

在Tiny-ImageNet数据集上训练Resnet模型

PB19000050 邓一川

运行环境

使用单卡 RTX2060 GPU 训练。

安装需求

在项目根目录下执行以下命令：

```
pip install -r requirements.txt
```

数据预处理

数据集采用 Tiny-ImageNet，可以在这里[下载](#)，因为数据集过大，不在仓库中展示。针对 Tiny-ImageNet 验证数据集中出现的标签与图片不匹配的情况，实验中运行了一个简单的预处理程序，依照 wnids.txt 和 val/val_annotations 进行标签的重新匹配，并且按照训练数据集的模式重新生成了 val 文件夹。在项目根目录下执行：

```
python process_validation.py
```

训练

训练任务针对 200 维的 Tiny-ImageNet 的数据，所以针对 resnet18 作了轻微的改动，修改了输出线性层的维度：

```
# modify the output dimension of pretrained model
if args.arch == "resnet18":
    num_fts = model.fc.in_features
    model.fc = nn.Linear(num_fts, args.num_classes)
```

训练需要在根目录下执行：

```
# add --tensorboard to launch the recording summarywriter
python main.py --tensorboard
```

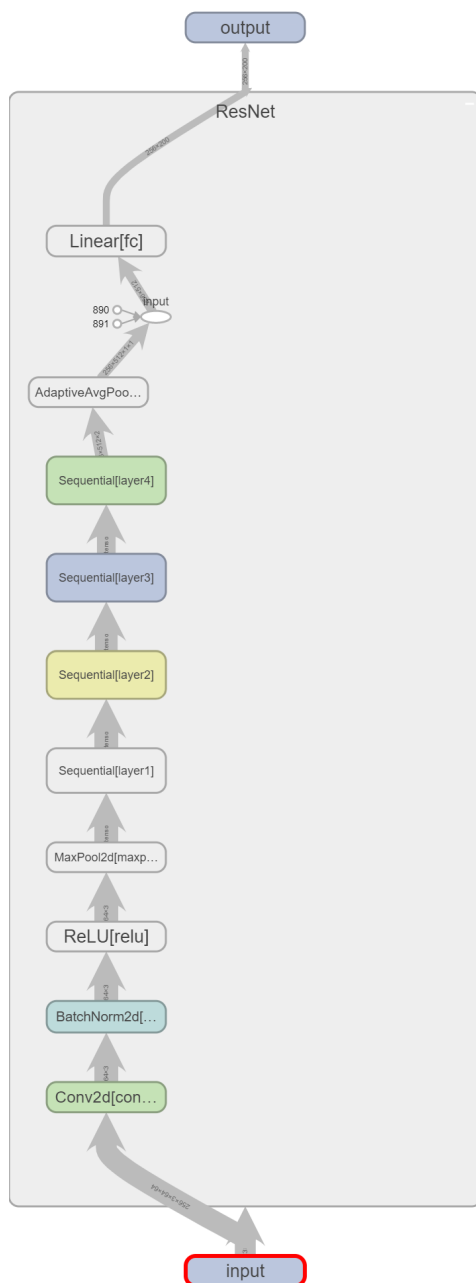
推理

推理默认在验证集上，在根目录执行以下命令：

```
python main.py --evaluate --resume your_path_to_checkpoint
```

