数据增强（二）-SamplePairing

实验背景

在数据增强（一）中介绍了imgaug图像增强库，本文介绍SamplePairing的数据增强策略。

参考文献：Data Augmentation by Pairing Samples for Images Classification

实验内容

增强策略

- 训练集中任一的图片A（256x256， 标签为A），经过普通增强后随机patch出224*224的区域
- 随机选择（256x256，标签为B）的图片B，也经过普通增强后随机patch出224*224的区域
- patch部分像素平均：(patch A + pathc B)/2
- 新合成的A（256x256 标签为A) 参与训练

> 关键点：合成后的图像保留标签A不变，不考虑标签B的部分；但是合成的数据中既有标签A也有标签B的数据，训练中会存在会存在训练误差和验证误差降低的情况； 所以需要再最后不使用SamplePairing增强情况，做finetune后的效果要高于训练中没有SamplePairing增强的训练；

训练步骤

- without SamplePairing: 先按照普通的数据增强策略训练若干epoch（如100个epoch）
- enable SamplePairing: 接下来的8个epoch SamplePairing增强+2个epoch普通增强， 执行若干个组合（enable SamplePairing for 8 epochs and disable it for the next 2 epochs）
- disable the SamplePairing as the fine-tuning: 执行普通的数据增强策略最后训练若干epoch。

实现细节

```
import random
def patch_range(h, w, patch_h, patch_w):
    diff_h = h - patch_h
    diff_w = w - patch_w
    h_select_range = [i for i in range(0, diff_h)]
    w_select_range = [i for i in range(0, diff_w)]
    h_idx = random.sample(h_select_range, 1)[0]
    w_idx = random.sample(w_select_range, 1)[0]
    return h_idx, h_idx + patch_h, w_idx, w_idx + patch_w

def sample_pair_batch(x, y, h=224, w=224, patch_h=196, patch_w=196, class_num=3):
    """
    Data Augmentation by Pairing Samples for Images Classification
    :param x: [n, h, w, c]
    :param y: [b, class_num]
    :param h: input height
    :param w: input width
    :param patch_h: patch height ( 3/4 * input height)
    :param patch_w: patch width ( 3/4 * input width)
```

```
    :param class_num: the number of class
    :return:
    """
    class_idx = np.arange(0, class_num)
    class_idx_another = class_idx + 1
    class_idx_another[-1] = 0
    class_rela_map = dict(zip(class_idx, class_idx_another))

    # get different lable index
    label_index = np.argmax(y, axis=1)

    # different class sample index
    class_label_index_list = [np.where(label_index == i)[0] for i in
range(class_num)]
    pair_sample = []
    for idx, sample in enumerate(x):
        sample_label = label_index[idx]
        label_index_list =
class_label_index_list[class_rela_map[sample_label]].tolist()
        if len(label_index_list) == 0:
            pair_sample.append(x[[idx]])
        else:
            pair_sample.append(x[random.sample(label_index_list, 1)])

    pair_sample_x = np.concatenate(pair_sample)

    begin_h, end_h, begin_w, end_w = patch_range(h, w, patch_h, patch_w)
    begin_h2, end_h2, begin_w2, end_w2 = patch_range(h, w, patch_h, patch_w)

    x[:, begin_h:end_h, begin_w:end_w] = (x[:, begin_h:end_h, begin_w:end_w]
    + pair_sample_x[:, begin_h2:end_h2,begin_w2:end_w2]) // 2
    return x, y
```

实验结语

本实验介绍了samples pair的数据增强策略，通过随机叠加两个图片的方式来形成一个强的正则化器， 提高模型的泛化能力。

希望对大家有帮助。