

<https://github.com/HatcherRobotics/intelligent-information-processing>.

本人对全部代码与训练结果负责，如有雷同即为抄袭。{jupyter notebook}里为训练的过程，{python}里为前向推理的过程。

## 数据集准备

使用OpenCV读取图片并进行双线性插值调整尺寸为(32X32)，归一化其像素值,再将其转换为batch\_size \* channel \* width \* height的张量格式。

同时增加了可选择的数据增强模块，包括直方图均衡化与镜像翻转。

```
1 def dataset_x(path, data_augmentaion=False):
2     pics = os.listdir(path)
3     pics_list=[]
4     if(data_augmentaion==True):
5         for pic in pics:
6             img = cv.imread(path+'/'+pic)
7             #img = cv.resize(img, (32, 32))
8             img = np.array(img)
9             img = img/255
10            flip_img = np.array(cv.flip(img, 1))
11            flip_img=flip_img/255
12            equ_img =
cv.equalizeHist(cv.cvtColor(img, cv.COLOR_BGR2GRAY))
13            equ_img=np.array(cv.cvtColor(equ_img, cv.COLOR_GRAY2BGR))
14            equ_img=equ_img/255
15            pics_list.append(img)
16            pics_list.append(flip_img)
17            pics_list.append(equ_img)
18        else:
19            for pic in pics:
20                img = cv.imread(path+'/'+pic)
21                img = cv.resize(img, (32, 32))
22                img = np.array(img)
23                img = img/255
24                pics_list.append(img)
25            x = torch.Tensor(np.array(pics_list)).permute(0, 3, 1, 2)
26            return x
```

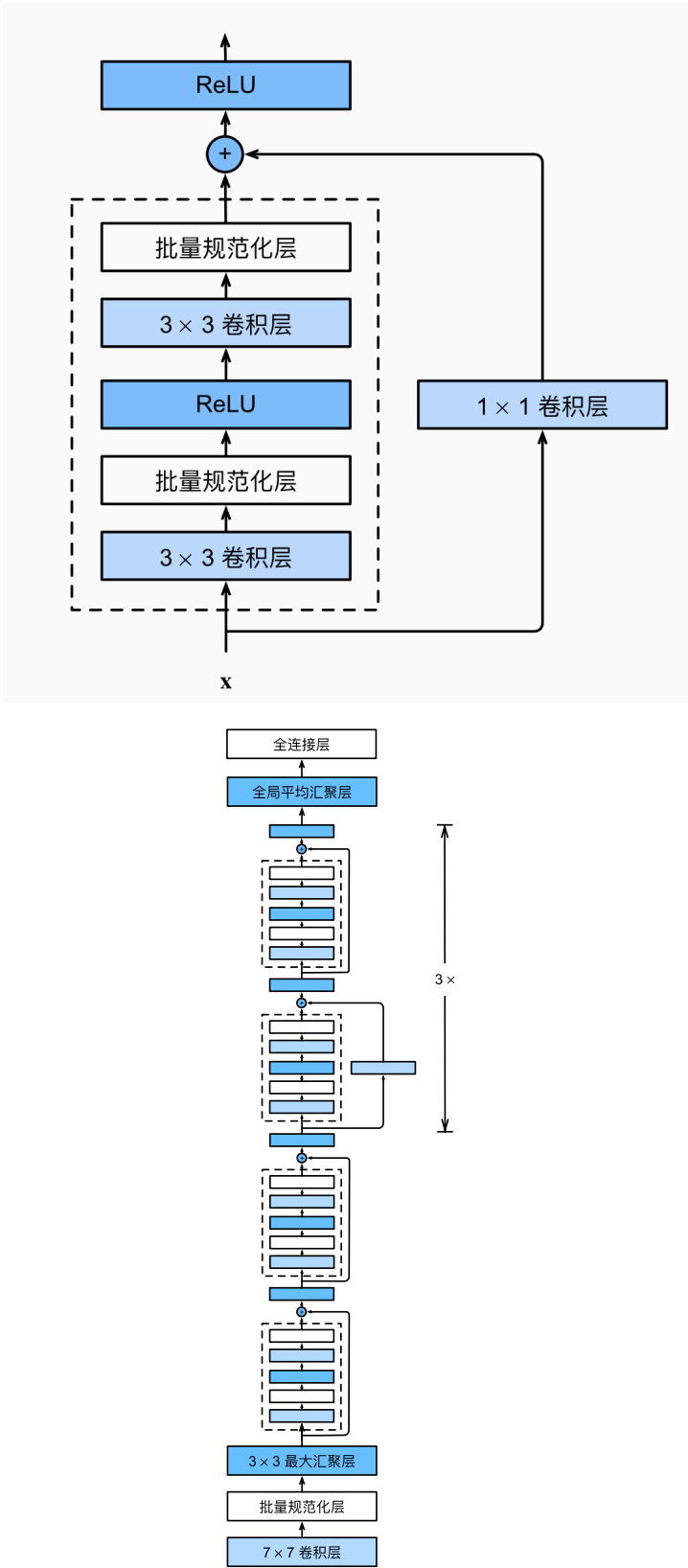
分别制作出训练集，验证集以及测试集并配置DataLoader。

## 设计思路

首先确定引入残差结构，为了保证超高准确率以ResNet18为基础进行改进，在残差块中对输入特征图增加1X1卷积，既能通过卷积核方便调整一致输出特征图数量，又能聚合不同特征图的特征。