模型结构

在正式开始训练前，实验使用了一些随机输入，在 `tensorboard` 中绘制 `resnet18` 的结构，如下图所示：

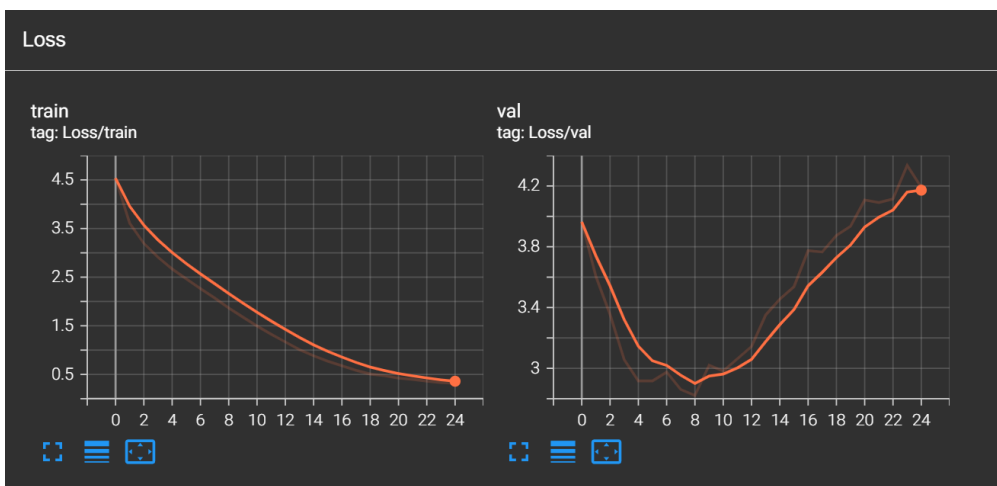
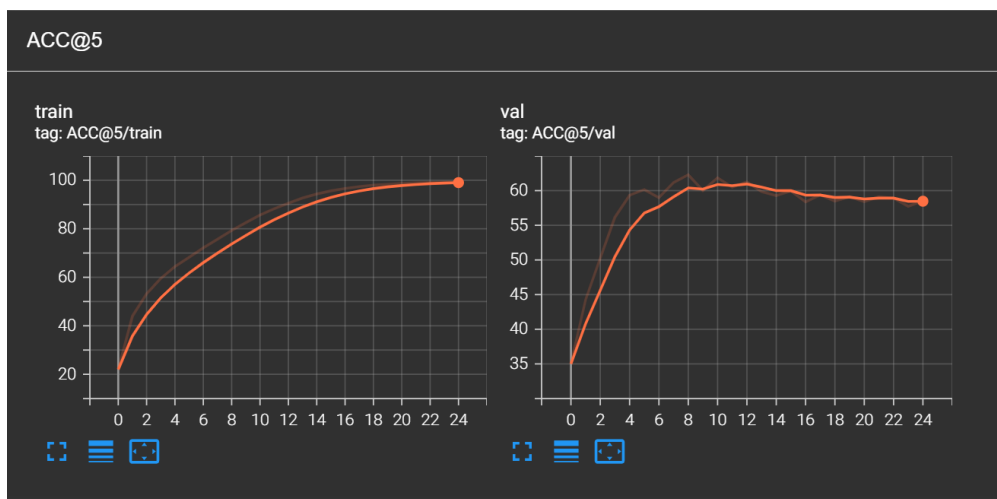


训练过程

在训练的过程中，启用 `tensorboard` 之后，训练过程中的所有数据会保存在 `runs/` 目录下（资源较大，没有上传）。然后在命令行中执行以下命令，在浏览器中查看运行过程图像：

```
tensorboard --logdir=runs
```

本次实验中过程图像如下：



分析实验过程中的变化情况，可以明显发现的是，在训练集上，最终 top5 的准确率是收敛于 100% 的，loss 也在不断下降。这是符合预期的。在验证集上，明显可以看到一个过拟合的趋势，准确率在第 10 个 epoch 左右达到了峰值，大约 60% 左右，之后就开始了下降。验证集上的损失也可以反映过拟合的趋势，在第 10 个 epoch 之后，损失函数的值不减反增。

评判结果











checkpoint.pth.tar	2022/6/27 21:05
checkpoint.pth.tar_pred_list.txt	2022/7/7 21:18
main.py	2022/7/7 21:15
model_best.pth.tar	2022/6/27 20:25

实验选取了两次 latest 和 best 的 checkpoint，按照 validation 数据输入的顺序 (shuffle=False, sampler=None) 依次打印出了预测结果，呈现在仓库根目录下的txt文件中。同时，测试的最终结果如下：

```
# last_checkpoint
Acc@1 33.320 Acc@5 58.510

# best_checkpoint
Acc@1 36.490 Acc@5 62.690
```

并且我们从测试结果中选取了第12张图片(0,18), 第70张图片(1, 58), 第134张图片(2, 178), 第208张图片(4, 19), 第568张图片(3, 10), 第756张图片(16, 198), 第884张图片(17, 30), 第7275张图片(164, 133), 第9086张图片(181, 188), 第9985张图片(152, 199), 这10张图片在两个模型中的评测结果不同，**括号内前一个为 best_model 的分类推断，后一个为 last_model 的分类推断。**

ID	12	70	134	208	568
					
ID	756	884	7275	9086	9985
					

致谢

本实验代码基于 `pytorch` 开源仓库[样例](#)。