复现点1：攻击MNIST分类器

$$\eta = \varepsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

论文取ε = 0.25时，去攻击一个MNIST的shallow softmax classifier，错误率达到99%。现尝试复现此结果。

由于我手头的MNIST分类器是一个CNN的深度网络，没有shallow softmax classifier，我就拿这个模型进行攻击。如图，当ε = 0.25时，错误率为64.11%。当ε 上升至0.4时，错误率高达83.46%，这个结果是很好的。

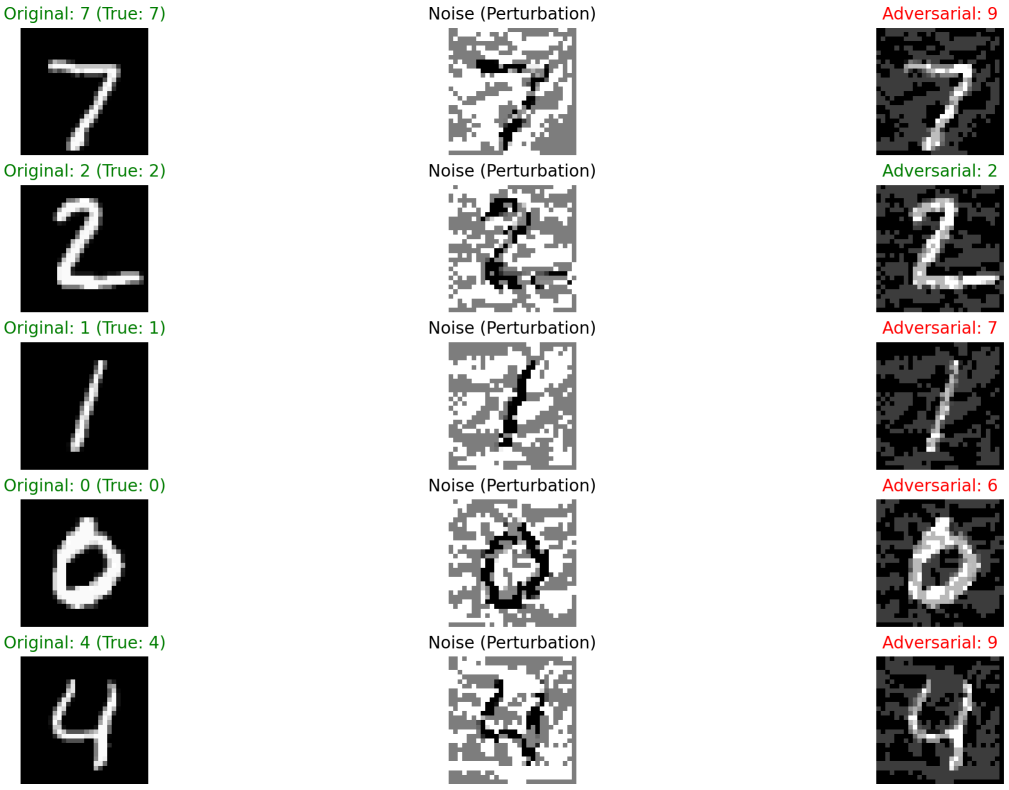
```
(d2l) D:\MNIST\MNIST>D:/conda/envs/d2l/python.exe d:/MNIST/MNIST/raw/FGSM_Attack.py
正在初始化 CNN 模型...
正在加载参数: raw/model/model.pkl
参数加载成功!

开始攻击 (使用 Epsilon = 0.25)...
Epsilon: 0      Test Accuracy = 98.87%  Error Rate = 1.13%
Epsilon: 0.25   Test Accuracy = 35.89%  Error Rate = 64.11%

(d2l) D:\MNIST\MNIST>D:/conda/envs/d2l/python.exe d:/MNIST/MNIST/raw/FGSM_Attack.py
正在初始化 CNN 模型...
正在加载参数: raw/model/model.pkl
参数加载成功!

开始攻击 (使用 Epsilon = 0.4)...
Epsilon: 0      Test Accuracy = 98.87%  Error Rate = 1.13%
Epsilon: 0.4    Test Accuracy = 16.54%  Error Rate = 83.46%
```

