AutomatedTesting2020 AI 数据扩增大作业

张程昱 181250183

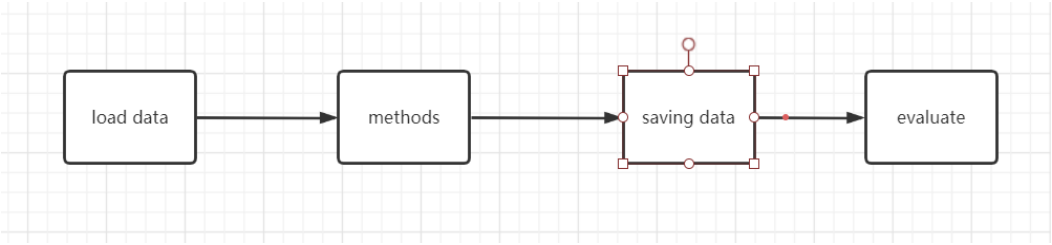
一、 选题方向：AI 数据扩增，CIFAR-10，CIFAR-100 数据集

二. 简介

由于 cifar10 与 cifar100 相似。所以大部分数据处理方法、评估方法类似，代码都可以重用。下文分析不做单独介绍。

代码详细介绍在 demo 视频中。

三. 代码运行结构图：




```
Xtrain, Ytrain, Xtest, Ytest = load.load_CIFAR_data(data_dir_10)
Xres, Yres = methods.nothing(Xtrain,Ytrain,50000)
eval.eval_10(Xtrain[:1000],Ytrain[:1000])
np.savez('../Data/cifar10_result_data/feature.npz',data=Xres,labels=Yres,allow_pickle=True)
```

四. 数据扩增方法：

PS:详细的处理效果放在/Data/ operation_data_png_cifar100 下了。每种方法 10 张图片。

0. Oringin

1. 旋转

通过把图片旋转，达到数据增强的目的。效果如下图。

2. 位移

对图片进行上下左右的位移操作（超过边界部分取同，位移幅度较小）

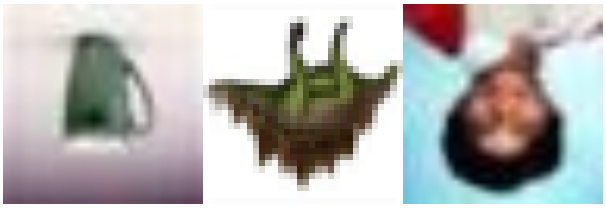
效果如下图。

3. Zca 白化

调用 keras 的 ImageDataGenerator 的库，白化图片。效果如下图。

4. 翻转

上下左右翻转，效果如下图。



5. Feature 标准化

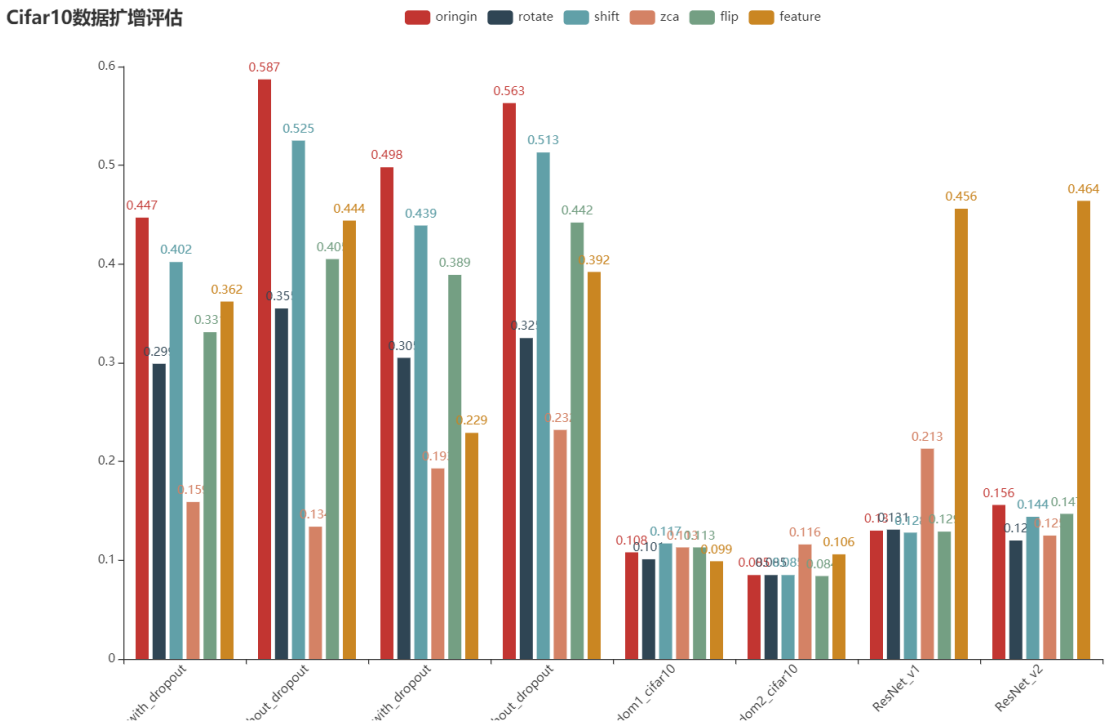
featurewise_center 使得输入数据集去中心化，就是把像素点三通道取平均值…还使用了 featurewise_std_normalization，这部分是把输入的数据除以数据集的标准差。（应该起到的是把特征显化的作用）

