Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks

Ambra Demontis[†], Marco Melis[†], Maura Pintor[†], Matthew Jagielski[*], Battista Biggio[†,‡], Alina Oprea[*], Cristina Nita-Rotaru[*], and Fabio Roli[†,‡]

[†]Department of Electrical and Electronic Engineering, University of Cagliari, Italy
[‡]Pluribus One, Italy
[*]Northeastern University, Boston, MA, USA

USENIX(2019)

Optimization Framework for Gradient-based Attacks

给定攻击者对 $K \in \kappa$ 的了解，攻击样本$x' \in \Phi(x)$ 及其标签 y，攻击者的目标可以用目标函数 $A(x', y, \kappa) \in R$(例如损失函数)来定义

$$\mathbf{x}^{\star} \in \underset{\mathbf{x}' \in \Phi(\mathbf{x})}{\arg\max}\, \mathcal{A}(\mathbf{x}', y, \kappa).$$

Attack Algorithm

Algorithm 1 Gradient-based Evasion and Poisoning Attacks

Input: $\mathbf{x}, y$: the input sample and its label; $\mathcal{A}(\mathbf{x}, y, \boldsymbol{\kappa})$: the attacker's objective; $\boldsymbol{\kappa} = (\mathcal{D}, \mathcal{X}, f, \mathbf{w})$: the attacker's knowledge parameter vector; $\boldsymbol{\Phi}(\mathbf{x})$: the feasible set of manipulations that can be made on $\mathbf{x}$; $t > 0$: a small number.

Output: $\mathbf{x}'$: the adversarial example.

1: Initialize the attack sample: $\mathbf{x}' \leftarrow \mathbf{x}$

2: **repeat**

3: Store attack from previous iteration: $\mathbf{x} \leftarrow \mathbf{x}'$

4: Update step: $\mathbf{x}' \leftarrow \boldsymbol{\Pi}_{\boldsymbol{\Phi}}\left(\mathbf{x} + \boldsymbol{\eta} \nabla_{\mathbf{x}} \mathcal{A}(\mathbf{x}, y, \boldsymbol{\kappa})\right)$, where the step size $\boldsymbol{\eta}$ is chosen with line search (bisection method), and $\boldsymbol{\Pi}_{\boldsymbol{\Phi}}$ ensures projection on the feasible domain $\boldsymbol{\Phi}$.

5: **until** $\left|\mathcal{A}(\mathbf{x}', y, \boldsymbol{\kappa}) - \mathcal{A}(\mathbf{x}, y, \boldsymbol{\kappa})\right| \leq t$

6: **return** $\mathbf{x}'$

Evasion Attacks

在规避攻击中，攻击者操纵测试样本以使其错误分类，即逃避学习算法的检测。

white-box:

$$\begin{aligned} \max_{\mathbf{x}'} \quad & \ell(y, \mathbf{x}', \mathbf{w}), \\ \text{s.t.} \quad & \|\mathbf{x}' - \mathbf{x}\|_p \leq \varepsilon, \\ & \mathbf{x}_{\text{lb}} \preceq \mathbf{x}' \preceq \mathbf{x}_{\text{ub}}, \end{aligned}$$

black-box:

对于黑盒案例，使用代理模型 ˆf 的参数 ˆ w 就足够了。

Transferability Definition and Metrics

假设有一个来自于替代模型的攻击样本$x' = x + \hat{\delta}$，将这个攻击样本迁移到目标模型，迁移性采用损失函数进行刻画：

$$\mathrm{T} = \ell(\mathrm{y}, \mathbf{x} + \hat{\delta}, \mathbf{w})$$

进行线性化近似，可以把$\hat{\delta}$拆出来:

$$\mathrm{T} = \ell(\mathrm{y}, \mathbf{x} + \hat{\delta}, \mathbf{w}) \cong \ell(\mathrm{y}, \mathbf{x}, \mathbf{w}) + \hat{\delta}^{\top} \nabla_{\mathbf{x}} \ell(\mathrm{y}, \mathbf{x}, \mathbf{w})$$

