Python 第二次作业 实验报告

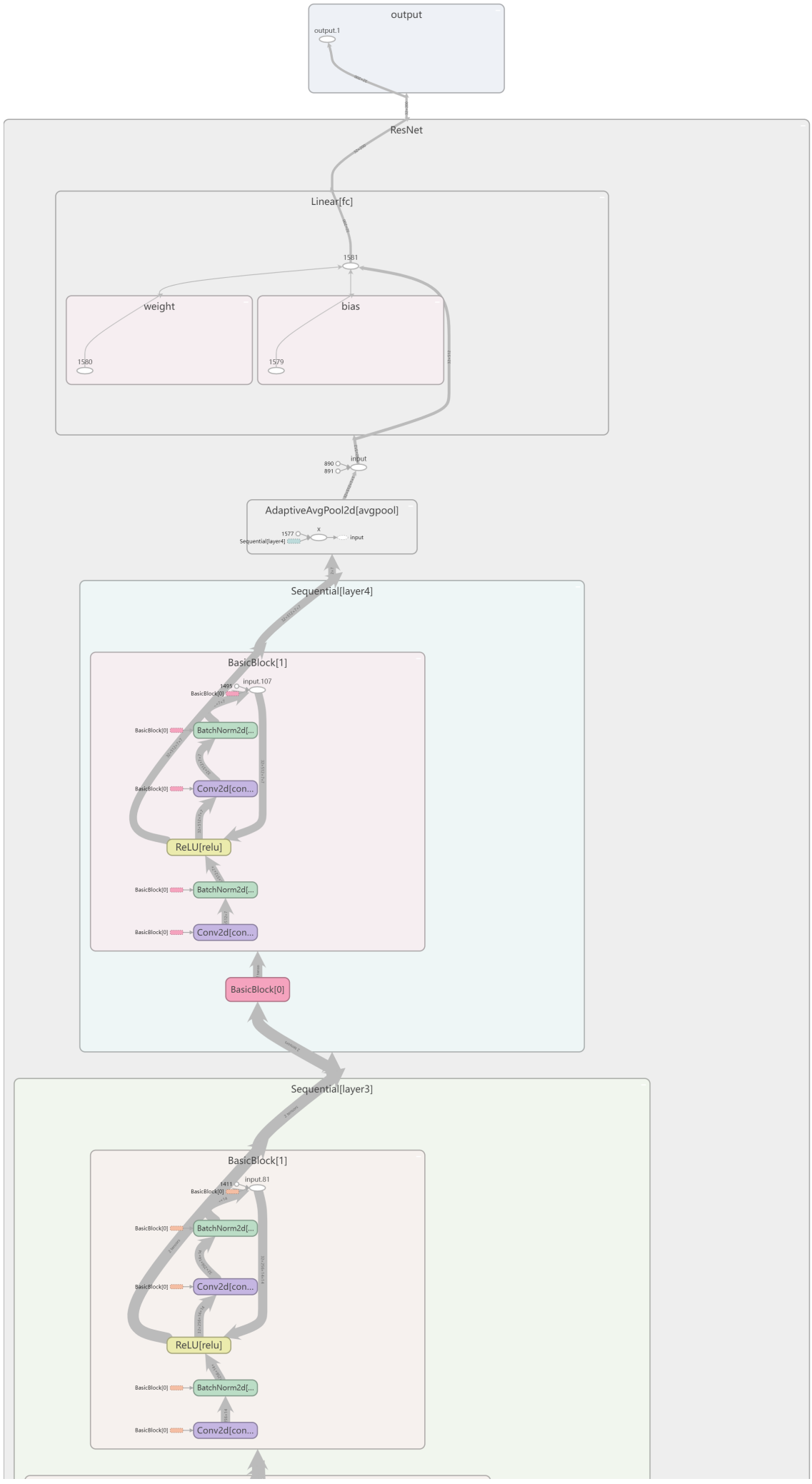
PB19000314 金小龙

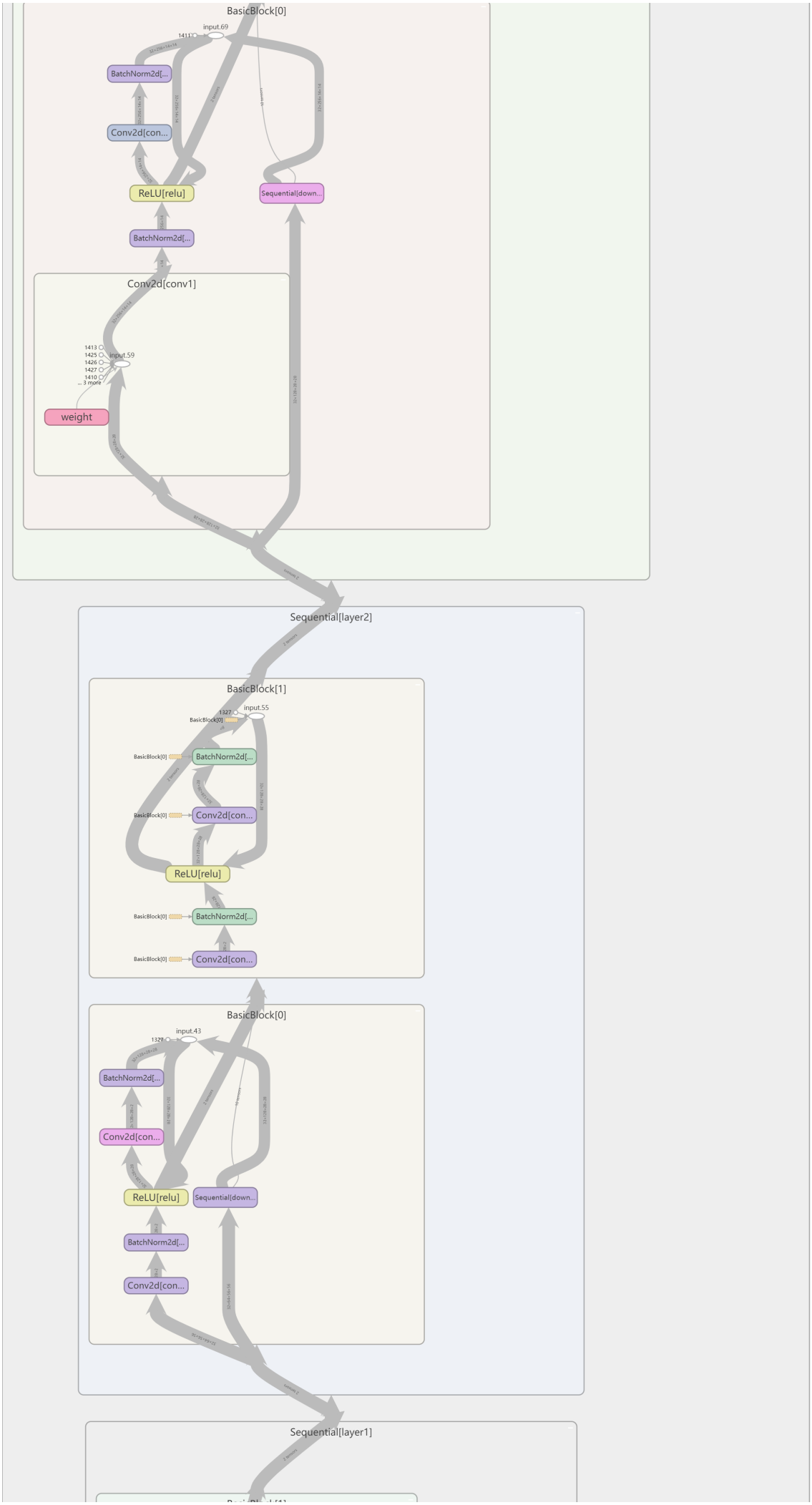
评分细则

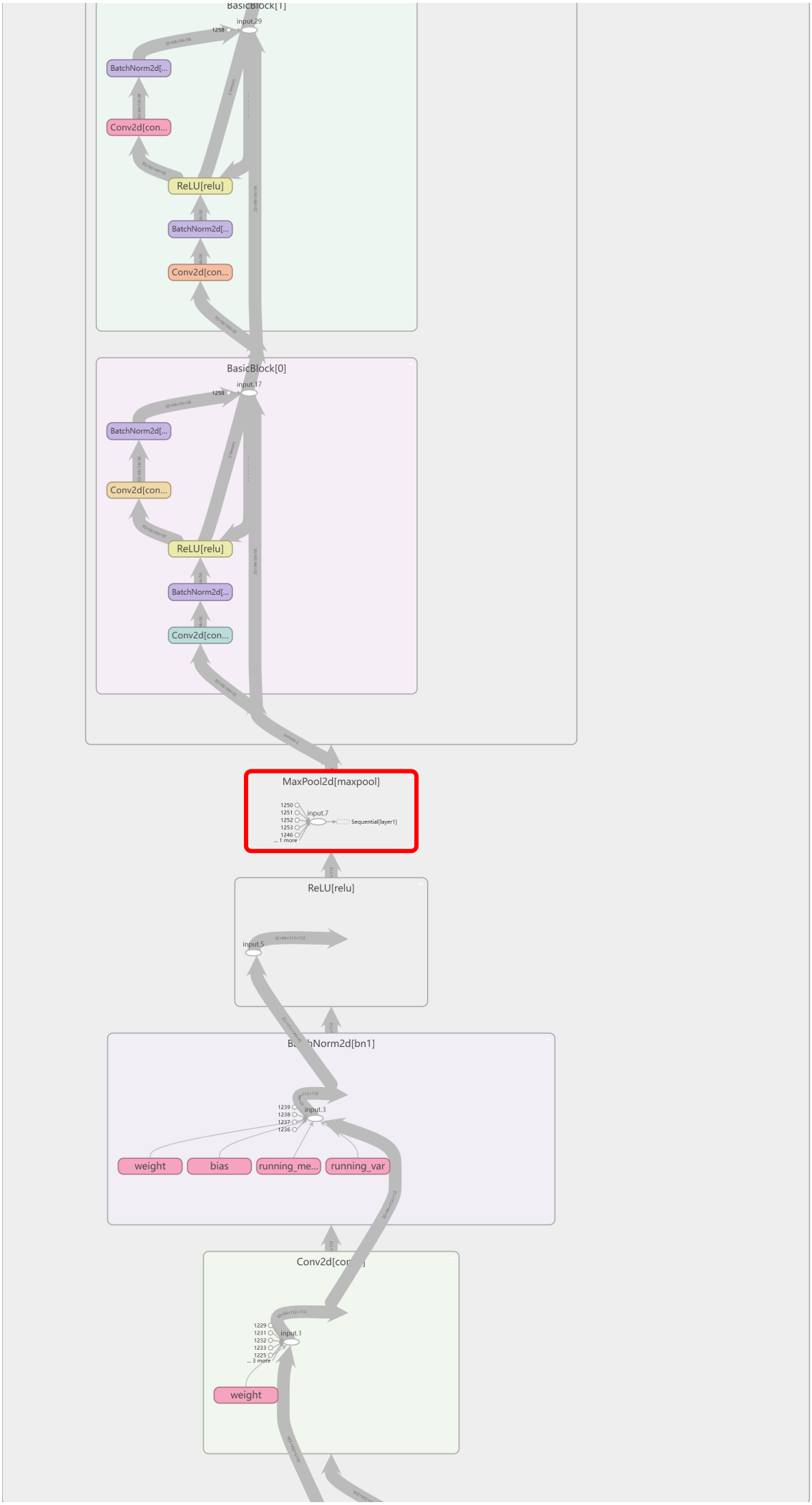
1.各层名称及输出大小截图

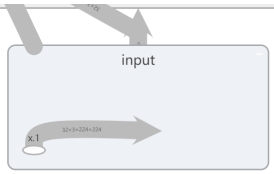
细节可以运行代码，在graph里看到

```
print('打印模型结构')
dataiter = iter(train_loader)
images, labels = dataiter.next()
images = images.to('cuda')
writer.add_graph(model, images)
print('打印完成')
```


output
output.1
ResNet
Linear[fc]
1581
weight
bias
1580
1579
input
890
891
AdaptiveAvgPool2d[avgpool]
1577
Sequential[layer4]
input
Sequential[layer4]
BasicBlock[1]
input.107
1495
BasicBlock[0]
BatchNorm2d[...
Conv2d[con...
ReLU[relu]
BasicBlock[0]