因为我的模型由两个卷积层，两个池化层，以及三个全链接层构成，整体的线性特性是比较弱的了，所以对于对抗的抵御是较好的，因而在ε = 0.25时无法达到错误率99%，但ε = 0.4时错误率高达83.46%，这可以说明该文献所提出的进攻手段的有效性与合理性，复现成功。



可视化 ε = 0.25

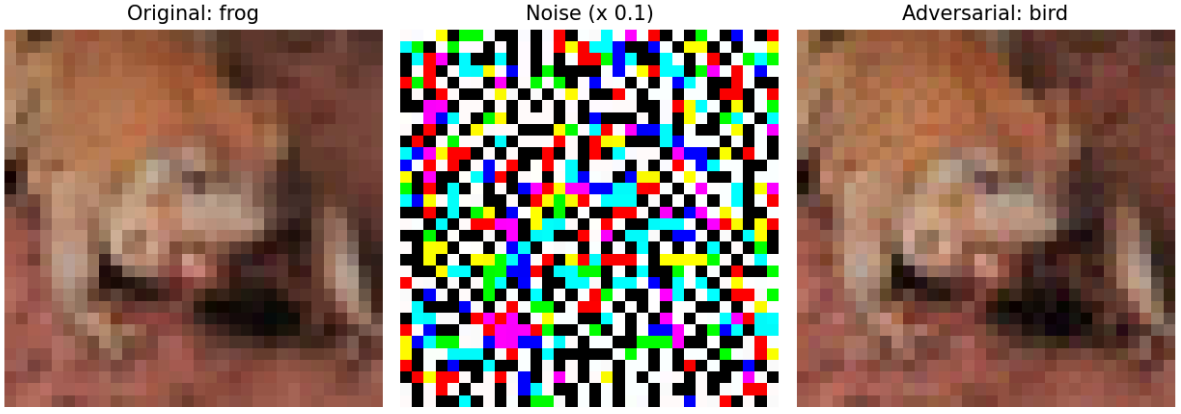
复现点2：攻击Cifar10分类器

论文取ε = 0.1，去攻击一个Cifar10的maxout卷积网络。错误率89.4%，置信程度96.6%。

我用我自己的模型复现出来，错误率70.2%，置信程度85.94%，结果还可以。

```
==============================
FGSM Attack Result (Epsilon = 0.1)
==============================
Total Evaluated Images (Correctly Classified Originally): 1000
Adversarial Error Rate (Attack Success Rate): 70.20%
Model Accuracy after Attack: 29.80%
Avg. Confidence on Wrong Labels: 85.94%
==============================
论文复现对比：论文中 Error Rate 87.15%, Avg Conf 96.6%
```

论文所攻击的模型是一个maxout的模型而我用的是RELU，按理来说我的RELU对于如此线性干扰应该更脆弱才对，但是我的网络深度确实很大，线性特性很弱。再加上论文说它做的Cifar10是“预处理”过的，所以可能会导致进攻表现上升。



可视化

复现点3：对抗训练MNIST

论文里提出的对抗训练方法是“一边对抗一边训练”，也就是对于每一个batch都是前向传播，算出当前输入

的梯度，然后根据这个梯度生成对抗样本。同时把清洁样本和对抗样本给模型，算Loss，反向传播更新参数。

直接看我自己的训练结果。

```
...   ————————第1轮测试开始————————
      训练次数为：0，损失值为：0.02171141840517521
      训练次数为：100，损失值为：0.0013463676441460848
      训练次数为：200，损失值为：0.004746945574879646
      训练次数为：300，损失值为：0.015061729587614536
      训练次数为：400，损失值为：0.005062873475253582
      训练次数为：500，损失值为：0.0006615120219066739
      训练次数为：600，损失值为：0.00022185560374055058
      训练次数为：700，损失值为：0.012584542855620384
      训练次数为：800，损失值为：0.014109604060649872
      训练次数为：900，损失值为：0.0008858605287969112
      测试准确率为：0.991200
```

