

1. 学习内容

1) 论文阅读

(1) 论文名字

Multi-label Image Recognition by Recurrently Discovering Attentional Regions

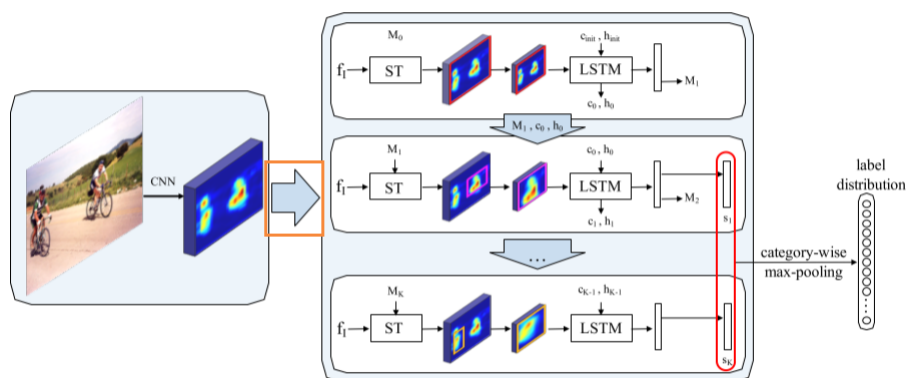
(2) 论文内容简述

论文提出了解决多标签图片识别问题的新模型。

模型结构：CNN+ST+LSTM

- ① 图片经过深度卷积神经网络（VGG-16）提取最后一层特征的响应图。
- ② ST 在每次循环中定位一个 Attention Region
- ③ LSTM 预测该区域的类别分数，同时更新 ST 的参数
- ④ 结合所有循环产生的类别分数，得到最终的类别分布

2) cifar 数据集上 ten crop 实验



本周任务是在 AlexNet 网络下加入 tencrop 机制，在 cifar 数据集上进行实验，观察其对模型性能是否有提升。

因为 cifar 数据集图片尺寸很小，仅 32×32 ，取最后一层卷积层的输出不足以进行 ten crop。

目前我对卷积网络做了一定修改，使得最后一层卷积层输出 feature map 为 $256 \times 5 \times 5$ 。

个人理解：我现在所做的 ten crop 部分即结构中橙色框出的箭头部分。

拟将 AlexNet 最后一层输出的 feature map 剪切翻转为 10 个 $256 \times 3 \times 3$ 的 feature map，依次放入一个只有全连接层的网络 FullNet 输出分类分数，将得到的 10 个向量做平均，得到原 $256 \times 5 \times 5$ feature map 最终的分类分数 outputs。

这里的 FullNet 暂代了 ST 和 LSTM 的分类功能。

反向传播时，计算 outputs 与原图 targets 对应的 loss，用于更新 FullNet 网络参数。

2. 遇到的困难以及解决方案，或者学习收获

1) 遇到的困难以及解决方案

(1) 预训练的卷积网络

我本来是想先训练一个新的最后输出 $256 \times 5 \times 5$ 的完整网络，保存参数用于输出 feature map。

因为这个网络没有输出分类结果，无法做反向传播更新网络参数，它应该是一个已经训练过的，并且在最后的网络中仅作为生成 feature map 部分不参与反向传播。

但是分给我们组的 2, 3 号卡当时看的时候都被占满了, 跑不起来, 就没有训练。

(2) tencrop

先把网络输出的 Variable 转成 numpy, 剪切矩阵并做翻转, 把十个结果作为 FullNet 的输入, 最后把网络的十个输出结果取平均计算 loss, 反向传播只传回 FullNet。

2) 学习收获

基本了解了论文算法。

大致确定任务做法。

3) 下周计划

实现 cifar 数据集上的 tencrop 操作。

不知道现在的想法是否可行, 但目标是先按当前预想完成可运行的代码。

备注 :