Sequential[layer3]
BasicBlock[1]
input.81
1411


BasicBlock[0]
input.69
1411
BatchNorm2d[...
Conv2d[con...
ReLU[relu]
Sequential[down...
BatchNorm2d[...
Conv2d[conv1]
1413
1425
1426
1427
1410
... 3 more
input.59
weight
Sequential[layer2]
BasicBlock[1]
1327
input.55
BasicBlock[0]
BatchNorm2d[...
Conv2d[con...
ReLU[relu]
BatchNorm2d[...
Conv2d[con...
BasicBlock[0]
input.43
1327
BatchNorm2d[...
Conv2d[con...
ReLU[relu]
Sequential[down...
BatchNorm2d[...
Conv2d[con...
Sequential[layer1]


BasicBlock[1]
input.29
1258
BatchNorm2d[...
Conv2d[con...
ReLU[relu]
BatchNorm2d[...
Conv2d[con...
BasicBlock[0]
input.17
1258
BatchNorm2d[...
Conv2d[con...
ReLU[relu]
BatchNorm2d[...
Conv2d[con...
MaxPool2d[maxpool]
1250
1251
1252
1253
1246
... 1 more
input.7
Sequential[layer1]
ReLU[relu]
input.5
BatchNorm2d[bn1]
1239
1238
1237
1236
input.3
weight
bias
running_me...
running_var
Conv2d[con...
1229
1231
1232
1233
1225
... 3 more
input.1
weight



2.修改output维数

增加参数`num_classes = 200`

```
# create model
if args.pretrained:
    # 预训练模型
    print("=> using pre-trained model
'{}'".format(args.arch))
    model = models.__dict__[args.arch](pretrained=True)
    else:
        # 直接使用模型
        print("=> creating model '{}'".format(args.arch))
        model = models.__dict__[args.arch](num_classes = 200)
```

