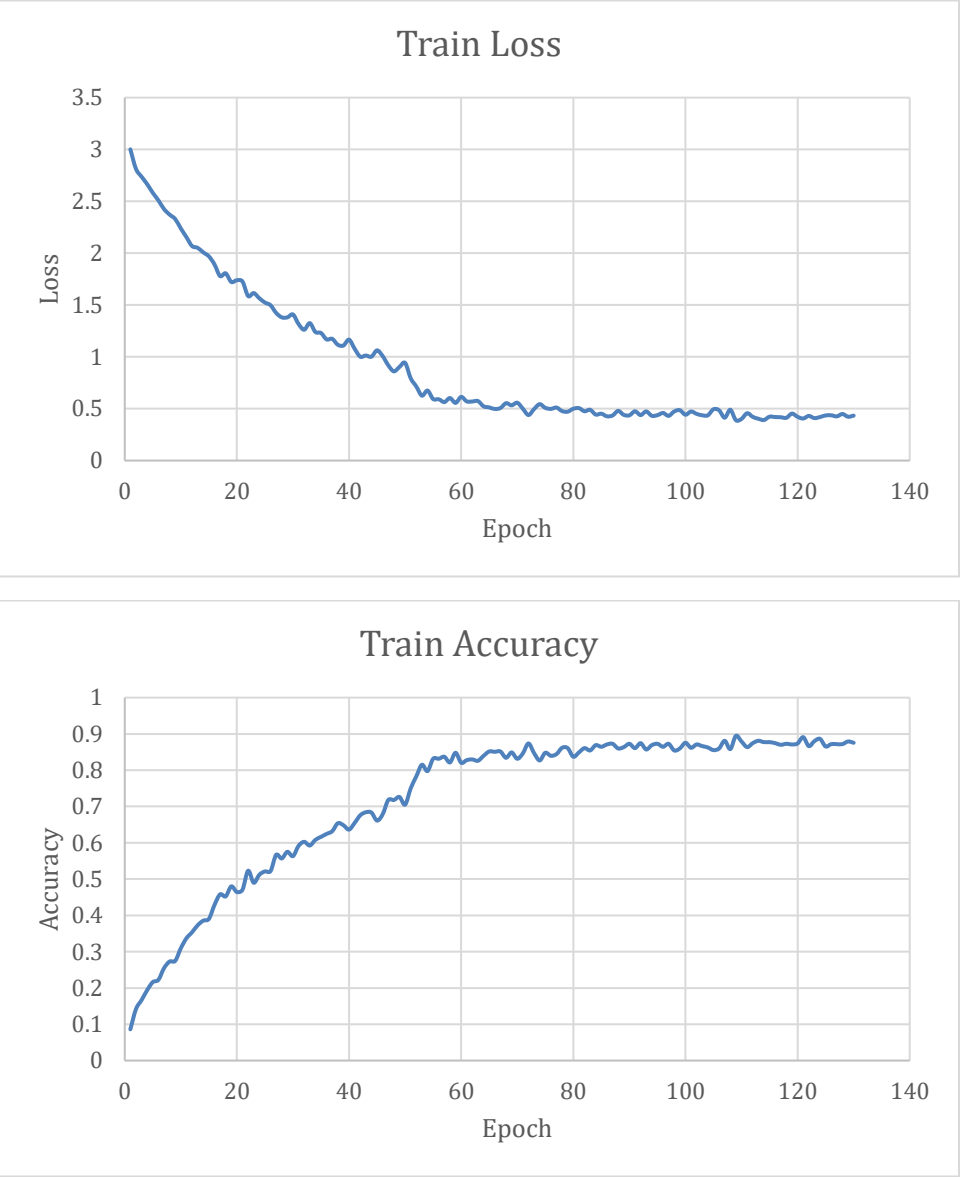
**参数配置:** batch\_size 48, learning rate 0.005, epoch 100, CE Loss

验证集上的准确率为 59.12%。

验证集上的 loss 曲线和准确率曲线:



