ReadMe

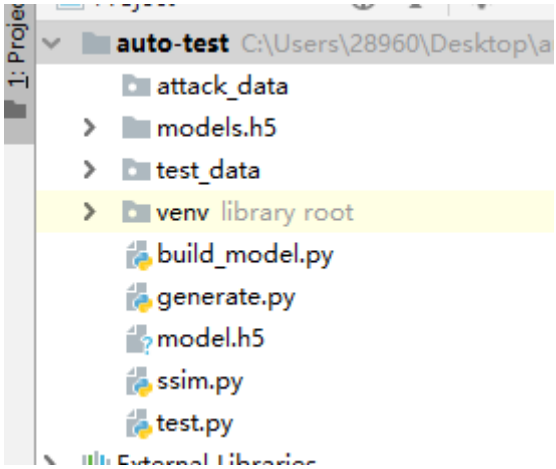
项目介绍

本项目是基于tensorflow生成针对fashion-mnist的对抗样本的的自动化测试方法。其代码结构拆分为build_model.py,generate.py,ssim.py以及生成好的模型model.h5和mode_faster.h5(为什么会有两个模型之后会有叙述)，训练前的test_data.npy和对应生成的对抗样本attack_data.npy,其位置均位于主目录下。其中generate.py中由generate方法可以直接输入参数图像集合和shape来直接调用，generate.py中含有生成10000张测试集的对抗样本的输入例子，即直接运行generate.py函数。生成10000张用例耗时70min。

而在其它的函数中，ssim.py通过调用SSIM方法来获取结构相似性的值，此外，build_model是我们生成模型的方法，test.py则是自己所用的进行评估的方法，这些方法都是直接运行，不存在方法调用。

本项目采用tensorflow2.0版本，劳烦助教调整，谢谢！

代码结构



算法详解

由于本次生成对抗样本的方法是黑盒攻击，因而主要是对于图像进行部分修改来达到“迷惑”训练好的模型的目的，便借鉴了Chuan Guo 等人发表的 Simple Black-box Adversarial Attacks中部分思想，决定采用在图像中随机添加“十”字的方法来进行攻击，并且通过加粗“十”字来进行对图像的二次乃至多次修改。（因为在实验中发现利用完全随机画“十”来进行多次修改的算法效果并不太好）

由于修改的方法是随机的，因而我们需要合理的验证修改是否满足达到“迷惑”的要求，因而我们通过已经生成好的model_fast.h5来进行预测新旧图像的预测标签是否相异，相比于另一个model.h5模型，这个模型采用了saved_model来避免内存的过高增长并提高运行速度。此外，我们也将比较新旧图像的结构相似性，即利用ssim.py中的SSIM方法来评估结构相似性，在ssim中，我们通过计算图像灰度级数（亮度对比函数），对比度对比函数和结构对比函数来进行联合判断。

在拥有了衡量的标准以后，我们的算法会对每一张图像执行20次修改尝试，攻击成功定义为模型会被“迷惑”且其与原图像的结构相似度超过75%，反之定义为攻击失败。其中攻击成功的图像会被缓存下来，而攻击失败的图像会进行更进一步的遍历，即通过尝试不同位置和颜色（至多500次）和十字的粗细程度（至多五次）来进行可能的成功试探。如果这些尝试依然失败，则代表我们攻击失败，会返回一张随机的图片来碰碰运气（划掉）。攻击成功的图像中，如果相似度大于90%则停止进一步尝试，并将攻击成功的部分图像中选举出最优解即相似度最高的那个作为结果返回。

在上述过程中，每张图片的尝试次数，进一步遍历深度以及十字粗细程度都可以修改来换取更短的时间或者更加优秀的对抗样本。

个人感受

本次作业是对机器学习的第一次尝试，虽然助教在课堂上讲了不少但是理解的真的不多……因而很多东西只能依靠自己去摸索。首先是在模型生成上，遇到的问题不算多，基本上按照博客上的内容按步都能成功构建我们自己的模型用来检测。之后是对于方法的探索，这是在本次作业中的第一个难点，首先是搜集资料+自我尝试，资料的搜集大多是一些难以看懂的对抗网络之类的算法……虽然尝试去理解但不懂的东西实在太多，另一方面是理解题意之后去对图像直接进行微调，比如random一个点直接改色之类，之后也尝试过学长的"直接返回算法"，但效果也不太理想，后来在发现了上面提到的文章之后收到了一点启发（当然也可能是我理解偏差），决定一次改变一个行向量一个列向量去进行调整即所说的画"十"字，得出的结果较为能够接受便采用这个办法。

第二个较难的点是如何去平衡结构相似度，攻击成功率和所耗时间的关系，也就是算法中那几个关键参数的设置，由于它们之间的互斥关系所以需要选择合理的值来区分，经过很长时间的探索决定采用了上面的值作为标准以达到较好的效果。下一步是ssim算法的书写，在看懂博客之后写起来问题并不是很大，主要是三个对比度的计算和乘积。

然后遇到的最大的问题是内存的泄露问题……开始由于不知道原因就眼睁睁看着内存一点点涨，一点点涨之后开始用虚存，然后磁盘都不够用，跑的越来越慢……起初一直找不到原因，因为自己所在循环里的东西既不需要大内存也不需要过多的运行时间，后来意识到在判断是否成功的时候都会调用模型的预测。既然模型是从文件读取的，会不会像其他文件一样是没有关闭什么的问题，但是在这个方向上一直无果。后来偶然，十分偶然，太偶然了，在搜索如何关闭的时候点进了一个如何存储的博客，在博客的末尾有一句工业上采用saved_model来进行存储，才发现自己存储的是整个模型，导致每次运行都会修改模型使得所占内存变大，之后采用saved_model来进行预测便节省了很多的时间和空间，其功能也恰好满足我们预测的需求。

总而言之这次作业让我学到了很多机器学习相关知识，但自己所写的算法里面所涉及到的专业的机器学习的相关知识并不多，反而觉得像是自己的"蛮力法"，这也是由于自己的知识域限制，在之后的学习过程中，也需要有意识地去接触相关理论的底层知识，来进一步增强自己能力。

参考文献

[1]Chuan Guo,Jacob R.Gardner.Simple Black-box Adversarial Attacks[cs.LG].arXiv:1905.07121

[2]极客兔兔.TensorFlow 2中文文档-保存与加载模型.https://geektutu.com/post/tf2doc-ml-basic-save-model.html

[3]_zZhe.tensorflow 2.0 keras 高层接口 之 模型的加载与保存.https://blog.csdn.net/z_feng12489/article/details/90754078

[4] ostartech.SSIM(structural similarity index)，结构相似性.https://www.cnblogs.com/wxl845235800/p/7692578.html

[5]爱扣脚的coder.利用TensorFlow进行Fashion MNIST数据集的基本分类问题.https://blog.csdn.net/m0_37393514/article/details/81010587