**西安电子科技大学网络与信息安全学院**

**信息与内容安全 实验报告**

**班 级：**

**学 号：**

**姓 名：**

**电子邮箱：**

**指导教师： 王善峰**

**2022年6月1日**

|  |
| --- |
| **实验题目：Adversarial Example Generation** |
| **实验摘要：**  **迄今为止最早和最流行的对抗性攻击之一被称为快速梯度符号攻击(FGSM)，并由Goodfellow等人描述。在解释和利用对抗性的例子 Explaining and Harnessing Adversarial Examples.** |
| **题目描述：**  **快速梯度符号攻击（FGSM）,通过利用神经网络的学习方式（梯度）来攻击神经网络,攻击不是通过根据反向传播梯度调整权重来最小化损失，而是调整输入数据以基于相同的反向传播梯度最大化损失。攻击使用输入数据的损失梯度，然后调整输入数据以最大化损失。**  所谓对抗样本就是指：在原始样本添加一些人眼无法察觉的扰动（这样的扰动不会影响人类的识别，但是却很容易愚弄模型），致使机器做出错误的判断。如下所示，这两张图片添加噪声（或者说扰动之后）被误分类。    **快速梯度符号攻击**  迄今为止最早也是最流行的对抗性攻击之一被称为*快速梯度符号攻击（FGSM），*由Goodfellow等人描述。在解释和利用对抗性示例中。攻击非常强大，但也很直观。它旨在通过利用神经网络的学习方式（*梯度）*来攻击神经网络。这个想法很简单，攻击不是通过根据反向传播梯度调整权重来最小化损失，而是*调整输入数据以*基于相同的反向传播梯度最大化损失。换句话说，攻击使用输入数据的损失梯度，然后调整输入数据以最大化损失。 |
| 上述图片就是一个生动的例子，左边是一张正常的熊猫图片，在添加了网络梯度噪声之后，生成右边的攻击图像，视觉效果差距不大，但是模型依然将其归为一只长臂猿。  那么 FGSM 的攻击噪声是怎么生成的呢？我们知道训练分类模型时，网络基于输入图像学习特征，然后经过 softmax 层得到分类概率，接着损失函数基于分类概率和真实标签计算损失值，回传损失值并计算梯度（也就是梯度反向传播），最后网络参数基于计算得到的梯度进行更新，网络参数的更新目的是使损失值越来越小，这样模型分类正确的概率也就越来越高。  **实验内容（详细内容）：**   1. **实验基本原理及步骤**   分类模型时，网络通过输入的图像特征学习，经过softmax层得到分类概率，损失函数基于真实标签和分类函数计算损失值，并将其回传计算梯度，最后根据梯度更新网络模型的参数，使损失值越来越小，分类模型的正确率越来越高。  FGSM攻击通过修改输入图像的像素值来达到攻击网络模型的目的，而非修改参数。结合上述过程，可将损失值回传到输入图像并计算梯度，其中J()为损失函数，X与Y为输入图像和真实标签，θ表示网络参数，    拟定函数sign（）为梯度的方向，使用梯度方向值的目的是控制扰动的L∞距离以此作为FGSM算法的评价指标。    正常的模型在训练时 通过减去计算得到的梯度来更新，是的损失值逐步降低，预测成功概率提高。而FGSM算法则是在输入图片上加上计算的梯度，使得训练后的模型损失值变大，预测成功率降低。梯度方向在FGSM算法中会有一个权重来控制攻击噪声的幅度 值，参数值越大，攻击强度越大。FGSM算法简单有效，在图像攻击领域扮演者重要角色。 |
|  |
|  |

**二 实验步骤**

对抗样本的线性分析：数字图像每个像素通常为8bit，因此小于1、255的信息会被舍弃，设原始图像为X，扰动噪声为η，则扰动之后的图像为：



在题目论述中叙述了非线性模型的线性扰动，通过添加一个肉眼无法识别的向量，值为ϵ \epsilonϵ与输入像素点关于误差的 梯度符号值。其中 ϵ = 0.007 \epsilon=0.007ϵ=0.007,符合经过GoogleNet转换为实数后8bit图像编码的最小数量级。使得熊猫被识别为长臂猿，而且置信度高达99%.3，因此该方法为 fast gradient sign method，简称FGSM算法。

