Week10实验

实验目标

上周我们探索的是朴素的数据投毒攻击，攻击者只能干预模型的训练集投毒。本周我们研究一种更强的投毒攻击：**后门攻击**。

后门攻击中，攻击者会干预模型的训练和测试两阶段：

- 在训练时投入一定比例的后门样本（在干净样本上贴上小块trigger，且将标签修改为target label）；
- 在测试时将trigger添加在干净图片上发起攻击。

本次实验主要是针对LeNet5模型，在MNIST数据集上进行后门攻击的实验，包括：

- 实现训练阶段的后门投毒；
- 在测试阶段，评估模型在干净样本上的预测准确率ACC，和干净样本贴上trigger的攻击成功率ASR；
- 调整不同的投毒比例，观察干净样本上的分类准确率ACC和后门攻击成功率ASR的变化情况。
- （bonus）实现Neural Cleanse、Strip其中的一种后门防御算法，并测试对后门攻击的防御效果。

实验步骤

本周实验包括一个后门攻击的任务，和一个Bonus（二选一）。

关于bonus，我们**不提供**任何bonus的基础代码，请有意向的同学自行从零开始基于pytorch实现，我们也会给出相关算法对应的paper原文，以及可参考的github上的代码（不一定是pytorch写的，可能是基于其他深度学习框架，请同学们**主要依靠对paper原文的理解**来做实现，github代码只是辅助作用）。

注意，后门防御算法实现中，都需加载已经被植入后门的模型。因此，**有意愿挑战bonus的同学，请在完成后门模型的训练后，将模型的参数进行持久化存储**，可自行尝试Task1中3种投毒比例的后门模型。（文件的存储与读取请自行查阅`torch.save()`和`torch.load()`函数）

任务一：后门攻击

- 根据notebook中的注释和要求，完成*TODO*内容

Bonus：后门防御 Neural Cleanse

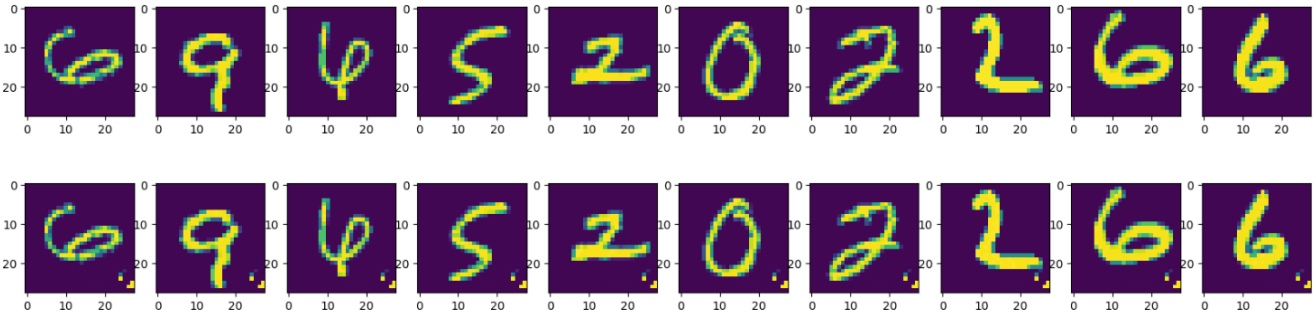
复现后门检测算法Neural Cleanse，来自论文 Neural cleanse: Identifying and mitigating backdoor attacks in neural networks（代码可参考 https://github.com/bolunwang/backdoor ）

请优先读懂原文的主要方法再考虑复现，并测试对后门攻击的防御效果。

我们作为防御者，这里的场景是：给定一个模型，我们不知道模型中是否存在后门，且不知道后门的target label或触发的trigger；**我们只能拿到干净训练集中10%的数据。**