初始模型清洁数据acc达到99.12%，对抗数据35.89%。

经过对抗训练之后，清洁数据98.87%，对抗数据95.64%。

```
预训练参数加载成功！在此基础上进行对抗微调。
开始对抗训练... Epsilon = 0.25
Train Epoch: 1 [0/60000]       ClnLoss: 0.0005  AdvLoss: 6.1185       ClnAcc: 100.0%  AdvAcc: 32.8%
Train Epoch: 1 [6400/60000]    ClnLoss: 0.0394  AdvLoss: 2.3895       ClnAcc: 98.7%  AdvAcc: 42.3%
Train Epoch: 1 [12800/60000]   ClnLoss: 0.0513  AdvLoss: 1.8289       ClnAcc: 98.5%  AdvAcc: 49.5%
Train Epoch: 1 [19200/60000]   ClnLoss: 0.0562  AdvLoss: 1.5835       ClnAcc: 98.4%  AdvAcc: 53.8%
Train Epoch: 1 [25600/60000]   ClnLoss: 0.0614  AdvLoss: 1.4461       ClnAcc: 98.3%  AdvAcc: 56.4%
Train Epoch: 1 [32000/60000]   ClnLoss: 0.0642  AdvLoss: 1.3490       ClnAcc: 98.2%  AdvAcc: 58.6%
Train Epoch: 1 [38400/60000]   ClnLoss: 0.0641  AdvLoss: 1.2736       ClnAcc: 98.2%  AdvAcc: 60.3%
Train Epoch: 1 [44800/60000]   ClnLoss: 0.0658  AdvLoss: 1.2205       ClnAcc: 98.1%  AdvAcc: 61.5%
Train Epoch: 1 [51200/60000]   ClnLoss: 0.0663  AdvLoss: 1.1789       ClnAcc: 98.1%  AdvAcc: 62.4%
Train Epoch: 1 [57600/60000]   ClnLoss: 0.0681  AdvLoss: 1.1463       ClnAcc: 98.1%  AdvAcc: 63.2%

Evaluating Robustness...
Test set: Clean Accuracy: 97.93%
Test set: Adversarial Accuracy (Epsilon=0.25): 68.77%
```

```
Train Epoch: 10 [0/60000]      ClnLoss: 0.0563  AdvLoss: 0.4165       ClnAcc: 98.4%  AdvAcc: 92.2%
Train Epoch: 10 [6400/60000]   ClnLoss: 0.0431  AdvLoss: 0.1809       ClnAcc: 98.5%  AdvAcc: 94.2%
Train Epoch: 10 [12800/60000]  ClnLoss: 0.0425  AdvLoss: 0.1751       ClnAcc: 98.6%  AdvAcc: 94.3%
Train Epoch: 10 [19200/60000]  ClnLoss: 0.0441  AdvLoss: 0.1723       ClnAcc: 98.7%  AdvAcc: 94.4%
Train Epoch: 10 [25600/60000]  ClnLoss: 0.0440  AdvLoss: 0.1689       ClnAcc: 98.7%  AdvAcc: 94.6%
Train Epoch: 10 [32000/60000]  ClnLoss: 0.0443  AdvLoss: 0.1649       ClnAcc: 98.6%  AdvAcc: 94.7%
Train Epoch: 10 [38400/60000]  ClnLoss: 0.0432  AdvLoss: 0.1612       ClnAcc: 98.6%  AdvAcc: 94.8%
Train Epoch: 10 [44800/60000]  ClnLoss: 0.0418  AdvLoss: 0.1569       ClnAcc: 98.7%  AdvAcc: 94.9%
Train Epoch: 10 [51200/60000]  ClnLoss: 0.0416  AdvLoss: 0.1553       ClnAcc: 98.7%  AdvAcc: 95.0%
Train Epoch: 10 [57600/60000]  ClnLoss: 0.0424  AdvLoss: 0.1532       ClnAcc: 98.7%  AdvAcc: 95.1%

Evaluating Robustness...
Test set: Clean Accuracy: 98.87%
Test set: Adversarial Accuracy (Epsilon=0.25): 95.64%

==> 发现新纪录！Adv Acc 从 95.64% 提升到了 95.64%。保存模型...
训练结束！最佳对抗准确率: 95.64%
```