相关方法代码例如下

```python
def composite(images, labels, length):
    print('composite:')
    datagen_sample = ImageDataGenerator(
        rotation_range=90,
        fill_mode='nearest',
        width_shift_range=0.2,
        height_shift_range=0.2,
        horizontal_flip=True,
        vertical_flip=True,
    )
    datagen_sample.fit(images)
    iter = datagen_sample.flow(x=images, y=labels, batch_size=10, shuffle=False,save_to_dir='../Data/operation_data_png_cifar
    Xtmp, Ytmp = iter.next()
    iter = datagen_sample.flow(x=images, y=labels, batch_size=length,shuffle=False)
    i = 0
    while i<10:
        x_res, y_res = iter.next()
        i += 1
    print(x_res.shape)
    return x_res, y_res
```

四．评估

将扩增后的数据用来测试模型正确率，寻找模型的泛化能力较弱的部分。下图是各种方法对各个模型的的正确率影响。注：下图是 html 文件截图，可以打开相关文件选取单个方法查看争取率情况。



由于笔者做测试的时候后面 4 个模型的初始效果也不是太好，所以就不做过多阐述。以及，在评估的时候所有模型还都是欠训练的…，个人认为不一定能真实反映出模型对于数据的接收能力，下面的分析建立在当前模型已经达到收敛状态下。

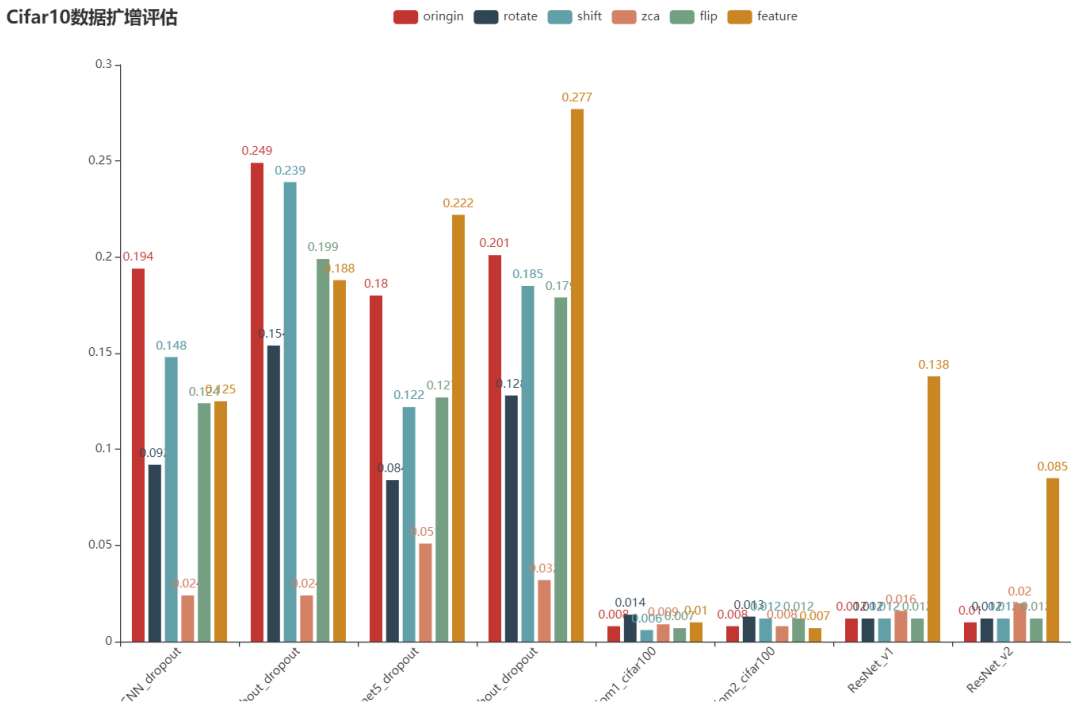
可以看到所有数据增强方式中。位移变换对于正确率的影响较小，zca 方法对正确率影响最大，每种方法都反应出模型目前可能提高的一些地方。

在 rotate

令人惊讶的是 featurewise 标准化之后数据在 ResNet 模型的测试当中正确率上升了。以及在 cifar100 中 lenet 模型测试下正确率也提高了。（盲猜应该