给定一个已经植入后门的LeNet5模型，请依次做如下实现：

- 将MNIST的某一个标签类视作潜在的后门target label y_t，在10%的干净训练集上优化如下的目标得到逆向的trigger mask m 和trigger pattern Δ（逆向优化的过程中，请通过tanh函数保证trigger mask和trigger的取值范围都在[0, 1]范围之内）：$\min_{m,\Delta}\ \mathcal{L}(f(A(x,m,\Delta)),y_t)+\lambda\cdot|m|\ \text{for}\ x\in X$
- 对MNIST的所有标签类重复上述步骤，得到10个类的逆向trigger mask和pattern
- 实现原文中的MAD异常检测算法，基于10个类的逆向trigger mask找到后门的target label
- 对找到的target label类对应的逆向trigger做可视化，画图形式可参考下图（这是助教自己做的针对普通后门攻击的逆向效果）



- 将找到的target label类对应的逆向trigger当作真实trigger的近似，加在干净样本上，输入给模型、测试攻击成功率ASR
- 读取后门模型，在**10%的干净训练集上做继续训练**（要求使用如下设置：batch_size=128，Adam优化器学习率为0.001，训练5个epoch），评估模型的

ACC以及ASR

- **重新**读取后门模型，**在10%的干净训练集上做后门修复**，将这些样本中20%子集的图片加上逆向trigger、保持标签不变，做同上设置的继续训练，评估模型的ACC以及ASR

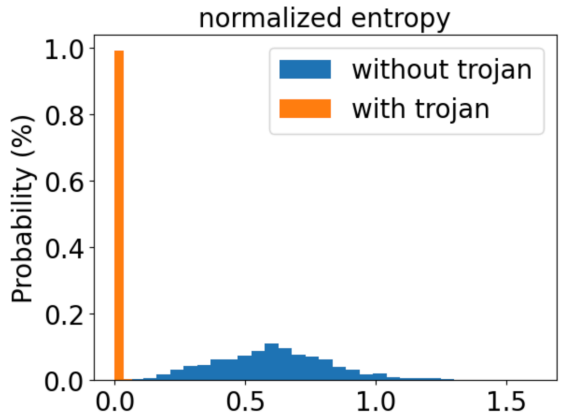
Bonus： 后门防御 STRIP

复现后门样本检测算法Strip，来自论文 Strip: A defence against trojan attacks on deep neural networks （代码可参考 https://github.com/garrisongys/STRIP ）

请优先读懂原文的主要方法再考虑复现，并测试对后门攻击的防御效果。

给定一个已经植入后门的LeNet5模型，请依次做如下实现：

- 实现两个样本做线性融合的操作，并可视化检查效果
- 分析2000个后门样本，每个与随机采样的100个干净样本做线性融合、输入给模型，分析得到2000个trojan entropy
- 分析2000个干净样本，每个与随机采样的100个干净样本做线性融合、输入给模型，分析得到2000个clean entropy
- 对统计得到的trojan entropy和clean entropy做可视化分析，分布直方图可参考下图来画：



- 基于clean entropy分析得到1%位置的阈值（ `scipy.stats.norm.ppf(0.01)` ），并计算FAR值（这些评估指标的具体含义请参考原文）

检查内容

- 任务一：

    - 后门样本可视化正确
    - 后门攻击: 3种投毒比例下，最佳效果的test_acc > 98.0、test_asr > 99.0%
    - 随着投毒比例的增大，ACC、ASR变化趋势正确
- Bonus：后门防御 Neural Cleanse
    - 能理解实现细节、阐述原文方法和核心原理
    - 能基于MAD异常值检测找到正确的后门target label
    - target label类的逆向trigger可视化效果接近真实trigger，且将逆向trigger当作近似trigger去攻击模型时，ASR超过90%
    - 与干净样本直接继续训练对比，基于逆向trigger的继续训练修复效果显著，修复后ACC高于90%，ASR低于30%
- Bonus：后门防御 STRIP
    - 能理解实现细节、阐述原文方法和核心原理
    - 线性融合可视化效果正确
    - trojan entropy和clean entropy分布图可视化效果正确，两个分布能明显区分开来
    - FAR值小于5%

附录

参考文献：

- Neural Cleanse: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks
- STRIP: Strip: A defence against trojan attacks on deep neural networks