1. 算法实现

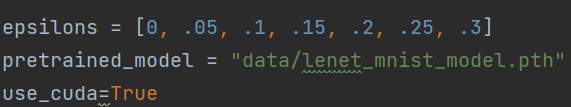
输入值：

• epsilons-用于运行的epsilons值列表。该列表的值越大，扰动越明

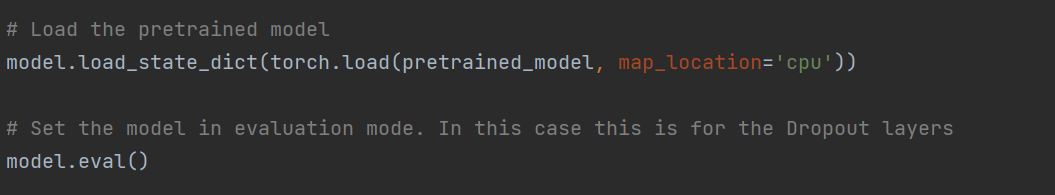
显，攻击在降低模型精度方面越有效。该值域为【0，1】。

• pretrained\_model -路径到预先训练的MNIST模型。

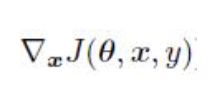
• use\_cuda – bool使用CUDA ，如需要可用。

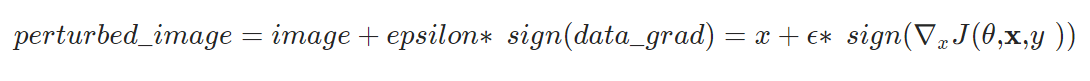