就是模型欠训练的问题，分析不出来为什么）。

下图是 cifar100 数据增强效果的测试状况。Cifar100 的情况与 cifar10 相似。



五．模型训练（失败反思和改进方向）

1．完成的工作

本来希望通过使用扩增过后的数据集重新训练一下，通过对比各个模型的测试正确率情况做分析，验证数据扩增（数据增强）的效果。

在笔者的模型训练过程中。主要出现的问题是时间太短，不能充分训练。本打算对每个模型，使用 3 个数据集进行训练：

Oringin：源数据集。

Composite：多种几何变换方法使用后的数据集。

Zca：通过 zca 白化数据训练模型。

相关代码如下：

```
model_names10 = ["CNN_with_dropout.h5",'CNN_without_dropout.h5','lenet5_with_dropout.h5','lenet5_without_dropout.h5']
model_names100 = ["CNN_with_dropout.h5",'CNN_without_dropout.h5','lenet5_with_dropout.h5','lenet5_without_dropout.h5']


def alchemy10(train_images, train_labels, method):
    for i in range(1):
        checkpoint_path = 'cp-{epoch:04d}.ckpt'
        cp_callback = tf.keras.callbacks.ModelCheckpoint(
            filepath=checkpoint_path,
            verbose=1,
            save_weights_only=True,
            period=5)
        new_model = tf.keras.models.load_model('../model/cifar10/'+model_names10[i])
        optimizer = tf.keras.optimizers.Adam(learning_rate=8e-5)
        new_model.compile(optimizer=optimizer, loss='sparse_categorical_crossentropy', metrics=['accuracy'])
        new_model.summary()
        new_model.fit(train_images,
                      train_labels,
                      epochs=50,
                      batch_size=128)
        new_model.save('../model/my_model_cifar10/'+method+"/"+model_names10[i])
```

（1） 但是在参数设置时，学习率当时设置的是 8e-10，学习率太低了。

（2） 训练时间太短。训练不充分，最后训练后的结果为（测试集为原始测试集和数据增强后的一万条数据——**未被拿去训练**）：

```
My_model oringin   CNN_with_dropout.h5 正确率为:19.75%
My_model oringin   CNN_without_dropout.h5 正确率为:24.75%
My_model oringin   lenet5_with_dropout.h5 正确率为:16.94%
My_model oringin   lenet5_without_dropout.h5 正确率为:20.51%
My_model composite   CNN_with_dropout.h5 正确率为:19.67%
My_model composite   CNN_without_dropout.h5 正确率为:24.34%
My_model composite   lenet5_with_dropout.h5 正确率为:16.94%
My_model composite   lenet5_without_dropout.h5 正确率为:20.50%
My_model zca   CNN_with_dropout.h5 正确率为:19.18%
My_model zca   CNN_without_dropout.h5 正确率为:24.01%
My_model zca   lenet5_with_dropout.h5 正确率为:16.62%
My_model zca   lenet5_without_dropout.h5 正确率为:19.95%
My_model oringin   CNN_with_dropout.h5 正确率为:7.16%
My_model oringin   CNN_without_dropout.h5 正确率为:8.17%
My_model oringin   lenet5_with_dropout.h5 正确率为:5.80%
My_model oringin   lenet5_without_dropout.h5 正确率为:6.74%
My_model composite   CNN_with_dropout.h5 正确率为:7.15%
My_model composite   CNN_without_dropout.h5 正确率为:7.95%
My_model composite   lenet5_with_dropout.h5 正确率为:5.79%
My_model composite   lenet5_without_dropout.h5 正确率为:6.68%
My_model zca   CNN_with_dropout.h5 正确率为:7.10%
My_model zca   CNN_without_dropout.h5 正确率为:7.88%
My_model zca   lenet5_with_dropout.h5 正确率为:5.64%
My_model zca   lenet5_without_dropout.h5 正确率为:6.60%
```

然后就没有时间进行模型训练了。

2. 笔者最后尝试对 CNN_with_dropout 通过原始数据集和几何扩增后数据集(composite)再训练然后分析其效果。注：两个模型仍未训练完全。

模型\测试集	oringin	composite
CNN_with_dropout	74.47%	30.16%
CNN_with_dropout_composite	51.03%	45.39%

虽然两个模型仍未训练完全，但是可以看出使用扩充过的数据集之后对几何变换的数据的接收能力，模型的泛化能力增强。但是对于原测试集的正确率下降。说明训练过程中，对原数据特征会有一定的损失。

2. 完善方向：

（1） 对多个扩增后的数据集进行训练，综合多种扩增后的数据评估模型的鲁棒性能力。

（2） 综合原数据集和扩增后的数据集进行训练，检测对多种数据的接收能力（测试正确率）