3.修改数据集

编写脚本将验证集的数据目录结构变 更为与训练集一致。

解决思路

> 1. val_annotations.txt 获取对应编号和种类
> 2. 读取`images`文件夹中文件，对应val_annotations.txt创建文件夹并移入
> 3. 删除多余文件

```
import io
import pandas as pd
import glob
import os
from shutil import move
from os.path import join
from os import listdir, rmdir

target_folder = './tiny-imagenet-200/val/'

val_dict = {}
```

```
# 获得对应关系
with open(target_folder + 'val_annotations.txt', 'r') as f:
    for line in f.readlines():
        split_line = line.split('\t')
        val_dict[split_line[0]] = split_line[1]

# 找出路径列表
paths = glob.glob(target_folder + 'images/*')
# print(paths[0].split('/'))
# print('paths[0].split('')[-1].split('')[-1]:')
# print(paths[0].split('/')[-1].split('\\')[-1])

for path in paths:
    file = path.split('/')[-1].split('\\')[-1]
    folder = val_dict[file]
    if not os.path.exists(target_folder + str(folder)):
        os.mkdir(target_folder + str(folder))

for path in paths:
    file = path.split('/')[-1].split('\\')[-1]
    folder = val_dict[file]
    dest = target_folder + str(folder) + '/' + str(file)
    move(path, dest)

# 删除多余文件
os.remove('./tiny-imagenet-200/val/val_annotations.txt')
rmdir('./tiny-imagenet-200/val/images')

print('over')
```

4.训练

曲线变化趋势

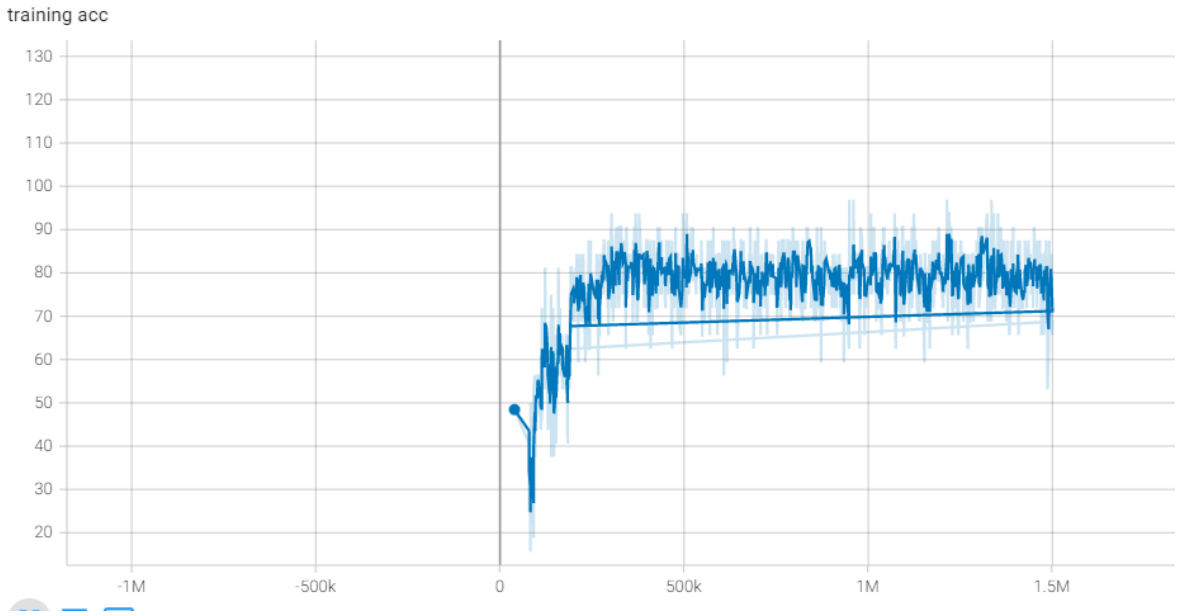
模型1（保留对图像的裁切）

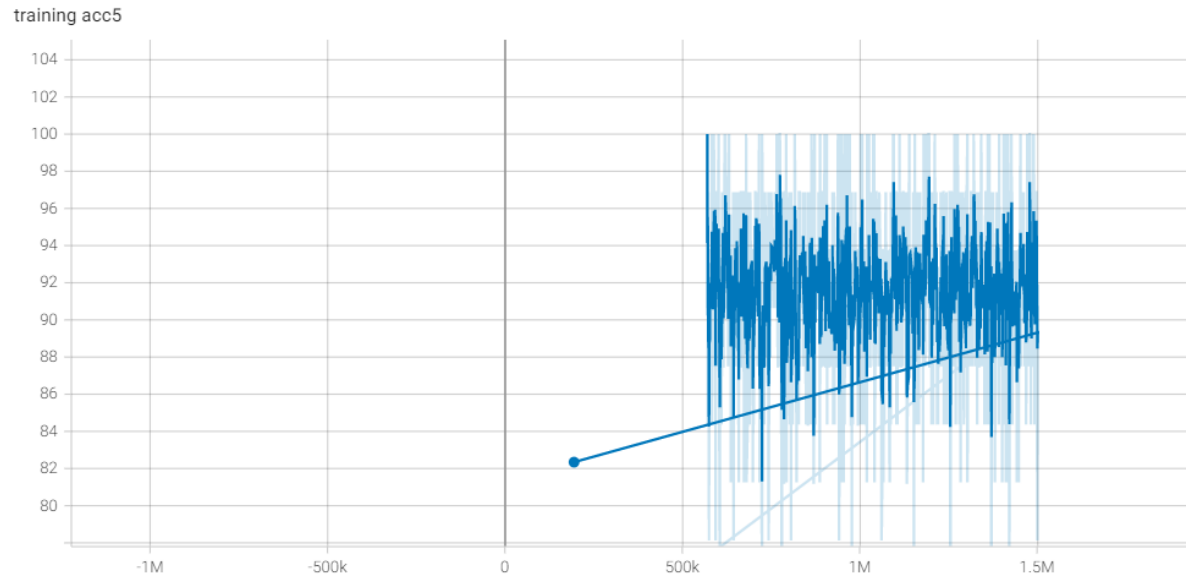
> 模型训练损失在初始阶段下降迅速，但后期趋于0速度放缓，模型开始收敛。但最终loss离0有一点差距，可以考虑调整模型结构。
>
> 模型准确率在初始阶段上升迅速，但后期增长速度放缓，模型开始收敛。但最终离100有一点差距，可以考虑调整模型结构。