## 网络结构

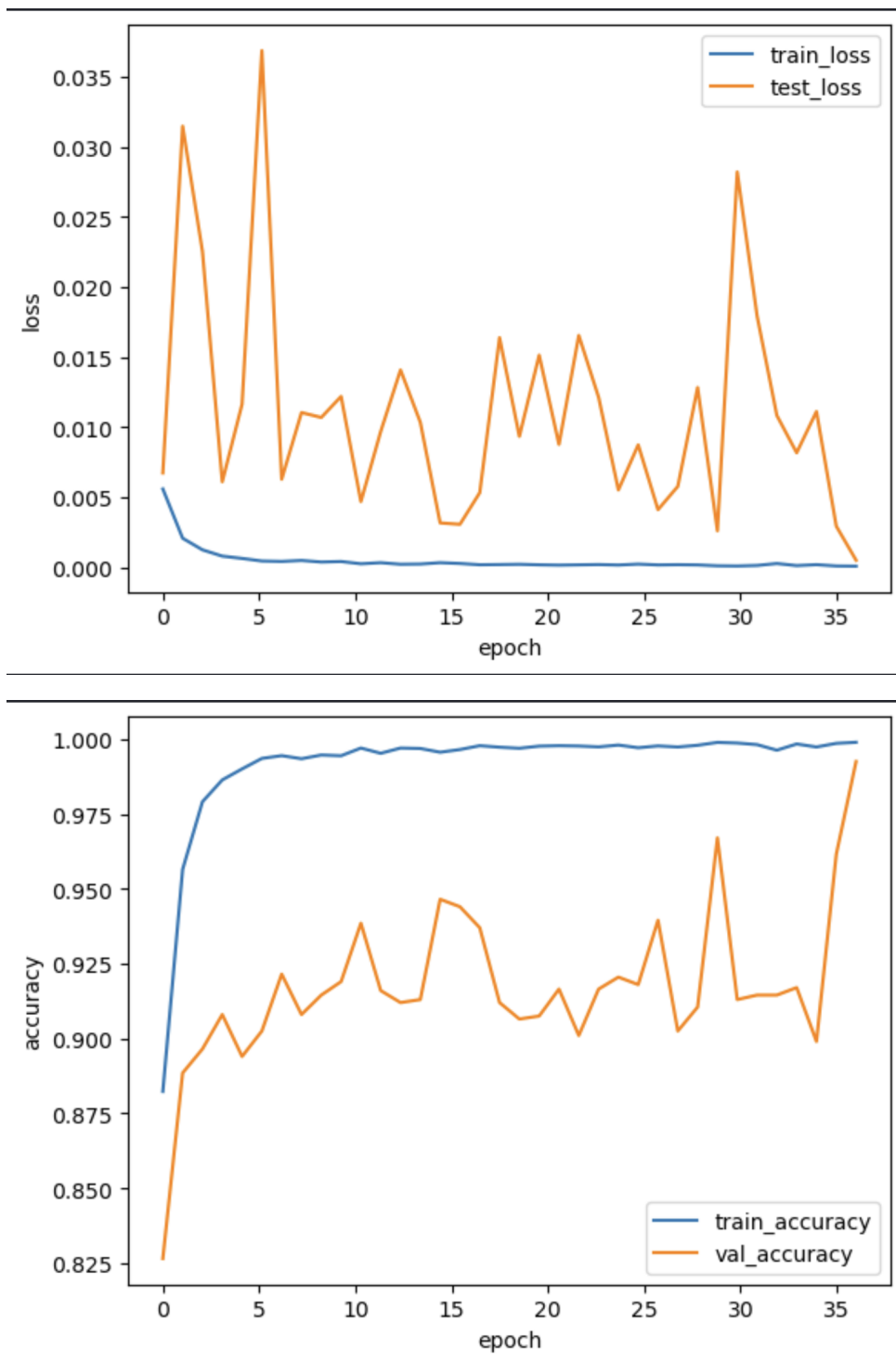
网络可分为两部分，第一部分为堆叠8个残差块的特征提取器，第二部分为全连接层组成的三分类器，最后接入Softmax输出每个分类的类别。使用BatchNormalization防止过拟合。



## 训练参数

优化器选择Adam，学习率设置为0.01，损失函数选择交叉熵损失，权值衰减为 $1e-5$ ，对模型的参数初始化满足正态分布。

## 识别准确率



训练与验证的损失与准确率如上图所示，迭代至39轮验证集上的准确率达到99%，保存权重为"resnet.pth"。

在测试集上进行前向传播，准确率也在99%以上。

```
test_acc,_ = evaluate_accuracy(test_iter,net,loss)
print("在测试集上的准确率为:%.3f"%(test_acc))
```

[42]

... 在测试集上的准确率为:0.993

## 模型参数数量、执行时的内存占用量、计算量

参数量为11,530,499，占43.99MB，申请的显存量为44.05MB，计算量为37.06 GMac，到调用函数为止所达到的最大的显存占用为728.09 MB。

```
-----
Total params: 11,530,499
Trainable params: 11,530,499
Non-trainable params: 0
-----
Input size (MB): 0.75
Forward/backward pass size (MB): 1056.00
Params size (MB): 43.99
Estimated Total Size (MB): 1100.74
-----
Warning: module ResidualBlock is treated as a zero-op.
Warning: module ResNet is treated as a zero-op.
Computational complexity:      37.06 GMac
Number of parameters:          11.53 M
显存占用量为:  44.05126953125 MB
到调用函数为止所达到的最大的显存占用字节数: 728.09130859375 MB
```