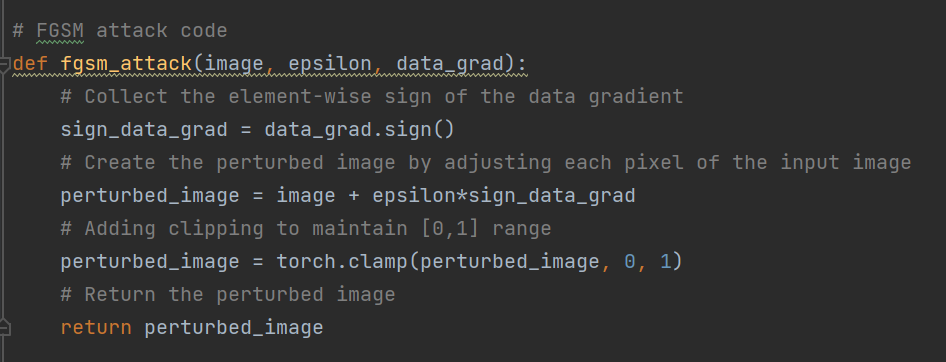




通过干扰原始输入来定义创建对抗示例的功能，输入原始图像，epsilon量，data\_gard

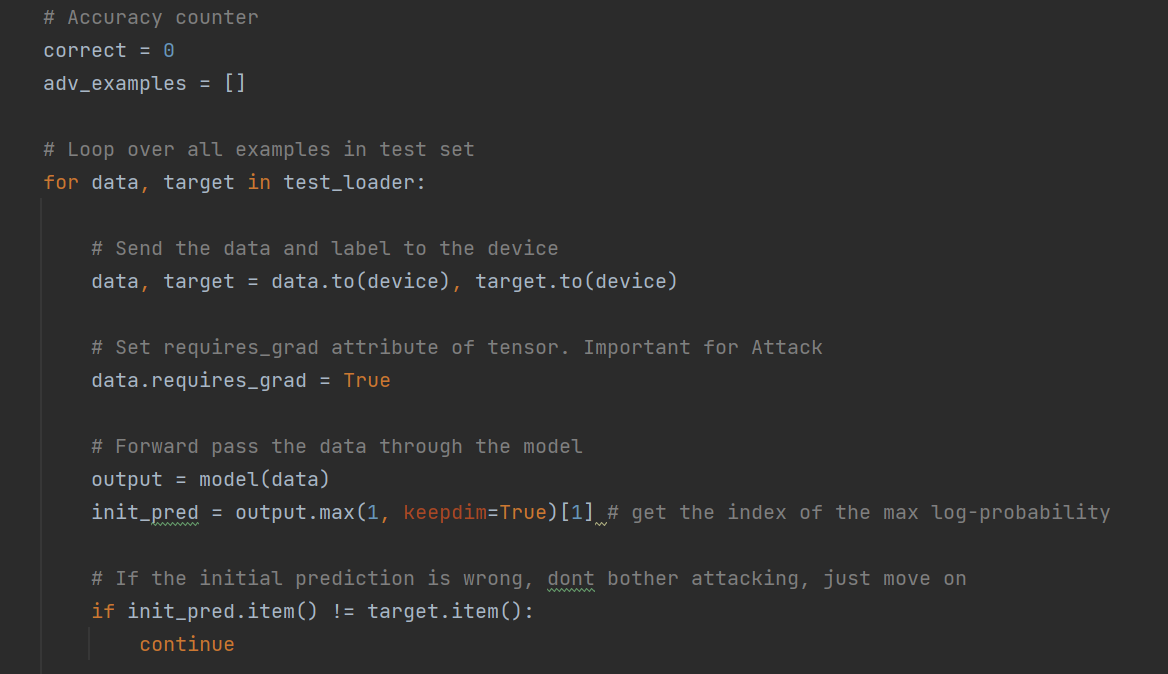






1. **功能测试**

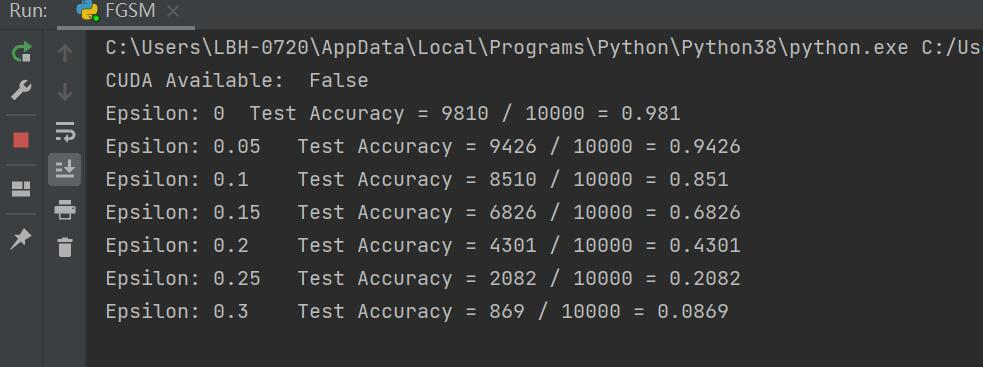
调用测试功能时需要输入epsilon参数，行完整的测试步骤，并报告最终准确性。



1. **运行攻击**

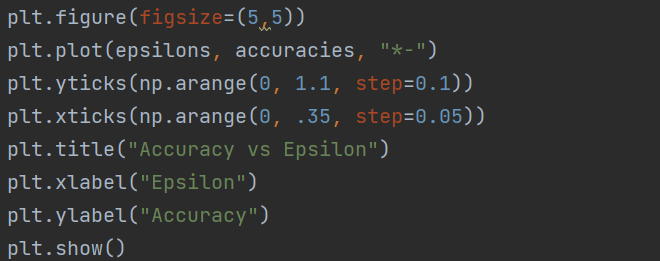
对数的每一个epsilon值运行一个完整的测试步骤，并保存最终精度。该值为0则意味着案列为原始精度，无攻击。