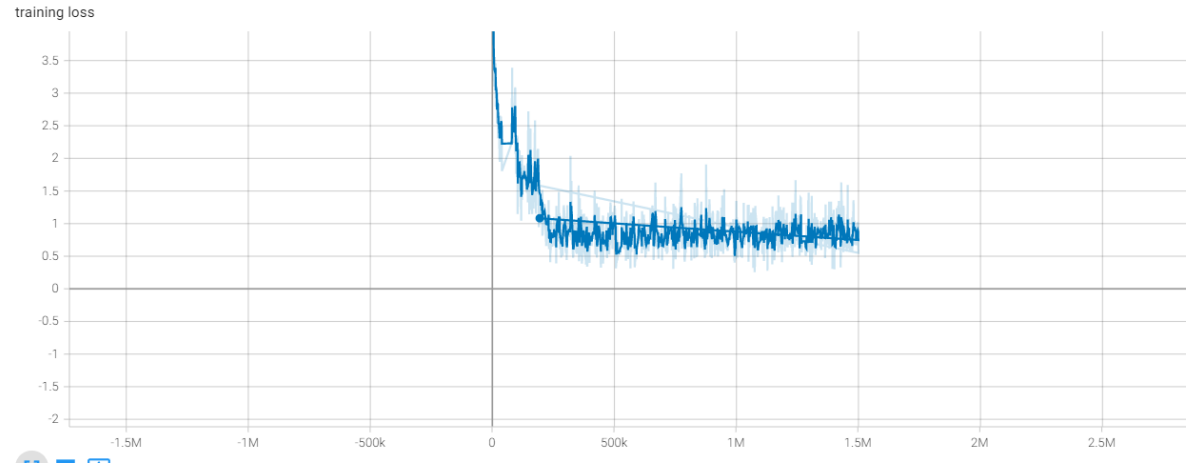
```
# 在 TensorBoard 中 观察训练集 Loss、训练集精度
```

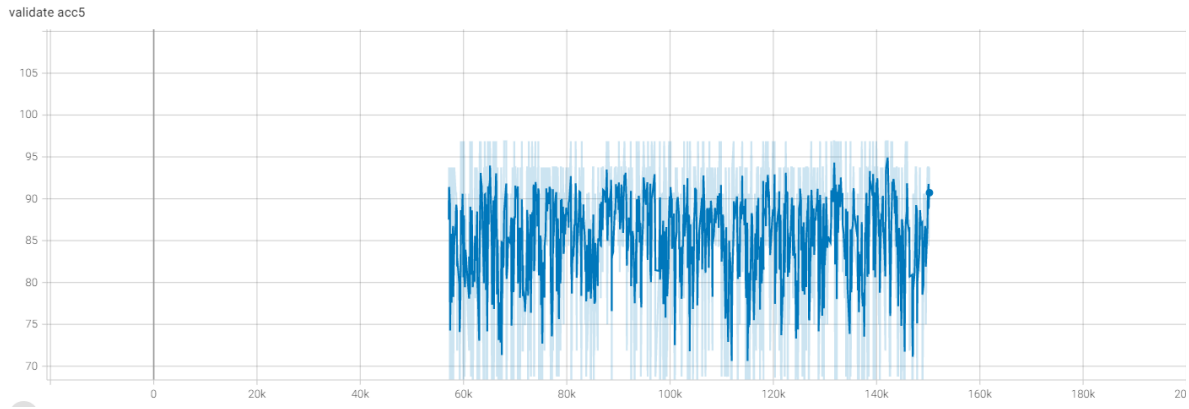
```
writer.add_scalar('training loss',
                              losses.val ,
                              epoch * len(train_loader) + i)
writer.add_scalar('training acc',
                  acc1 ,
                  epoch * len(train_loader) + i)
writer.add_scalar('training acc5',
                  acc5,
                  epoch * len(train_loader) + i)


# 在 TensorBoard 中 观察验证集 Loss、验证集精度
writer.add_scalar('validate loss',
                  losses.val,
                  epoch * len(val_loader) + i)
writer.add_scalar('validate acc',
                  acc1,
                  epoch * len(val_loader) + i)
writer.add_scalar('validate acc5',
                  acc5,
                  epoch * len(val_loader) + i)
```





