Neural Tangent Generalization Attacks

Chia-Hung Yuan, Shan-Hung Wu
National Tsing Hua University

Introduction & Motivation

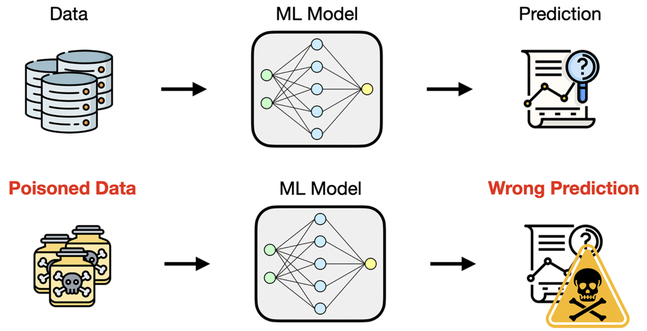
The remarkable performance achieved by Deep Neural Networks (DNNs) in many applications is followed by the rising concern about **data privacy** and **security**.

Motivation: **Is it possible to prevent a DNN model from learning on given data?**

Our work: We propose the Neural Tangent Generalization Attack (NTGA) to reach this goal, which is the first work enabling **clean-label**, **black-box** generalization attack against DNNs.

Background

- **Poisoning attacks**: Disrupting a training procedure to output a defective models by poisoning certain amount of training data.
- **Generalization attacks**: One type of the poisoning attacks, the goal is to decline the generalization ability of the model.



Generalization attacks can be formulated as a **bilevel optimization** problem:

$$\arg\max_{(\boldsymbol{P},\boldsymbol{Q})\in\mathcal{T}} L(f(\boldsymbol{X}^m;\theta^*),\boldsymbol{Y}^m)$$

$$\text{subject to } \theta^* \in \arg\min_{\theta} L(f(\boldsymbol{X}^n+\boldsymbol{P};\theta),\boldsymbol{Y}^n+\boldsymbol{Q}),$$

where $\mathbb{D}=(\boldsymbol{X}^n\in\mathbb{R}^{n\times d},\boldsymbol{Y}^n\in\mathbb{R}^{n\times c})$ and $\mathbb{V}=(\boldsymbol{X}^m,\boldsymbol{Y}^m)$ are training and validation sets of n and m examples. $f(\cdot\,;\theta)$ is a model parameterized by θ. $\boldsymbol{P}$ and $\boldsymbol{Q}$ are perturbations to be added to $\mathbb{D}$.

Challenge

Solving the bilevel problem by gradient ascent suffers from the **high-order differential** issues.

- It can be solved exactly and efficiently only when the learning model is **convex**, which is not transferable to non-convex DNNs.
- Some works solve the relaxations of the bilevel problem with a **white-box** assumption.

Challenge: Efficient computing of a black-box, clean-label generalization attack against DNNs remains an open problem.

Neural Tangent Generalization Attacks

Problems: 1. Solve the bilevel problem efficiently against a non-convex model f.
2. Let f be a "representative" surrogate of the unknown target models.

Solutions: 1. We let f be the mean of **a Gaussian Process(GP) with a Neural Tangent Kernel** that approximates the training dynamics of a class of wide DNNs.
2. The GPs behind NTGA model the evolution of an **infinite ensemble** of **infinite-width** networks.

At time step t during the gradient descent training, the mean prediction of the GP $\bar{f}$ over $\mathbb{V}$ evolves as:

$$\bar{f}(\boldsymbol{X}^m;\boldsymbol{K}^{m,n},\boldsymbol{K}^{n,n},\boldsymbol{Y}^n,t)=\boldsymbol{K}^{m,n}(\boldsymbol{K}^{n,n})^{-1}(\boldsymbol{I}-e^{\eta\boldsymbol{K}^{n,n}t})\boldsymbol{Y}^n,$$

where $\boldsymbol{K}^{n,n}\in\mathbb{R}^{n,n}$ and $\boldsymbol{K}^{m,n}\in\mathbb{R}^{m,n}$ are kernel matrices. In particular, $K^{n,n}_{i,j}=k(x^i\in\mathbb{D},x^j\in\mathbb{D})$, $K^{m,n}_{i,j}=k(x^i\in\mathbb{V},x^j\in\mathbb{D})$ and $k(\cdot,\cdot)$ is a Neural Tangent Kernel, which measures the similarity between two data points from the network class' point of view.

This allows us to rewrite the bilevel optimization as a more straightforward problem

$$\arg\max_{\boldsymbol{P}\in\mathcal{T}} L(\bar{f}(\boldsymbol{X}^m;\hat{\boldsymbol{K}}^{m,n},\hat{\boldsymbol{K}}^{n,n},\boldsymbol{Y}^n,t),\boldsymbol{Y}^m),$$

where $\hat{\boldsymbol{K}}^{n,n}$ and $\hat{\boldsymbol{K}}^{m,n}$ are kernel matrices built on the poisoned training data $\boldsymbol{X}^n+\boldsymbol{P}$. $\mathcal{T}$ is threat model controls the allowable values of perturbations.

Experiments

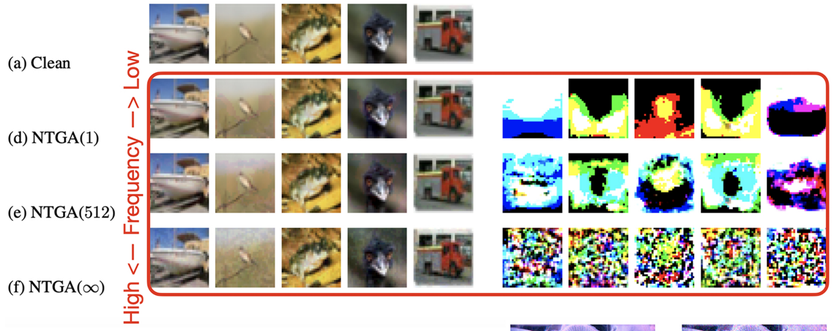
Black-box attacks: An attacker knows nothing about the target models, including architectures, optimization methods, loss functions, etc.

	MNIST	CIFAR-10	2-class ImageNet
Clean	99.5%	92.7%	98.4%
RFA[1]	87.0%	88.8%	90.4%
DeepConfuse[2]	46.2%	55.0%	92.8%
NTGA	**15.6%**	**37.8%**	**72.8%**
	+57.4%	**+45.6%**	**+220.0%**

[1]Chan-Hon-Tong, 2019
[2]Feng et al., 2019

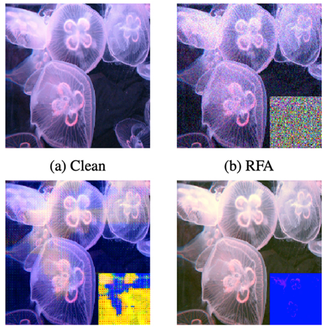
NTGA is **107.7% more effective** than the baselines, while taking **96.5% less time** to generate the poisoned data

Visualization: The hyperparameter t controls how an attack looks. NTGA(•) denotes an attack with a specific time step t.



Visualization of some poisoned images and their normalized perturbations are shown. **Smaller t leads to simpler perturbations**. It is consistent with the previous findings that a network tends to learn low-frequency patterns at the early stage of training.

As we can see, the only difference between the clean and poisoned images generated by NTGA is the hue!


(a) Clean
(b) RFA
(c) DeepConfuse
(d) NTGA(1)