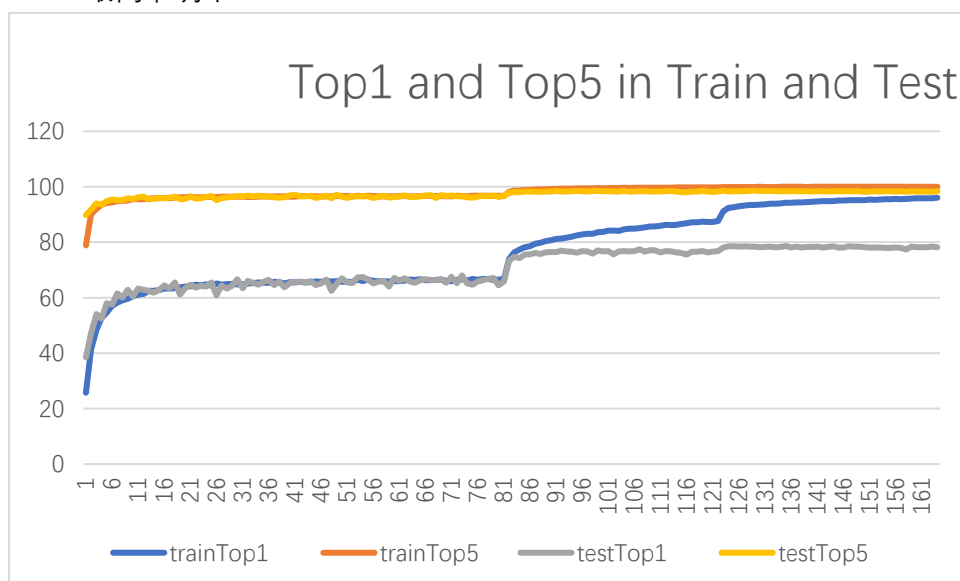
上次提到的关于测试集数据预处理采用和训练集不同方法的建议, 初步实验之后发现的确提高了模型准确率。

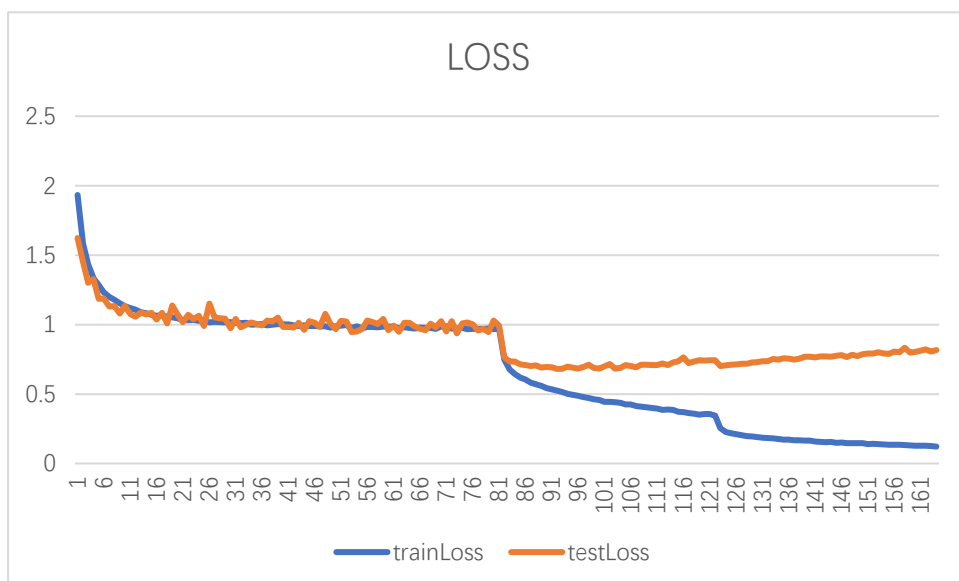
```
#### 数据预处理
# train
transform = transforms.Compose([
    transforms.RandomCrop(32, padding=4), # 随机剪裁
    transforms.RandomHorizontalFlip(), # 随机水平翻转
    transforms.ToTensor(), # 转为tensor
    transforms.Normalize((0.5,0.5,0.5),(0.5,0.5,0.5)), # 归一化
])

# test
transform_test = transforms.Compose([
    transforms.RandomHorizontalFlip(), # 随机水平翻转
    transforms.ToTensor(), # 转为tensor
    transforms.Normalize((0.5,0.5,0.5),(0.5,0.5,0.5)), # 归一化
])
```

测试集取消了随机剪裁。

TOP1 最高准确率 : 78.65





之前训练集和测试集用相同预处理方式：
最好 TOP1 结果：78.2