Size of Input Gradients

目标模型损失函数关于样本的梯度，梯度越大模型越容易受到迁移攻击

$$S(\mathbf{x}, y) = \|\nabla_{\mathbf{x}} \ell(y, \mathbf{x}, \mathbf{w})\|_q ,$$

此处的q范数是扰动约束p范数的对偶范数

Gradient Alignment

在黑盒情况下，攻击者掌握的替代模型和目标模型之间的区别：

$$R(\mathbf{x}, y) = \frac{\nabla_{\mathbf{x}} \hat{\ell}^{\top} \nabla_{\mathbf{x}} \ell}{\|\nabla_{\mathbf{x}} \hat{\ell}\|_2 \|\nabla_{\mathbf{x}} \ell\|_2}$$

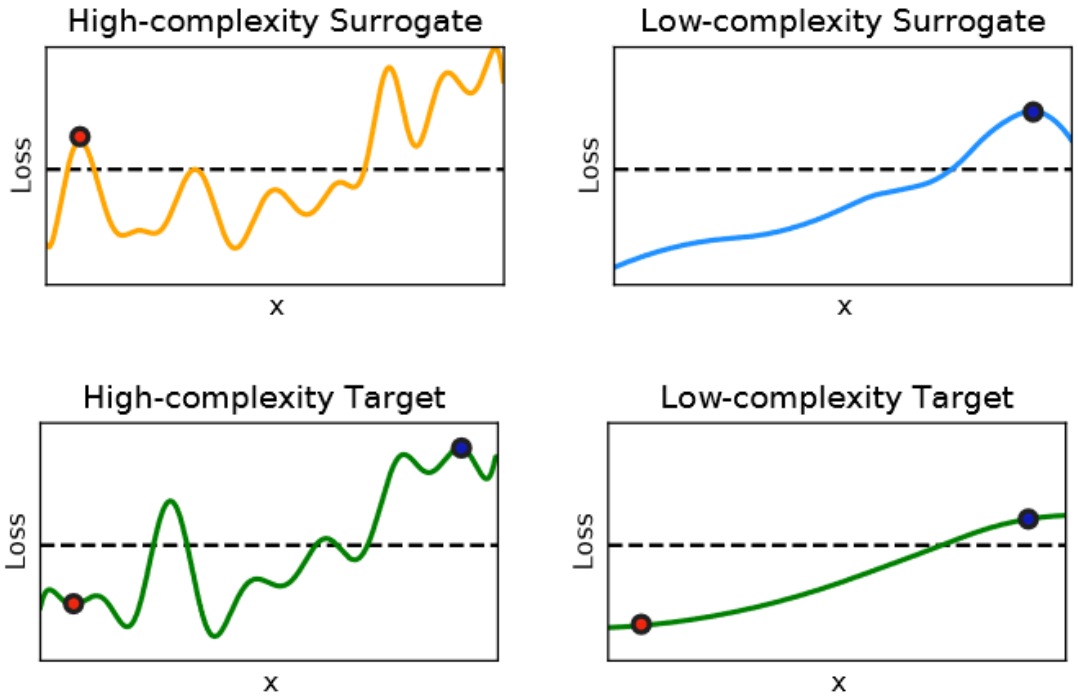
表征对抗攻击的可转移性，余弦相似度越高，迁移性越好

Variability of the Loss Landscape

更稳定且方差更低的替代模型的损失函数往往有利于基于梯度的优化攻击，以找到更好的局部最优

$$V(\mathbf{x}, y) = \mathbb{E}_{\mathcal{D}}\{\ell(y, \mathbf{x}, \hat{\mathbf{w}})^2\} - \mathbb{E}_{\mathcal{D}}\{\ell(y, \mathbf{x}, \hat{\mathbf{w}})\}^2,$$

