白盒攻击

how to run

```
python train.py
python test.py
python white_box_attack.py
```

分类器在 test 集上的准确率为 0.9138。

分类器源码见 cnn.py。

白盒攻击成功率为 0.997。每个样本迭代 50 次。若迭代次数设定为 60+，则攻击成功率为 100%。

白盒攻击成功率随迭代次数增加的变化曲线见图 1。

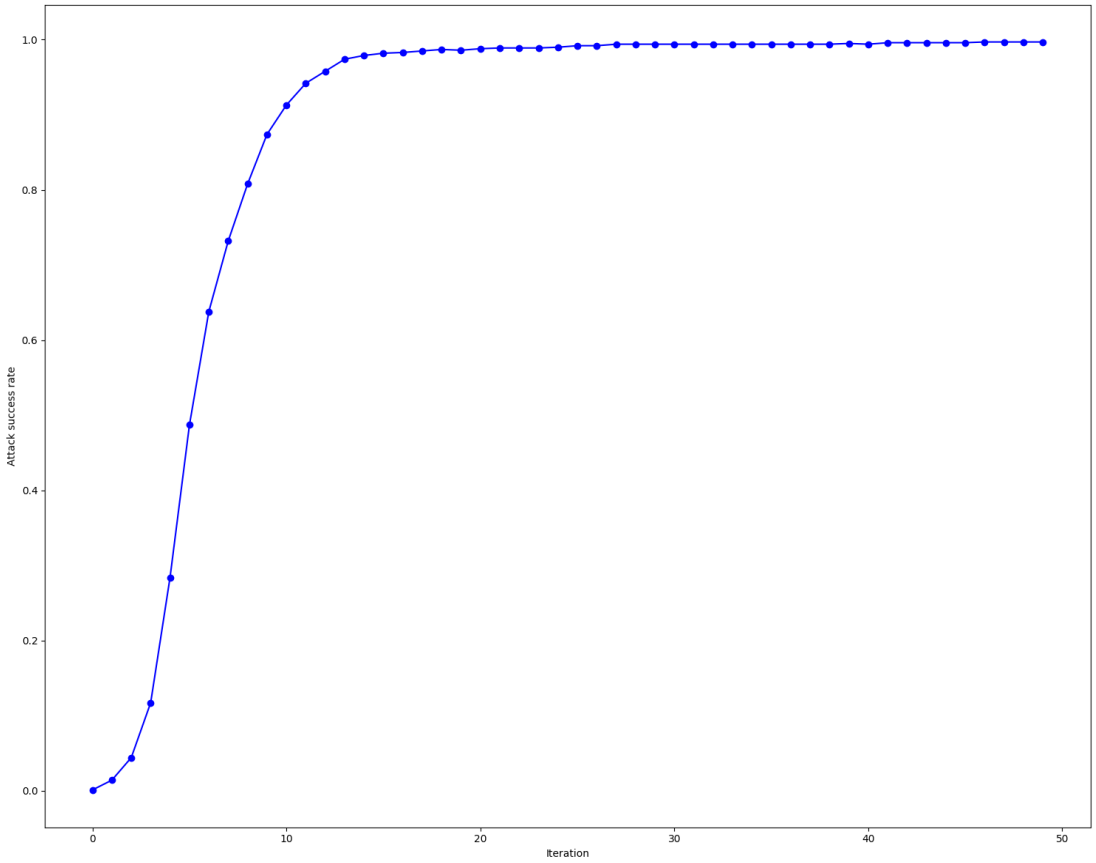

图 1 白盒攻击迭代曲线

随机抽取的 10 组原图像及其对抗样本图像见图 2。

图 2 原图像及其对抗样本图像

部分对抗样本图像与原图之间存在失真现象，主要原因是我对所有样本批量迭代了 50 次。批量迭代是为了加速运算。但实际上若某一样本已经攻击成功，则不必再迭代。在黑盒攻击中，我执行了若攻击成功则停止迭代的策略。