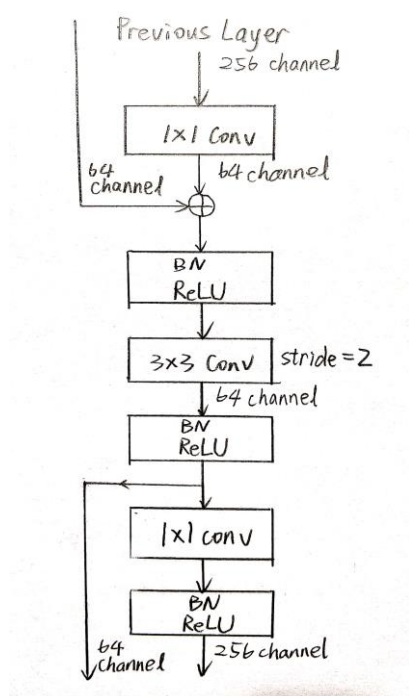
训练集上的 loss 曲线和准确率曲线：



### 3. Model C

#### a. 网络设计

参考了 ResNet 飞线连接的思路以及 GoogleNet 的 bottleneck 方法，设计了用 bottleneck 结构取代 residual block 结构的低参数量的"ResNet"。每一个 block 的结构如下图所示（图例中的输入、输出 channel 均为 256）：



Block	input channel	output channel
1	64	128
2	128	128
3	128	256
4	256	256
5	256	512
6	512	512

整个网络的输入部分采用步长为 2、dilation=2 的带洞  $3 \times 3$  卷积将输入图像大小减小为原来 1/4，channel 数变为 64；然后堆叠 6 个上述 block（堆叠方式如上表所示），最终得到 channel 数为 512 的 feature map；再通过池化变为  $1 \times 1 \times 512$  的张量并输入全连接层。

在 ResNet18 是 18 层卷积，Model C 是 19 层卷积的情况下，ResNet 的参数量为 1194836，而 Model C 的参数量仅为 441748，仅为 ResNet18 的 2/5，而且在 Validation Set 上也取得了和 ResNet18 相近的准确率，同时更加不容易过拟合。

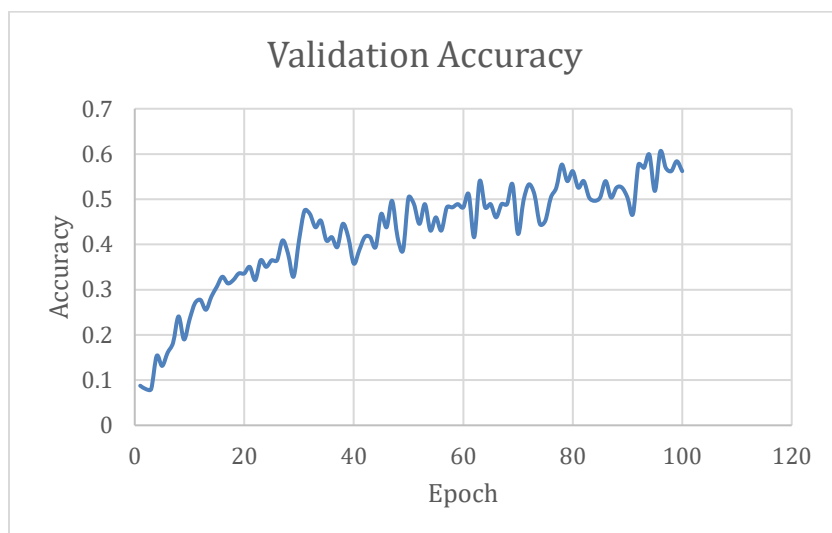
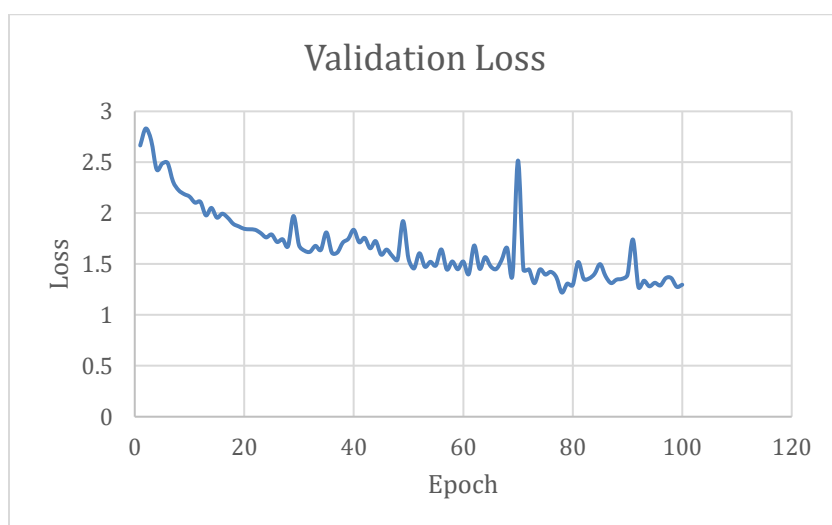
## b. 实验结果