第一轮与最后一轮训练结果

论文中说，经过对抗训练，错误率从 89.4% 降到了 17.9%（Maxout网络）。我的CNN网络用的是RELU，按理来说抗干扰能力应该不如Maxout网络，但是做出来效果远好于论文的结果。

而且论文中说对抗训练有正则化作用，能把正确率从99.06%提到99.16%，这个正则化效果可以和dropout等常规正则化手段相媲美，但是这个正则化我没复现出来。我对抗训练之后清洁数据的acc掉了一点点，但好在是也没掉太多。

论文提到了两个针对欠拟合的方法，一是增大模型容量，二是更改early stop策略，可是这是针对对抗集的欠拟合，我当前的问题不是对抗集拟合程度不够，而是原始集欠拟合了。我的训练是没有用early stop的。我尝试调大α（α=0.6）进行训练，效果甚至不如α=0.5。

这个问题困扰了我好久，后来Ai告诉我说在2015年后，大家达成了一个普遍共识：鲁棒性（Robustness）和 清洁精度（Clean Accuracy）之间往往存在矛盾，要进行取舍的。

复现点4：对照实验随机噪声训练

论文为了证明对抗训练的不可替代性，做了两个随机噪声的训练，分别是随机$\pm\varepsilon$，以及随机$(-\varepsilon,+\varepsilon)$，论文说错误率分别是86.2%和90.4%。我自己做出来分别是69.07%和80.89%。

```
Test set: Clean Accuracy: 98.91%
Test set: FGSM Accuracy (Epsilon=0.25): 30.93%
Test set: FGSM Accuracy (Epsilon=0.25): 30.93%
```

```
正在进行真正的 FGSM 攻击测试 (Exam)...
Test set: Clean Accuracy: 98.92%
Test set: FGSM Accuracy (Epsilon=0.25): 19.11%
```

这个算是复现出来了，整体错误率低，这个和前面的的全部复现结果保持一致，说明就是模型架构的问题。论文是连续型的噪声错误率高于离散型噪声，这个我的复现也是一样。我的复现结果足够说明对抗训练的不可替代性。

复现点5：针对隐藏层的对抗训练

论文说对抗训练，基于输入图片的方式正则化效果要好于基于隐藏层的。基于隐藏层的对抗训练易出现欠拟合，由于模型容量相对不足，模型会通过提升前端的输出（隐藏层激活值）来提高信噪比。

```
Test set: Clean Accuracy: 97.12%
Test set: Input FGSM Accuracy: 15.75%
```

我做了一下针对隐藏层的对抗训练，取第一个全链接的输出来生成对抗样本，然后对输入图片施加FGSM进行测试，结果如下：清洁数据acc97.12%，对抗数据acc15.75%，这个甚至还不如完全不进行对抗训练的准确率。这说明针对隐藏层的对抗训练不具备泛化性，完全无法应付输入的扰动。

```
[激活值大小对比] Linear1 Output (Activation):
  - Input Model : Mean=0.61, Max=2.84
  - Hidden Model: Mean=1.85, Max=11.11
  - Mean 倍数   : 3.01x
```

清洁数据的表现（97.12%）不如针对输入的对抗训练（98.87%），模型出现了更为严重的欠拟合。这验证了论文说的“正则化效果欠佳”。同时，对比隐藏层对抗训练与常规训练后的隐藏层激活值，发现隐藏层对抗训练确实让激活值大幅提升，这成功复现Goodfellow的发现。