黑盒攻击

使用数值解代替解析解，通过估计梯度来模拟白盒攻击[1]。这种方法计算量大，较为耗时。

[1] https://arxiv.org/pdf/1708.03999.pdf

how to run

```
python black_box_attack.py
python black_box_attack_other.py
```

对自己的模型进行黑盒攻击，攻击成功率为 100%，每个样本的迭代上限为 50 次，逐个迭代。随机抽取的 10 组原图像及其对抗样本图像见图 3。

图 3 原图像及其对抗样本图像

对助教的模型进行黑盒攻击，攻击成功率为 100%，每个样本的迭代上限为 50 次，逐个迭代。随机抽取的 10 组原图像及其对抗样本图像见图 4。

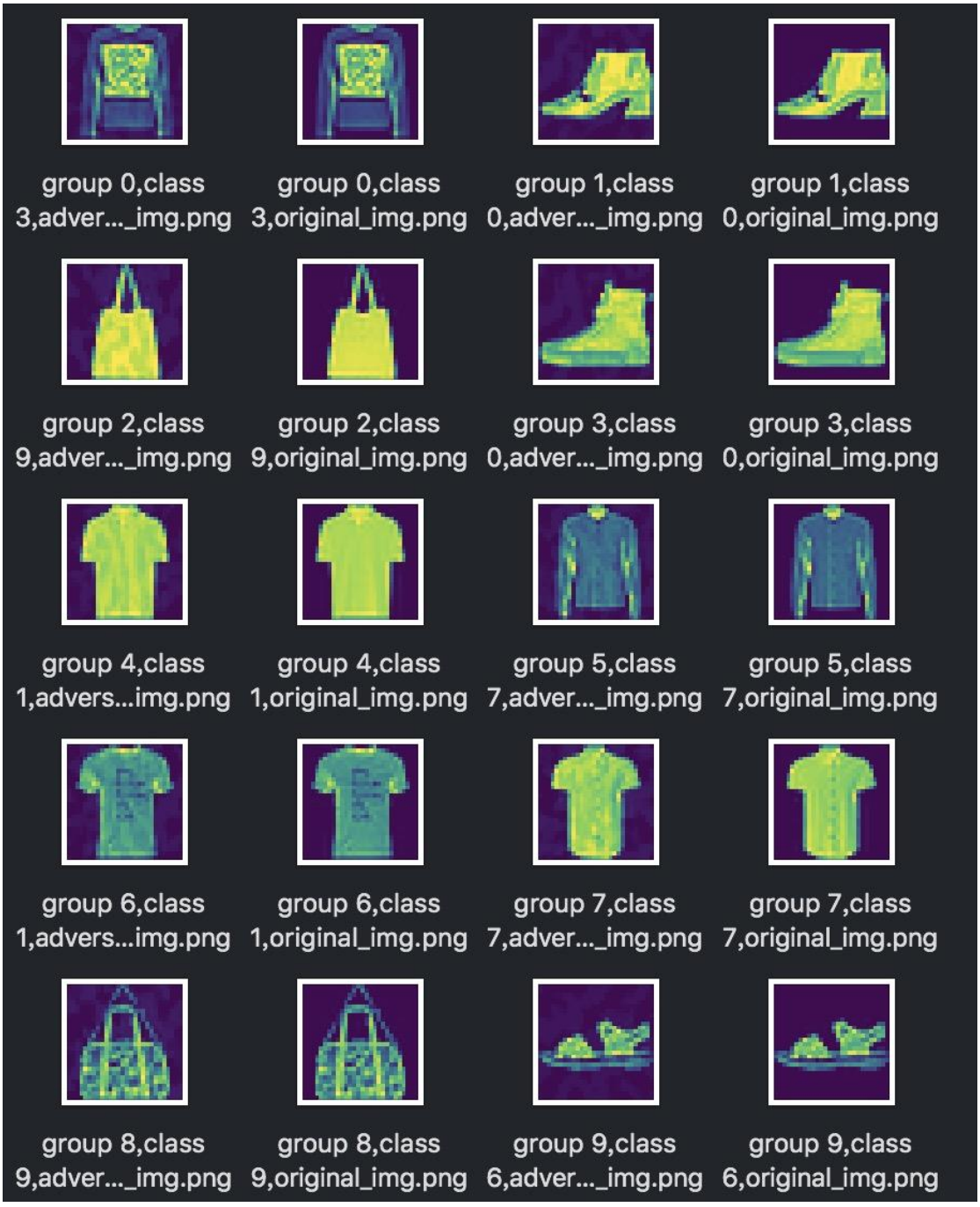

图 4 原图像及其对抗样本图像

采取了若攻击成功则停止迭代的策略，失真现象基本消失。

对抗训练

how to run

```
python adv_train.py
python white_box_attack_adv_trained.py
python black_box_attack_adv_trained.py
```

旧分类器在 test 集上的准确率为 0.9138。
新分类器在 test 集上的准确率为 0.9135。
白盒攻击在旧分类器上的成功率为 99.7%。
白盒攻击在新分类器上的成功率为 100%。
随机抽取的 10 组原图像及其对抗样本图像见图 5。