参数配置： batch\_size 48, learning rate 0.005, epoch 100, focal loss gamma=2

### Loss 和准确率曲线

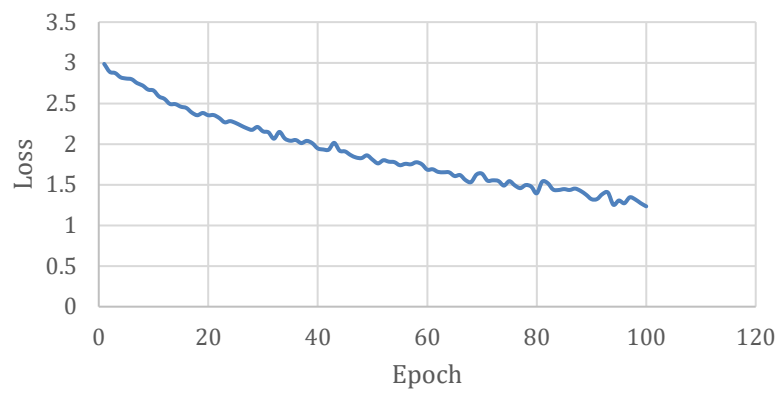
在验证集上的准确率为 60.58%。

验证集上的 loss 曲线和准确率曲线：

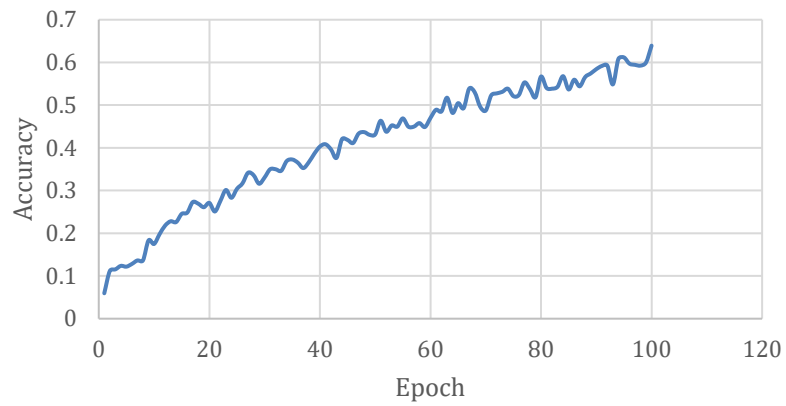


训练集上的 loss 曲线和准确率曲线：

### Train Loss



### Train Accuracy



Confusion Matrix

	Prediction																					
	Class	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Ground Truth	0	5	0	0	0	0	0	0	1	1	0	0	0	0	0	2	0	0	0	1	0	
	1	0	1	1	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	
	2	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	3	0	1	0	1	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	1	
	4	0	0	0	0	7	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	
	5	0	0	0	0	1	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	
	6	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	
	7	0	0	0	0	0	0	0	2	1	1	1	0	0	0	2	0	0	0	0	0	
	8	1	0	0	1	2	0	0	0	5	0	0	0	0	0	0	0	1	0	0	0	
	9	1	0	0	0	1	0	0	1	0	4	0	0	0	0	0	0	0	0	0	0	
	10	0	0	0	0	1	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	
	11	0	0	0	0	0	1	0	0	0	0	0	3	0	0	0	0	0	0	1	0	
	12	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	
	13	0	0	0	0	0	0	0	0	1	0	0	0	0	5	1	1	0	0	0	0	
	14	1	0	0	0	0	0	0	0	1	0	0	0	0	0	6	0	1	1	0	0	
	15	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	5	0	0	0	1	
	16	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	5	0	0	0	
	17	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	4	0	0	
	18	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	6	0	
	19	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	

c. 训练策略

数据增强

一共尝试了随机旋转、随机位置大小裁剪、随机水平翻转、随机纵向翻转、随机颜色和对比度抖动六种数据增强的方式。对这六种数据增强方式中的 6 种组合方式进行了实验，保证其他参数配置相同，记录 130 个 epoch 之后的验证集准确率如下：

数据增强方式	验证集准确率
随机位置大小裁剪	50.36%
随机旋转+随机位置大小裁剪	40.88%
随机旋转+随机位置大小裁剪+随机纵向翻转	37.23%
随机位置大小裁剪+随机纵向翻转+随机水平翻转	37.23%
随机位置大小裁剪+随机颜色和对比度抖动	42.34%
随机旋转+随机位置大小裁剪+随机颜色和对比度抖动	23.36%