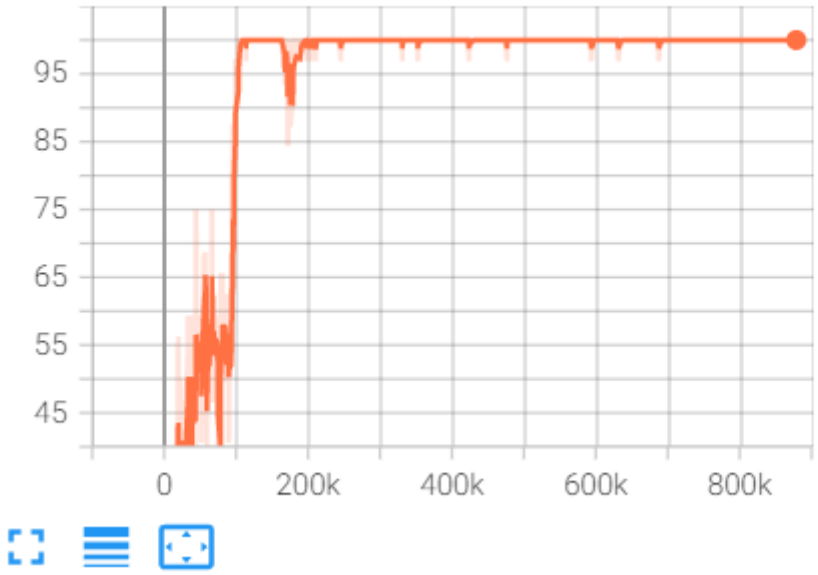




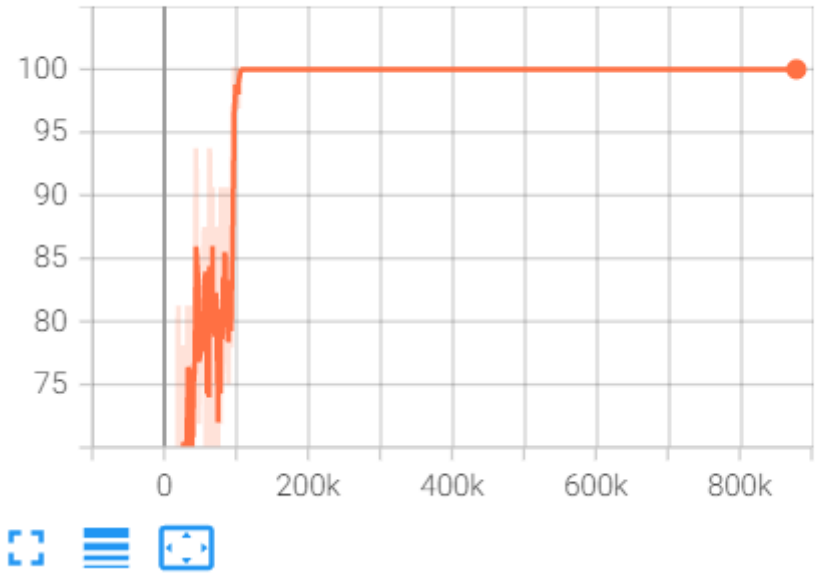
模型2（去除对图像的裁切）

> 模型训练损失在初始阶段下降迅速，后期趋于0，模型收敛效果较好。
>
> 模型准确率在初始阶段上升迅速，后期稳定100，模型收敛。
>
> 但是考察验证集的表现，发现模型可能存在过拟合情况。