E_D 是对不同替代采用的训练集的期望。



Experimental Analysis

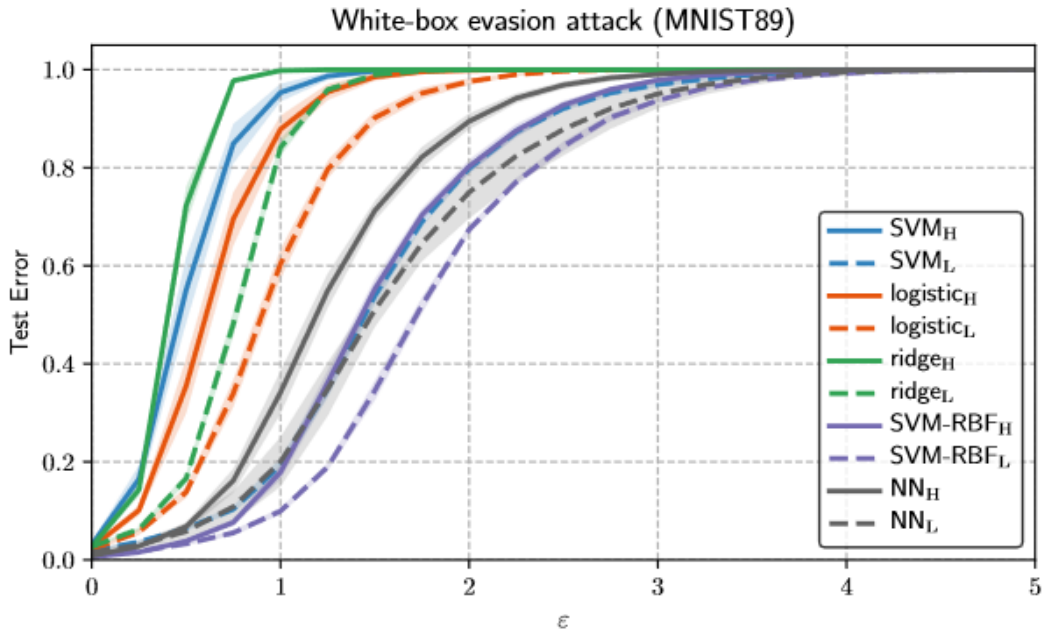
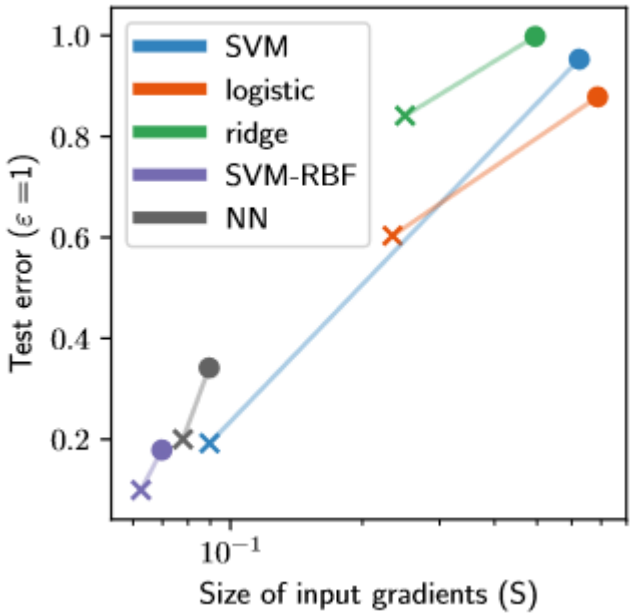


Figure 5: White-box evasion attacks on MNIST89. Test error against increasing maximum perturbation ε.


SVM
logistic
ridge
SVM-RBF
NN
1.0
0.8
0.6
0.4
0.2
Test error (ε =1)
10^{-1}
Size of input gradients (S)