最终发现只进行随机位置和大小进行裁剪得到的结果反而是最高的。这与数据集有关，因为数据集本身较小，同时训练集和验证集数据的差异很小，通过数据增强学到的泛化能力并不能在与训练集非常相似的验证集上体现出优势，反而会导致过拟合，使原本的一些特征被忽略，因此各种数据增强的方法效果反而不好。



## 学习率衰减

使用 `torch.optim` 的 `MultiStepLR`，从第 50 个 epoch 开始，每 30 个 epoch 作为一个 milestone，将学习率衰减为原来的 1/10。

实验中，对比了不衰减、普通的 `exponential` 衰减（`gamma=0.3`，从第 50 个 epoch 开始衰减）以及最终采用的上述的 `MultiStep` 衰减三种策略，在初始学习率（`lr=0.005`）和其他条件相同的情况下，实验结果如下：

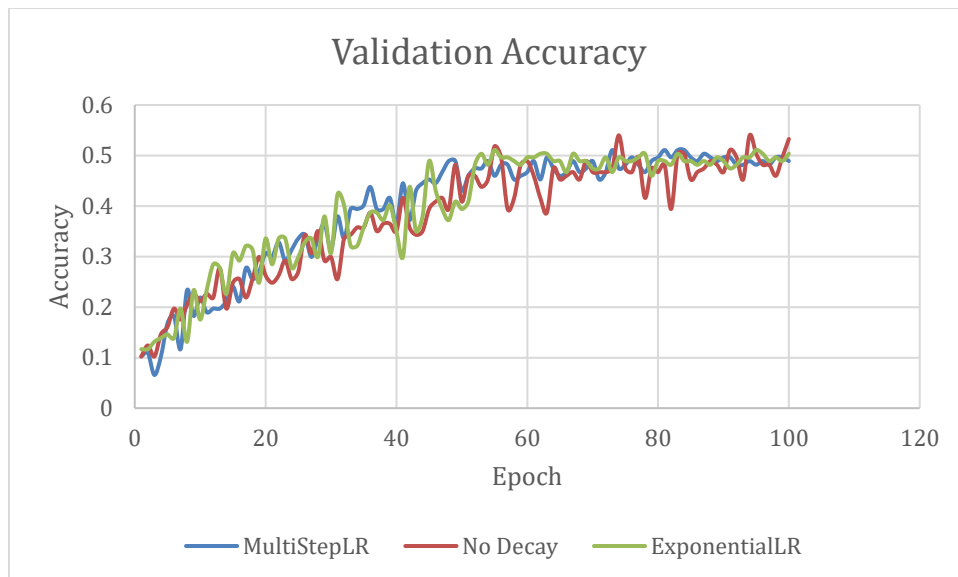
学习率衰减方案	验证集准确率
---------	--------

不衰减	50.36%
-----	--------

Exponential	51.09%
-------------	--------

MultiStep	54.01%
-----------	--------

在 100 个 epoch 上三种方案的验证集准确率变化情况如下：



可能是因为数据集较小已经存在了一定程度的过拟合，并没有看出理想中学习率衰减的点出现准确率一下子提升的现象，但采用学习率衰减后确实可以减小准确率的波动程度。

## Focal Loss

通过 `Confusion Matrix` 发现，有几个类的验证集准确率为 0，因此认为数据中不同类别的难易有差别，同时也发现不同类的训练样本存在偏差，于是尝试了使用 `Focal Loss` 关注困难样本，提升在某些困难的类上的准确率。`Focal Loss` 的公式如下：

$$FL(p) = -(1 - p)^{\gamma} \log p$$

为了确定较优的  $\gamma$  值，在其他参数相同的情况下进行了对比实验，结果如下：

$\gamma$	验证集准确率
0(CEL)	54.01%
0.3	54.74%
0.5	54.74%
1	55.36%
2	60.58%

可以看出，使用了 Focal Loss 后，在相同条件下，验证集准确率得到了明显的提升，同时观察上文实验结果一节中的混淆矩阵也可发现，不再有准确率为 0 的分类了，确实证明了在本任务中 Focal Loss 有效地使网络学到了某些困难样本（类别）的特征。

### Early Stopping

因为发现在验证集上准确率的最大值往往不是在训练完 150 个 epoch 后取到，因此想到可以通过 Early Stopping 技术提前停止训练，以获得最优的网络。

