可信机器学习 HW2 Report

王瑞环 2100013112

1 PGD Attack

Step 1. 产生随机噪声. 使用 `torch.Tensor.uniform_` 生成 $[-\varepsilon, \varepsilon]$ 上的均匀分布噪声 δ, 并做 `clamp` 操作使得 $\texttt{lower\_limit} \leq X_\delta = \delta + X \leq \texttt{upper\_limit}$.

Step 2. 更新 X_δ. 计算交叉熵损失损失 $\mathcal{L}$, 得到新的 $X'_\delta = \texttt{clamp}\left(X_\delta + \alpha \text{sign}\left(\frac{\partial \mathcal{L}}{\partial X_\delta}\right)\right)$. 此时 `clamp` 应满足两个条件: $\texttt{lower\_limit} \leq X'_\delta \leq \texttt{upper\_limit}$ 和 $-\varepsilon \leq X'_\delta - X \leq \varepsilon$.

Step 3. 得到结果. 反复执行 **Step 3.** , 直至循环次数达到设定值. 取 `restarts` 次执行的过程中使得最终的 $\mathcal{L}$ 最大的 δ 为返回值.

2 C&W Attack

代码所需补全的主要部分为计算 loss 并进行梯度下降, 后者可以直接使用 `loss.backward()` 和 `optimizer.step()` 实现, 故接下来只需实现 Untargeted Attack 的损失函数.

记 $\texttt{scaler}(\text{arctanh}(X)+\delta)=X'$. 用 $Z(X)$ 表示模型 (去掉最后的 Softmax 层) 的输出. 实验中使用 L_2 Attack, 即 $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{margin}} + \mathcal{L}_{L_2}$.

2.1 Margin Loss

$\mathcal{L}_{\text{margin}} = \max(Z(X')_y - \max\{Z(X')_i : i \neq y\}, -\kappa)$, 其中 y 为 X 的正确分类. 最小化 Margin Loss 的目的在于使得非 y 类的分数与 y 类分数差尽可能大, 也即使得样本容易被误判为其他类别.

2.2 L_2 Loss

$\mathcal{L}_{L_2} = ||X' - X||_2^2 = \sum (X' - X)^2$. 最小化 L_2 Loss 的目的在于使得对抗样本与原始图片差距尽可能小.

3 Defense

3.1 AT Algorithm

作业原始代码的 AT 策略为: 在训练过程中, 将 Loss 的算法由直接计算原数据交叉熵损失 $\mathcal{L}_{\text{nat}}$, 改为计算由 PGD 生成的对抗样本的交叉熵损失 $\mathcal{L}_{\text{adv}}$.

参考论文Theoretically Principled Trade-off between Robustness and Accuracy提出的 TRADES 方法, 相较于原始的 AT, 将 Loss 改写为 $\mathcal{L} = \text{CrossEntropy}(p(X), y) + \beta \text{KL}(p(X)||p(X'))$.

3.2 Training Strategy

作业原始代码为两轮训练都进行 AT, 可将其修改为第一轮正常训练得到一个准确率稍高的 model, 第二轮再使用 AT, 可以在保持效果的同时加快训练速度.

3.3 Models

作业原始代码使用的模型为 ResNet18, 可使用 WideResNet(参数量 46.2M) 代替.

3.4 Results

实验中, 各超参数设置除将 batch size 设置为 64 外, 其余各项与作业源代码一致.

Method	Acc(%)	Rob(%)	Score=Acc/2+Rob(%)
AT	38.39	26.57	45.77
AT+DiffTrain	40.26	25.92	46.05
AT+WideResNet+DiffTrain	**49.45**	**30.03**	**54.76**
TRADES ($\beta = 6.0$)	42.37	24.47	45.66
TRADES+DiffTrain ($\beta = 6.0$)	42.04	23.24	44.26
TRADES+DiffTrain ($\beta = 1.0$)	46.93	17.68	41.15
TRADES+WideResNet+DiffTrain ($\beta = 6.0$)	47.64	26.50	50.32

3.5 Analysis & Conclusions

- 同一模型准确率与鲁棒性大致成负相关关系
- 使用先正常训练再做 AT 的训练模式也可以达到直接做 AT 的效果
- TRADES 中的参数 β 增大时, 确实有提高鲁棒性的作用, 说明参数 β 是合理有效的
- 在小批次训练中, AT 的变体 (TRADES) 不一定比直接 AT 表现更好, 猜测可能与算法收敛速度有关