training acc

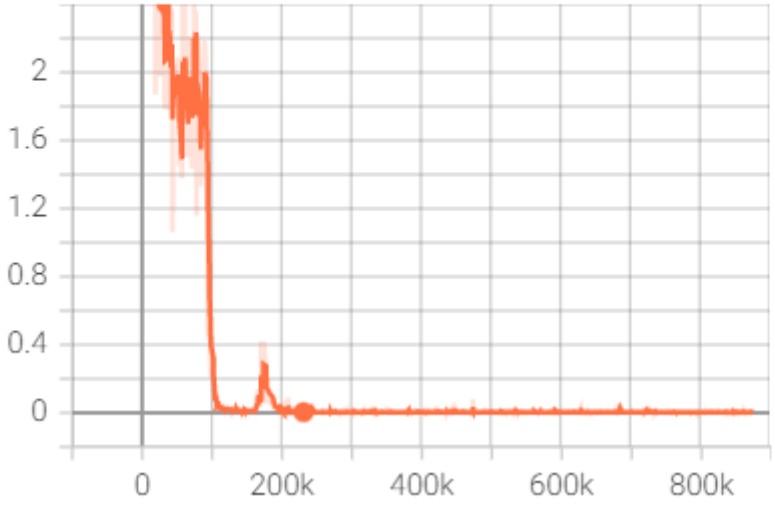


training acc5

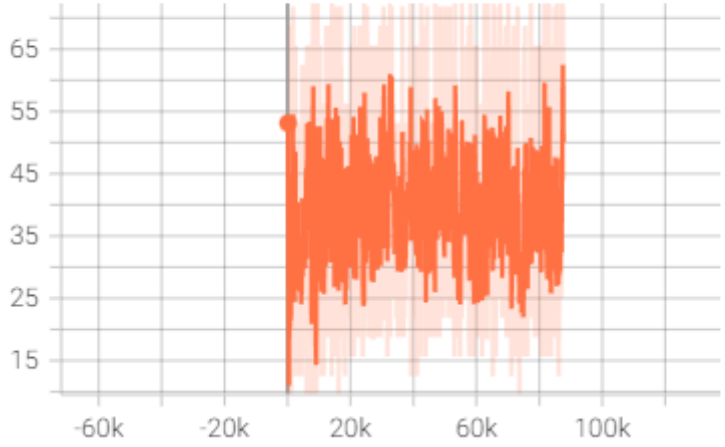


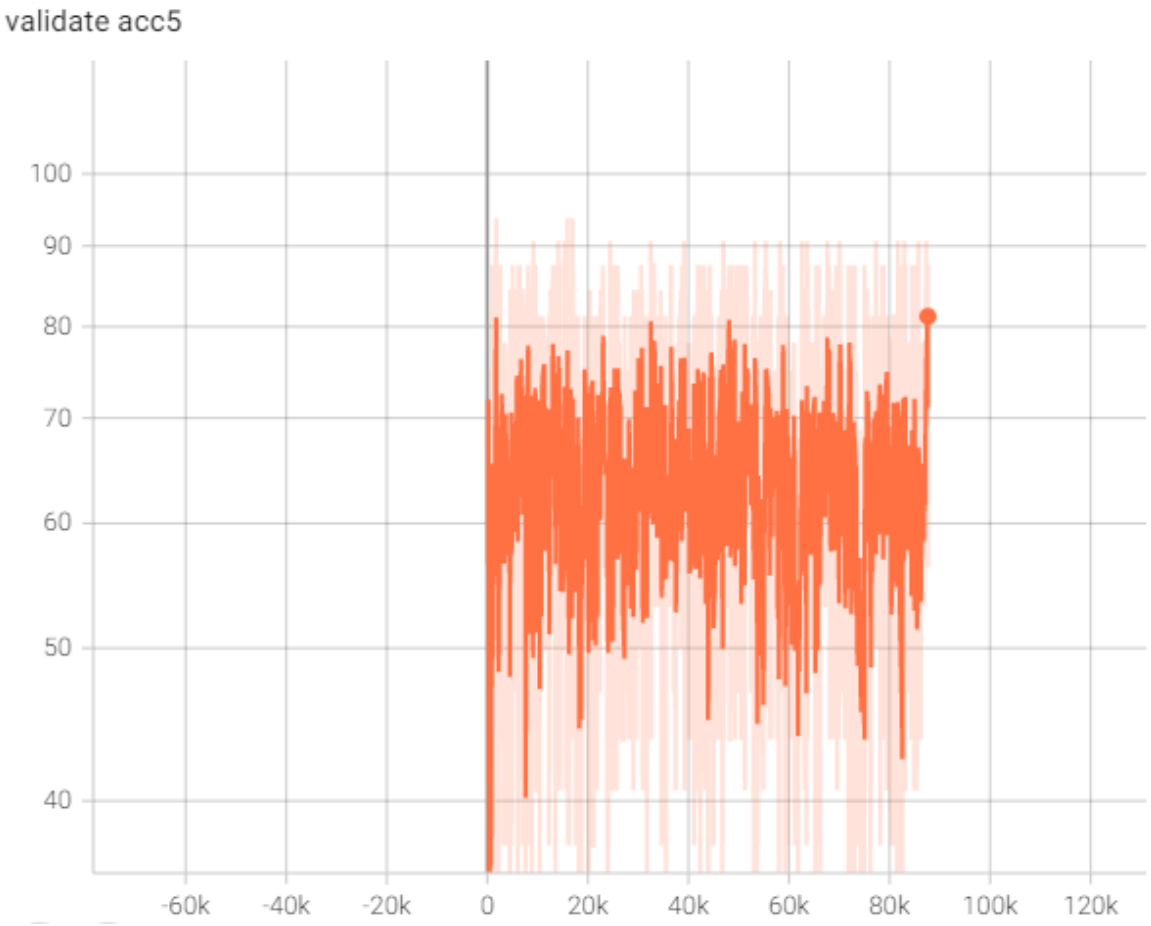
training loss

training loss



validate acc



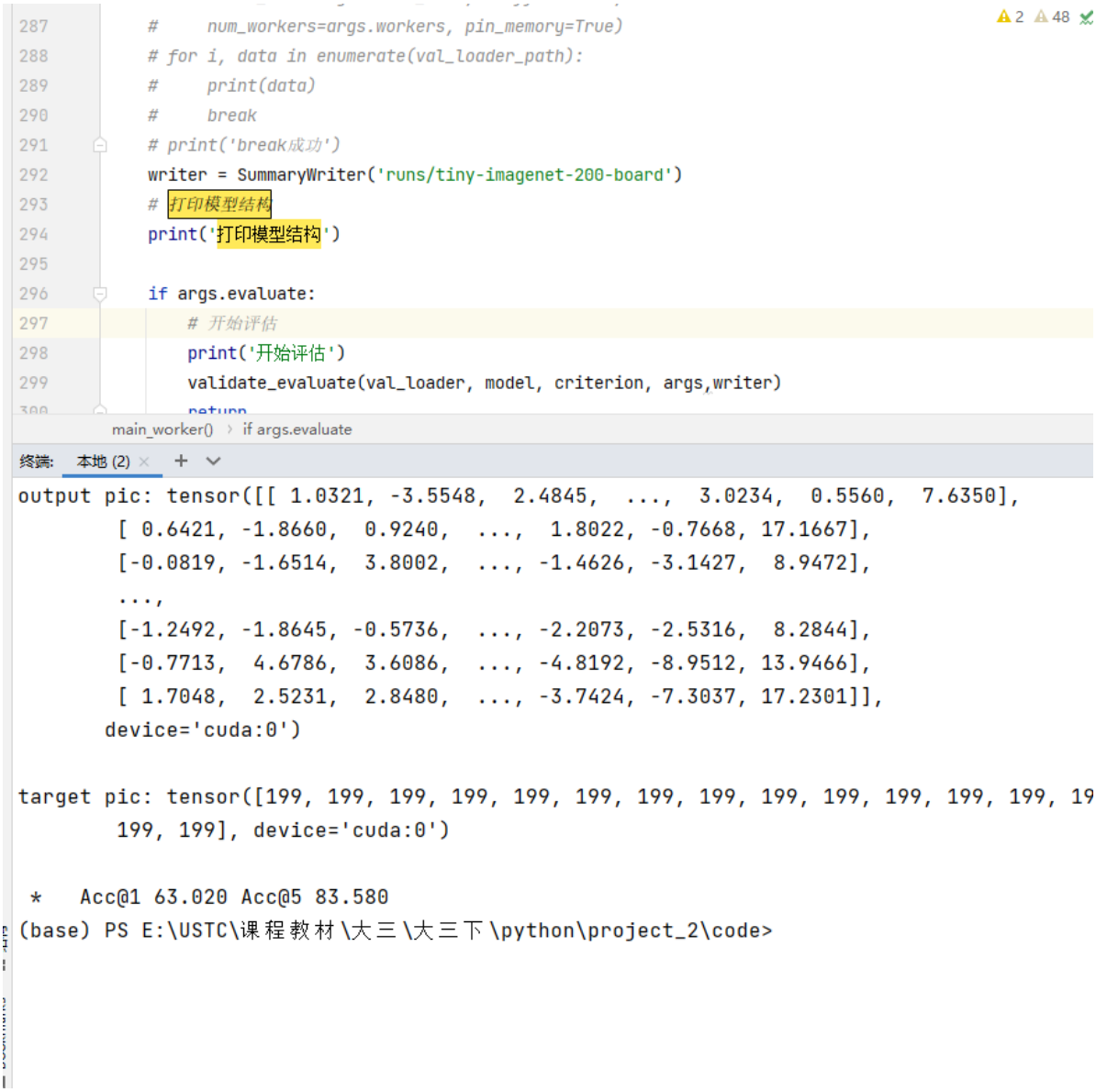


5.模型评估

模型1（保留对图像的裁切）

checkpoint.pth.tar

> python main.py tiny-imagenet-200 --gpu 0 --resume checkpoint.pth.tar -e
>
> 结果 Acc@1 63.020 Acc@5 83.580



checkpoint1.pth.tar

> python main.py tiny-imagenet-200 --gpu 0 --resume checkpoint1.pth.tar -e
>
> Acc@1 62.940 Acc@5 83.480

开始找错

```
outputsize: torch.Size([16, 200])
output pic: tensor([[ 0.9757, -3.3310,  2.0873,  ...,  2.5932,  0.4226,  7.5247],
        [ 0.7568, -2.0121,  0.4857,  ...,  1.2087, -0.8026, 17.0241],
        [-0.3346, -1.5511,  3.7180,  ..., -1.8467, -3.2398,  9.0691],
        ...,
        [-1.1295, -1.5533, -0.7370,  ..., -2.4130, -2.6574,  7.7317],
        [-1.0845,  5.3267,  3.0983,  ..., -5.1179, -9.4422, 13.7208],
        [ 1.8000,  2.5775,  2.6741,  ..., -3.7649, -7.3561, 16.9411]],
       device='cuda:0')

target pic: tensor([199, 199, 199, 199, 199, 199, 199, 199, 199, 199, 199, 199, 199, 199
        199, 199], device='cuda:0')

 *   Acc@1 62.940 Acc@5 83.480
```

由于两次断点都是在训练后期，所以验证测试差距不大。