以泛化损失(GL)作为 Early Stop 的标准，GL 定义如下：

令  $E_{opt}(t) = \min_{t' \leq t} E_{valid}(t')$ ，即当前最低的 Loss 值，则有

$$GL(t) = 100 \cdot \left( \frac{E_{valid}(t)}{E_{opt}(t)} - 1 \right)$$

将 GL 的阈值设为  $\alpha$ ，进行了实验调整  $\alpha$  的取值：

$\alpha$	epoch
(No early stop)	100
3	100
2	100
1	5
0.5	2

从结果看来， $\alpha$  的值并不好调整，本身 Validation Set 上的 loss 波动就很大，因此一旦  $\alpha$  设置得稍大，Early Stop 就几乎不会起作用；而一旦设的稍小，就会过早地停止，远没有达到最优。

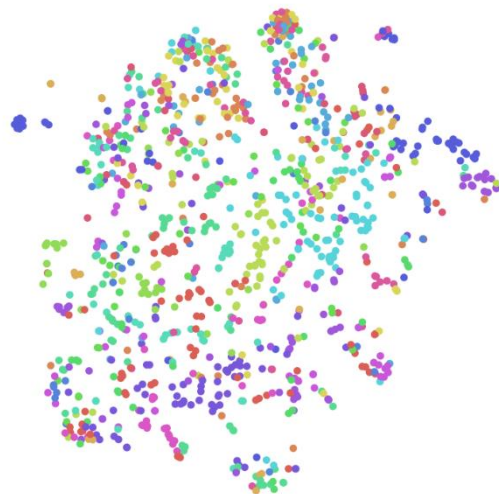
#### d. 可视化结果

##### t-SNE 可视化全连接层前的 feature

在验证集上的 512 维 feature 可视化结果:



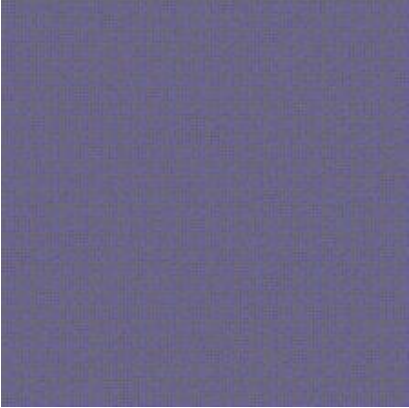
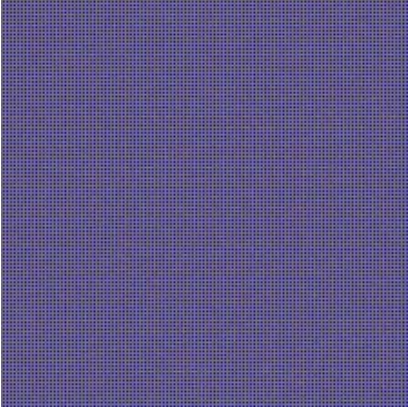
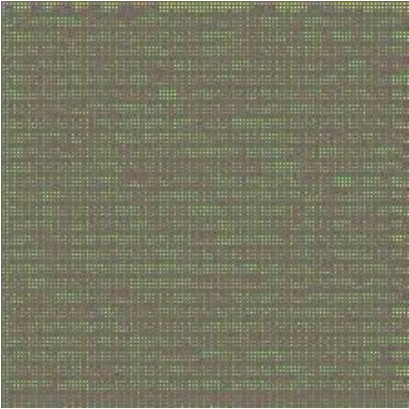
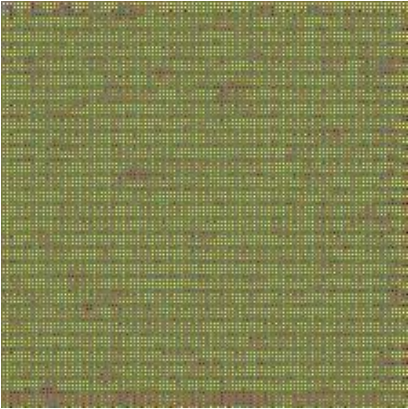
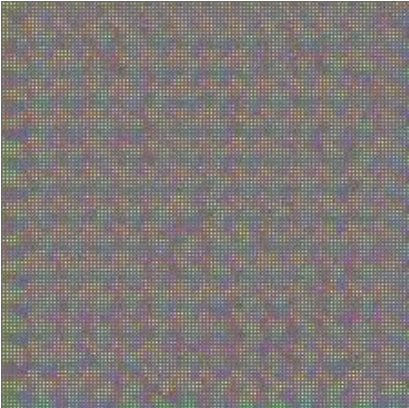
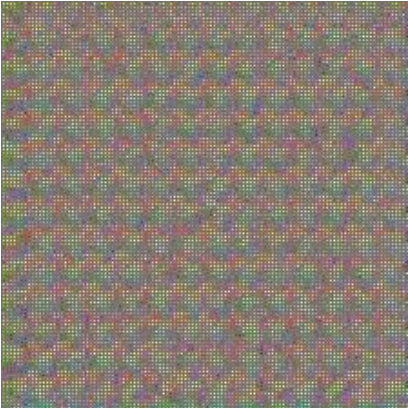
在训练集上的 512 维 feature 可视化结果:



可以看出在验证集上，确实有部分同颜色的点聚集在一起，但是绝大部分点的分布还是较为分散的，组间距离相比组内距离没有明显变大。

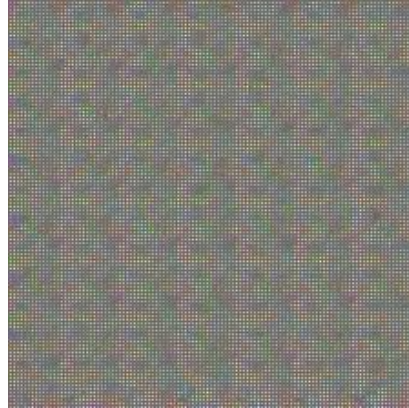
卷积层的可视化

通过输入随机噪声，经过指定卷积层后反向传播优化后，得到的 CNN filter 的可视化图像如下：

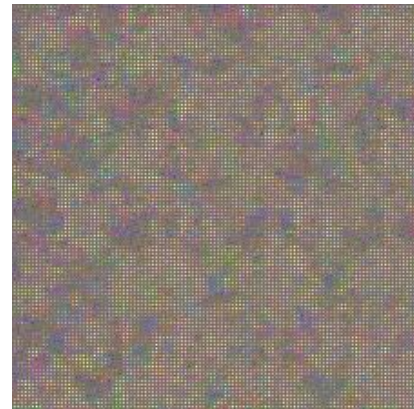
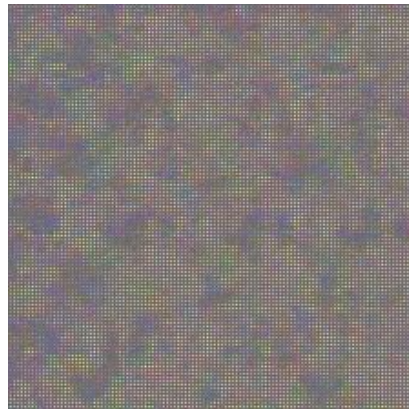
Filter	15 次迭代图像	30 次迭代图像
Block1 conv1 filter0(1*1conv)		
Block2 conv2 filter16(3*3conv)		
Block3 conv3 filter50(1*1conv)		



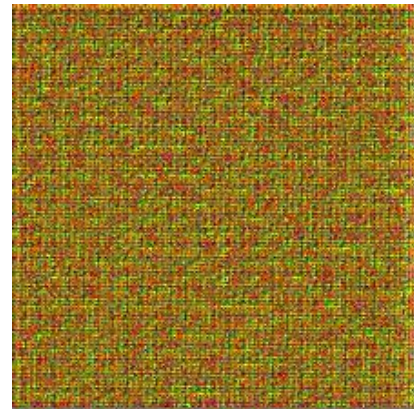
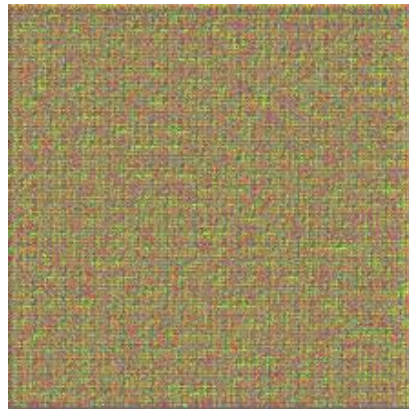
Block4 conv1  
filter9(1\*1conv)



Block5 conv2  
filter36(3\*3conv)



Block6 conv3  
filter48(1\*1conv)



## 4. References

1. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778, 2016.
2. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages 1–9, 2015.

3. A. Shrivastava et al. Focal Loss for Dense Object Detection. ICCV 2017.

4. Utku Ozbulak. PyTorch CNN Visualizations.

<https://github.com/utkuozbulak/pytorch-cnn-visualizations>