图 5 原图像及其对抗样本图像

由于白盒攻击基于梯度且数据集较为简单，攻击成功率基本没变化。

黑盒攻击在旧分类器上的成功率为 100%。

黑盒攻击在新分类器上的成功率为 100%。

随机抽取的 10 组原图像及其对抗样本图像见图 6。

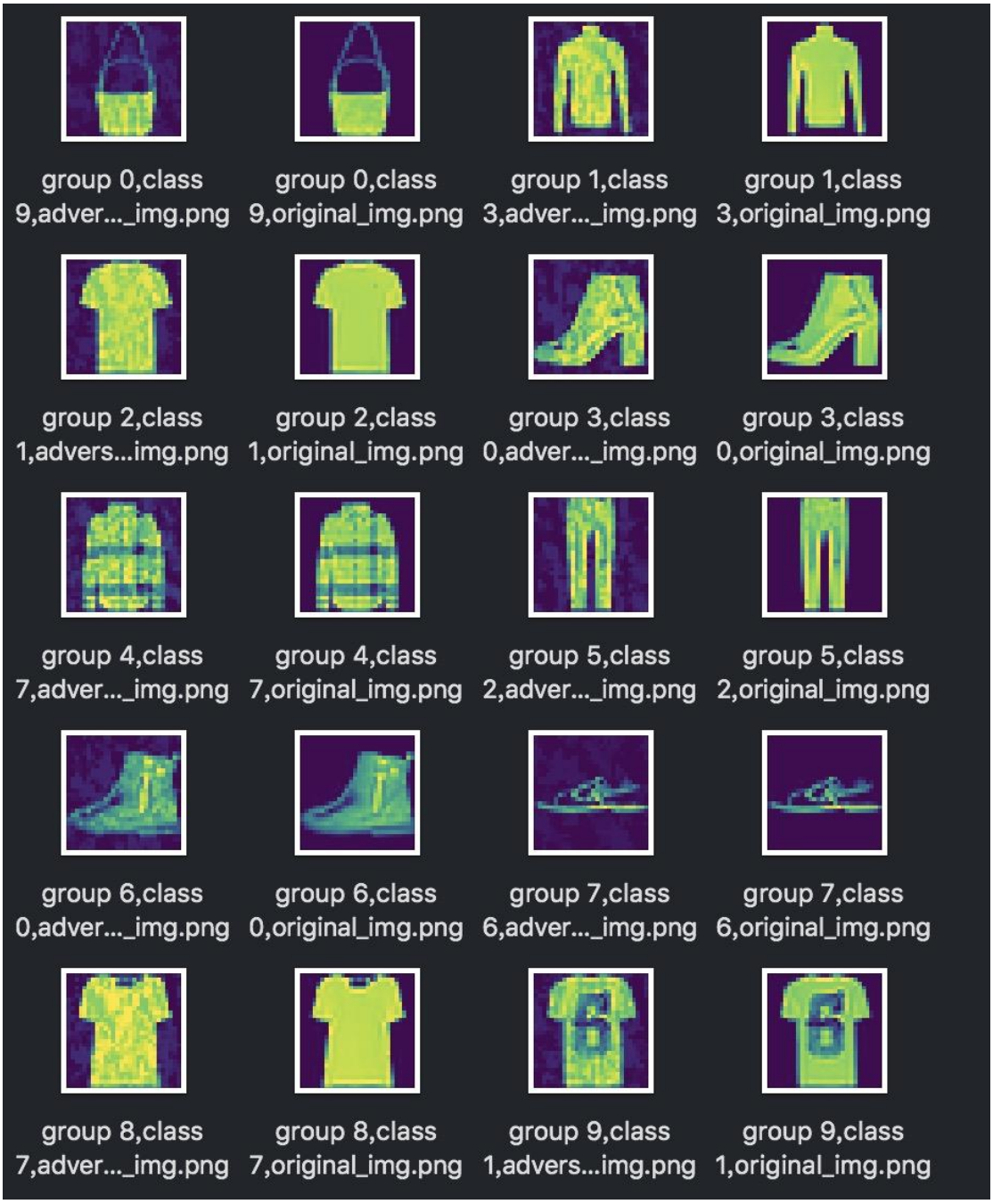

图 6 原图像及其对抗样本图像

由于我实现的黑盒攻击基于梯度估计且数据集较为简单，攻击成功率基本没变化。