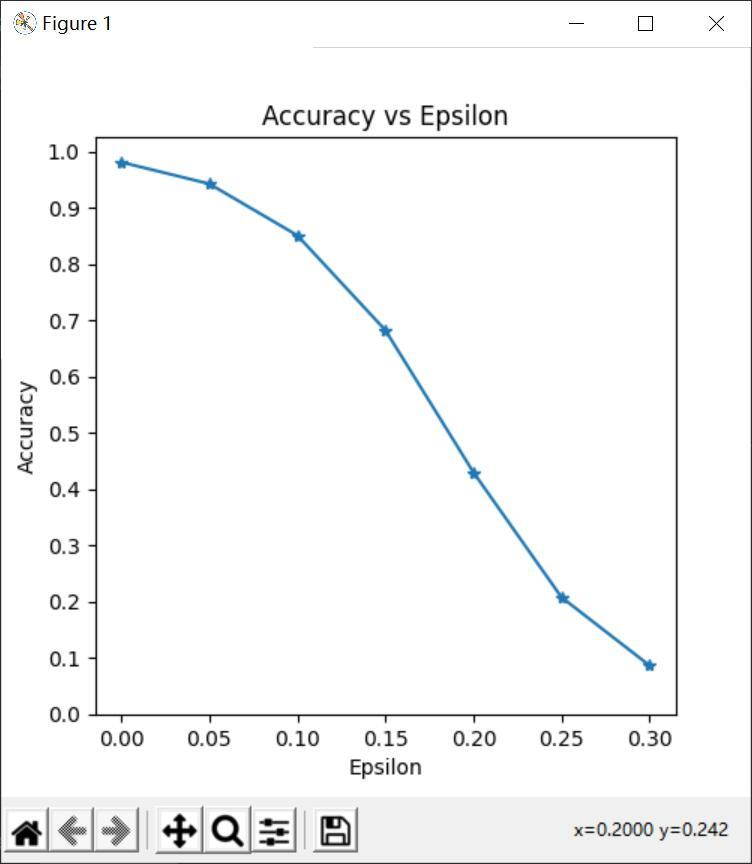
1. **实验结果**



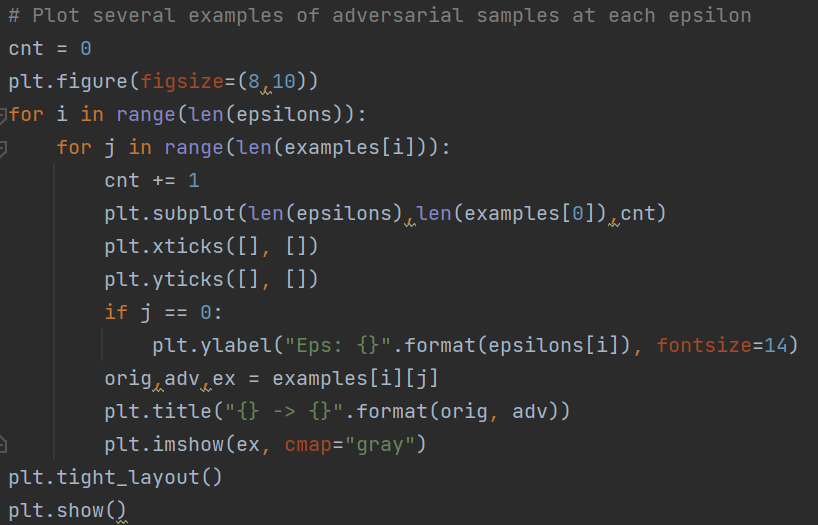
1. **实验结果的分析**

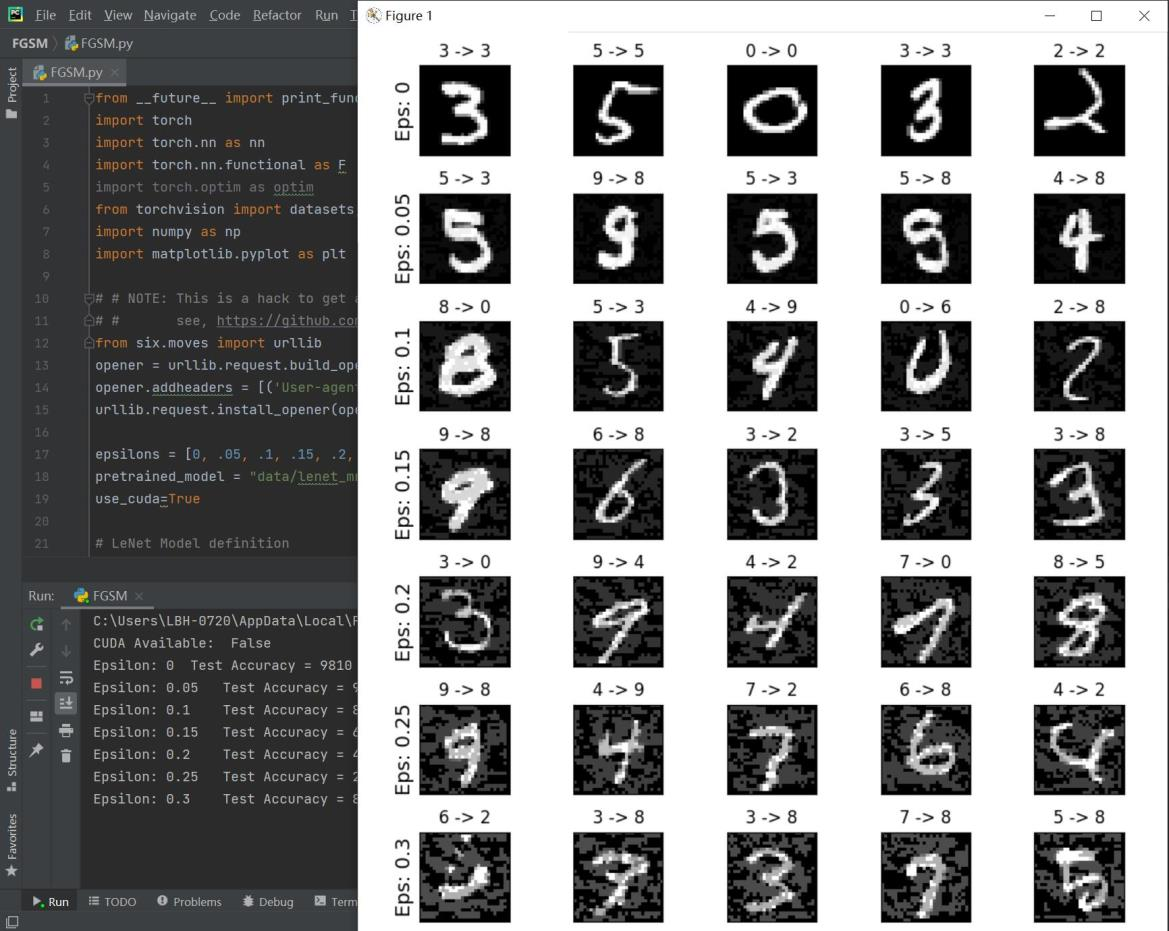
随着epsilon值的增加，我们预计测试精度会降低，因为随着该值增大，预测精度逐步降低。





**对抗示例**





**实验总结（完成心得与其它，主要讲自己碰到的问题和解决问题的方法）：**

实验前配置pytorch的时候，发现自己重做了系统之后所有环境变量都没了，于是我重新安装了python和anaconda，并且通过anaconda安装了pytorch。并且在参考老师给的网站时，代码方面没有搞太清楚，只能说是跑出来了结果，但是其中的原理还是不太清楚。

**参考文献（包括参考的书籍，论文，URL等，很重要）：**

[**Adversarial Example Generation — PyTorch Tutorials 1.11.0+cu102 documentation**](https://pytorch.org/tutorials/beginner/fgsm_tutorial.html)

[**安装PyTorch详细过程\_MC云鸷骚峰的博客-CSDN博客\_pytorch安装**](https://blog.csdn.net/MCYZSF/article/details/116525159)

[**https://blog.csdn.net/u014380165/article/details/90723948**](https://blog.csdn.net/u014380165/article/details/90723948)