

Tinymind多标签比赛内容介绍

参加人员:何晨,李怡新,涂修建,谢伟铭,战柏瑞

环境安装以及文件位置

所需要环境包在requirements.txt里面,需要使用conda安装环境.

```
conda install --yes --file requirements.txt
```

分别将train和test文件夹放在ipynb文件所在文件夹,在当前文件夹运行jupyter notebook既可使用.

本次比赛的亮点

- 预处理图片

#读取图片函数

```
def get_image(img_paths, img_size):
    x = np.zeros((len(img_paths), img_size, img_size, 3), dtype=np.uint8)
    i = 0
    blackIm = Image.new('RGB', (800, 800), 'Black')
    for img_path in img_paths:
        img = Image.open(img_path)
        if img.mode != 'RGB':
            img = img.convert("RGB")
        #平铺图片, 不改变图片比例
        width, height = img.size
        copyIm = blackIm.copy()
        for left in range(0, 800, width):
            for top in range(0, 800, height):
                copyIm.paste(img, (left, top))
        img = copyIm
        img = img.resize((img_size, img_size), Image.LANCZOS) #用LANCZOS插值算法, resize质量高

        arr = np.asarray(img)
        x[i,:,:,:] = arr
        i += 1
    return x

def get_data_batch(X_path, Y, batch_size, img_size):
    while 1:
        for i in range(0, len(X_path), batch_size):
            x = get_image(X_path[i:i+batch_size], img_size)
            y = Y[i:i+batch_size]
            yield x, y #返回生成器
```

在预处理图片时,我们队伍发现处理时运用普通的数据增强几乎没有效果.所以自己进行了图片平铺的处理.由于图片大小height 或者width最大值都为800,所以我们将图片平铺到800*800的方格中.虽然牺牲了一部分标签训练的能力,但是对模型整体起到了一点的增益,而且在resize的lanczos插值算法的辅助下,单模型能力从val_fmeasure42,到了44~45之间, 然后融合3模型之后模型到达48左右 (每融合一个模型大约增加1左右)

最初的想法是将图片加上一块黑色或者白色留框,从而不改变本身图片比例,后来想想平铺的话,对特征有一定增强效果,而且白色黑色也是一直混淆的标签,所以讨论之后的结果就是平铺到800*800然后resize到500*500放入模型

- 取值

```
threshold = 0.25
def arr2tag(arr):
    tags = []
    for i in range(arr.shape[0]):
        tag = []
        index = np.where(arr[i] > threshold)

        index = index[0].tolist()
        tag = [hash_tag[j] for j in index]
        tags.append(tag)
    return tags
y_tags = arr2tag(y_pred)

import os
img_name = os.listdir('test/')

df = pd.DataFrame({'img_path':img_name, 'tags':y_tags})
for i in range(df['tags'].shape[0]):
    df['tags'].iloc[i] = ','.join(str(e) for e in df['tags'].iloc[i])
df.to_csv('merged_moudle_best9_27_3_%.csv'%(threshold),index=None)`
```

再多标签的取值过程中,从一开始的0.5到0.25(本身如果多试几次可能会更高点),由于本身的标签数量多,所以不是每一个标签的sigmoid概率都被训练上来了,有少部分训练上来了,但是模型本身对其不自信,所以更改一下取值阈值,是模型将更多已经训练到的但是不是很自信的标签加到预测来.调整阈值从0.5到0.25,模型fmeasure从43.64到了47.42.可以上下继续调整一下,估计可以稍微再提高点,但是本身没有太大提高了

- 训练分batch载入

电脑里预读所有图片对内存要求太高,所以使用分batch将图片放入显存.

- 在输出层和GAP层中加入一个层

加上这个层后,训练最终效果不明显,大约0.5,但是训练效率提高很多,在第一个epoch就可以到达很高的val_fmeasure,30~40左右,成型较快.假如机器运行,可以尝试多加几个没准有效果只是本身没有尝试而且当时已经放弃299*299的图片像素.

- 图片大小选择

299*299图片大小增加到500*500时,fmeasure有提升提升.但是提升到800*800的图片大小时,效果不是太好,出现了收敛慢,可能特征数量太多,没准中间假如dense可以帮助处理,但是没有时间尝试.