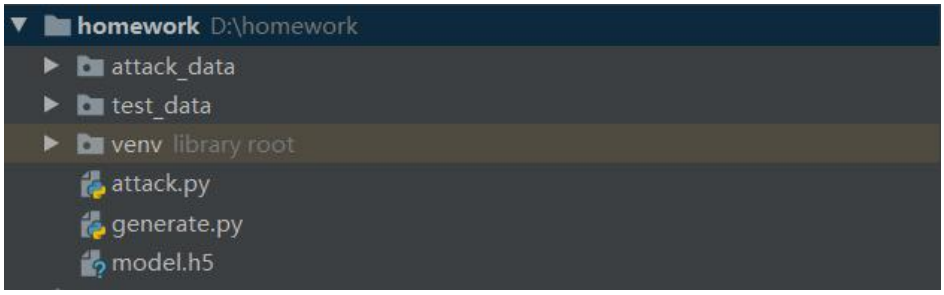
基于深度学习生成对抗样本项目说明

一、项目介绍

1. 项目代码结构如下




表格描述如下（红色为目录名，黑色为文件名）

| 文件/文件夹名 | 子目录/子文件 | 描述 |
|---|---|---|
| homework | attack test attack.py generate.py model.h5 | 项目根目录 |
| homework/test_data | test_data.py | test_data 目录存储 fashion-mnist 测试集文件 |
| homework/attack_data | attack_data.npy | attack_data 目录存储生成的对抗样本文件 |
| homework/attack.py | | attack.py 为对抗攻击生成对抗样本的代码 |
| homework/generate.py | | generate.py 提供了项目对外调用的接口 |
| homework/model.h5 | | 保存的自训练模型文件 |
| homework/attack_data/attack_data.npy | | 存储生成的前 1000 个对抗样本文件 |
| homework/test_data/test_data.npy | | Fashion-mnist 提供的测试集，保存了 10000 个样本 |

二、代码使用说明

**本项目使用 tensorflow1.15，numpy1.17.4。**使用 FGSM 算法生成对抗样本。

generate.py 为项目对外提供的运行接口，代码如下。

```
import attack
def generate(images,shape):
    if not(images.min()==0 and images.max==1):
        print("数据像素点取值范围不是[0,1],已将数据进行标准化")
        images.astype(float)
        images = images / float(images.max())
    return attack.attack_main(images,shape)
```

generate.py 中定义了 generate(images,shape)函数，此即为对外调用的接口。

generate 函数的两个参数分别为：

images: 测试样本集，类型为 numpy 数组，即 narray。

shape: 测试样本形状，类型为一维整数数组，遵循（n,28,28,1）格式。

generate 函数返回值为：

程序根据输入的 images 生成的对抗样本集，类型为 narray，形状与输入的 shape 相同,数据范围为[0,1]，即标准化数据。

在调用 generate(images,shape)前，需要将模型文件放置于项目根目录下，模型文件命名

为 model.h5。此模型类型为 tensorflow.keras 模型。程序会根据此模型生成对抗样本。在 attack.py 中将使用 model=tensorflow.keras.models.load_model(path)方法载入此模型，并使用 prediction=model.predict(image)方法进行预测。预测出的标签为 numpy.argmax(prediction)的返回值。

项目 attack_data 目录下保存了前 1000 条测试样本生成的 1000 条对抗样本，文件名为 attack_data.npy。本地生成 1000 条对抗样本用时 461.6 秒。

## 三、算法详解

项目使用 FGSM 算法生成对抗样本的方法为 attack.py 中的 fgsmAttack(image,model)方法。

对抗攻击是通过给增加微小扰动而使得模型分类错误的方法。根本思想实际上就是通过提供扰动，以增大模型的损失（loss），攻击的目的就是使得 loss 增大，直至导致对样本的分类错误。

FGSM 算法是基于梯度的攻击。它的主要思想就是在攻击时，使变化量与梯度变化方向一致，从而使 loss 函数有效的增大，导致分类结果发生变化。FGSM 算法之所以能够成功进行攻击，是因为对线性模型而言，微小的扰动会在神经网络中不断地被放大。

FGSM 的优点是多数情况下生成对抗样本的速度非常快，缺点则是对抗样本可迁移性不高，且当决策函数的梯度方向并非一直指向 loss 增大的方向（非线性模型）时会导致失效。

项目实现中主要是通过循环每次对样本进行微小的改动，直至预测结果发生变化。其中设置了修改上限和修改步长，以防止扰动过大，失去对抗攻击的意义。但问题在于算法中的学习率是人为设置定死的，而不能保证它的优越性。

同时，攻击成功（循环结束）的条件是对样本的预测结果发生改变，但不能保证攻击后的预测结果本身就是原有预测中概率较小的。举例说明，原有样本预测结果为，第 1 类 50%，第 2 类 30%，其余类别相加为 20%。而攻击成功可能只是使得预测结果从第 1 类变成第 2 类，但实际上这两类的概率本身就是相近的，攻击所造成的扰动实际上并没有那么大。

但若想避免上述问题，想让攻击变得更显著而使用定向攻击（攻击后结果为原有预测中概率最小的标签）或是使得正确标签在攻击后预测中的概率最小等手段则会导致图片变化过大或是生成对抗样本速度变的极慢。

其次，在线性模型上，FGSM 表现良好，但不同线性模型的梯度都有区别，所以，在使用 FGSM 生成的对抗样本对其他模型进行黑盒攻击时，攻击效果并不那么好。即对抗样本可迁移性较低。

## 四、个人感受

个人认为项目的难点在于算法的确定和理解。如果之前对机器学习了解不多的话，对于一些基础概念的理解是非常耗时的。同时由于缺乏对对抗攻击算法的了解，就算找到了一些实现好的算法也需要时间去理解和对各种 api 的功能查询。

这次项目中使用的 FGSM 算法是网络上比较流行，讲解和实现也比较多的算法。个人使用的感受就是优缺点都非常明显。优点在于相比于例如差分的其他算法，FGSM 的速度是非常快的，针对 fashion-mnist 生成一个对抗样本只需要不到一秒钟。FGSM 的缺点则是生成的对抗样本实际上并不够优秀，其对非线性模型的攻击效果比较差，且对于梯度不同的模型而言，FGSM 生成的对抗样本的可迁移性也不高。

总而言之，对 FGSM 使用让我体会到算法的效率和生成对抗样本的优越度是很难兼得的。FGSM 有较高的效率，生成速度快，但同时，生成的样本本身也有缺陷，不够优越。

本次项目个人最大的收获是对机器学习的一些基础概念有了了解。对 tensorflow 的使用初步入门。同时了解了一些热门的对抗攻击算法。

五、参考资料


1. FGSM（Fast Gradient Sign Method）_学习笔记+代码实现
https://blog.csdn.net/qq_35414569/article/details/80770121
2. 基于梯度的攻击——FGSM
https://www.cnblogs.com/tangweijqxx/p/10615950.html
3. Tensorflow 中文社区
http://www.tensorfly.cn/tfdoc/get_started/introduction.html
4. TensorFlow 学习（四）：梯度带(GradientTape)，优化器(Optimizer)和损失函数(losses)
https://blog.csdn.net/xierhacker/article/details/53174558
5. FGSM（Fast Gradient Sign Method）生成对抗样本（32）---《深度学习》
https://blog.csdn.net/u014038273/article/details/78773515
6. Github 仓库：fast-gradient-sign-method-tensorflow
https://github.com/yangdechuan/fast-gradient-sign-method-tensorflow