**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%。

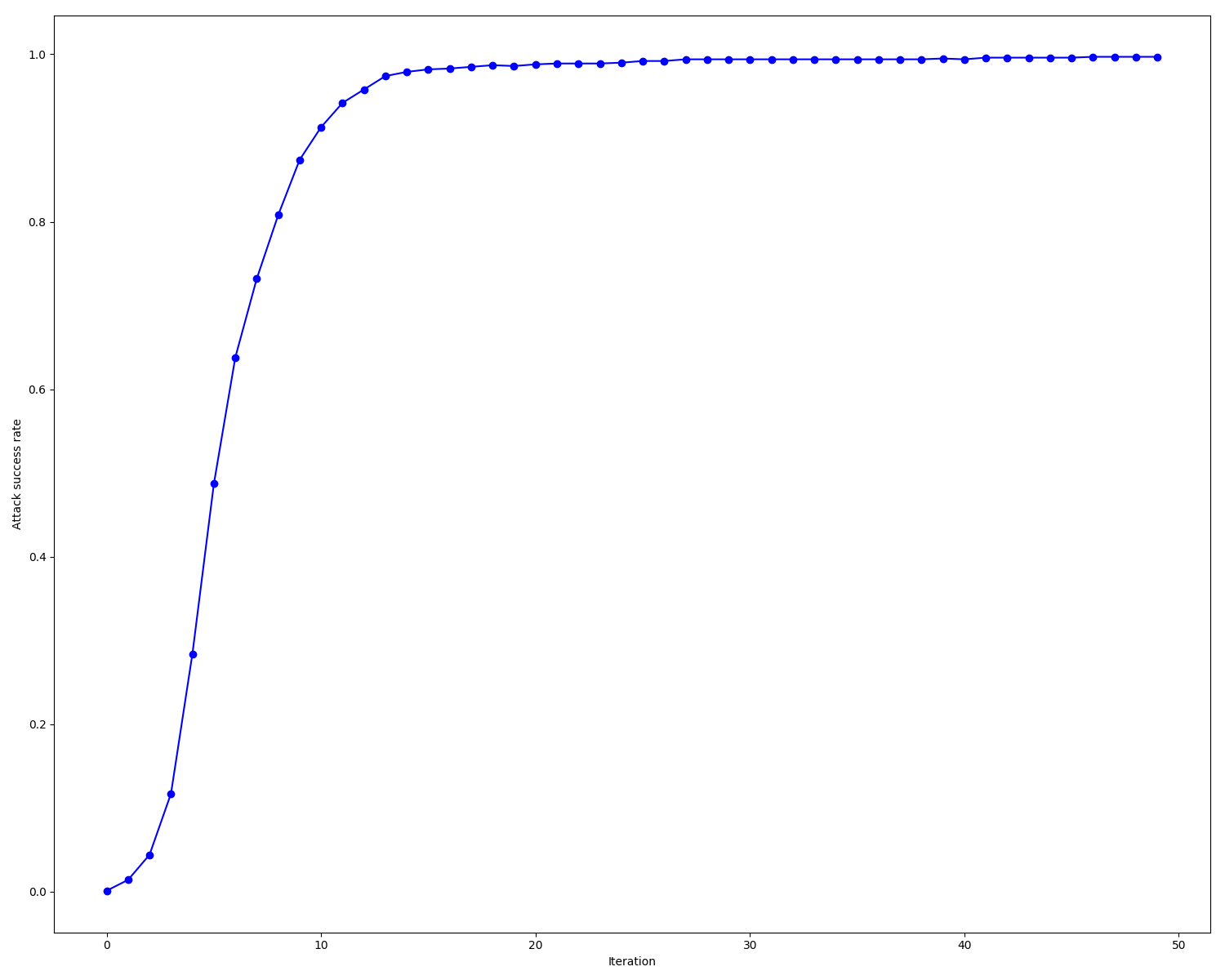
白盒攻击成功率随迭代次数增加的变化曲线见图1。

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]](#footnote-1)。这种方法计算量大，较为耗时。

**how to run**

python black\_box\_attack.py

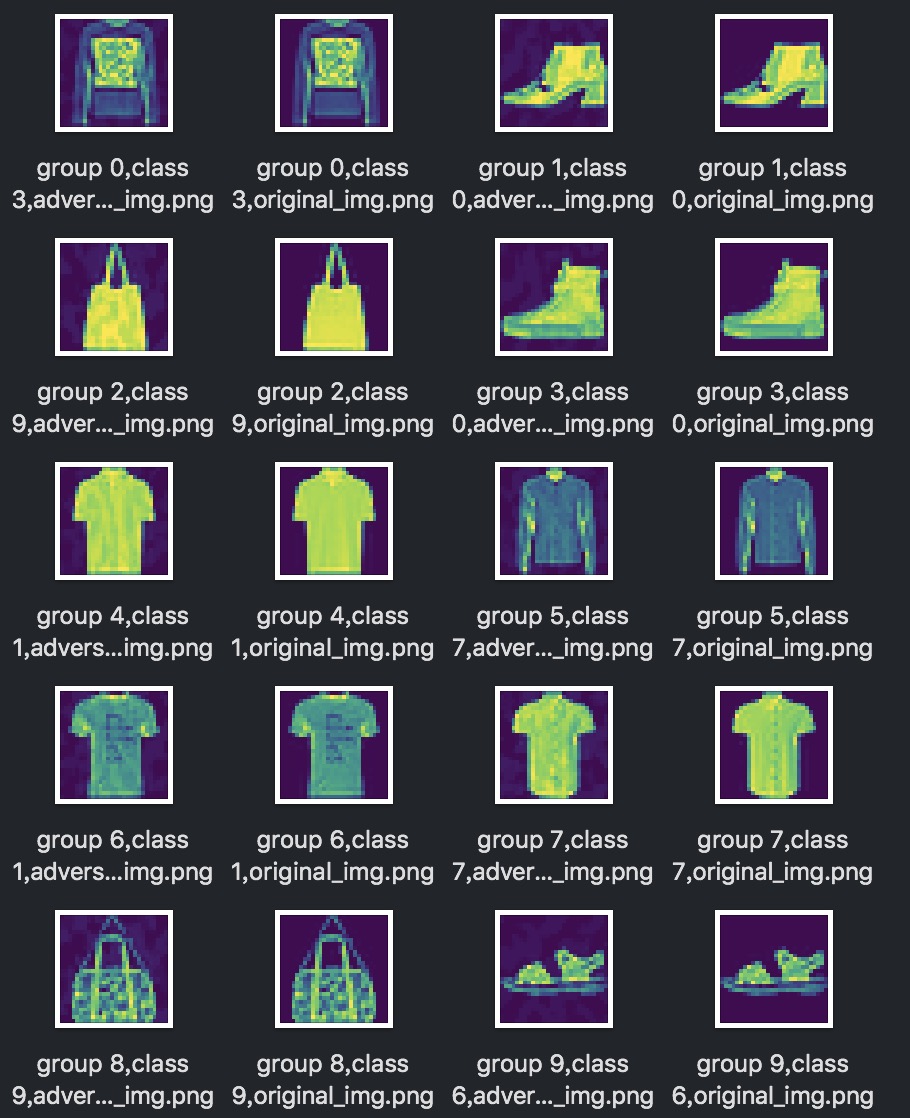
python black\_box\_attack\_other.py

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



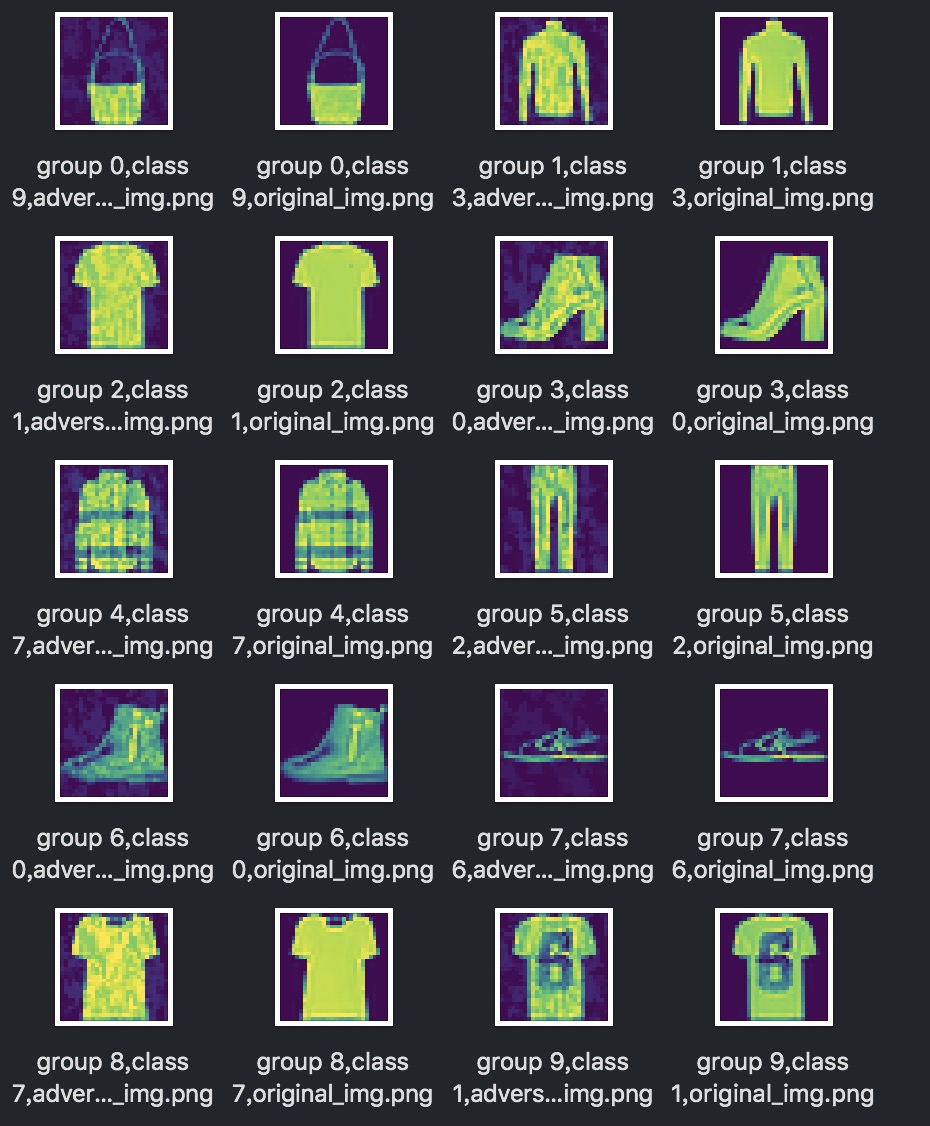
由于白盒攻击基于梯度且数据集较为简单，攻击成功率基本没变化。

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。

1. https://arxiv.org/pdf/1708.03999.pdf [↑](#footnote-ref-1)