

Phương pháp tìm kiếm lân cận rộng thích ứng cho một số lớp bài toán định tuyến phương tiện

Nguyễn Mạnh Linh

Khoa Toán-Cơ-Tin học
Đại học Khoa học Tự nhiên

2023/12

Outline

Giới thiệu

Thực trạng

- Khai phá dữ liệu là một chu cầu thiết yếu
- Lượng dữ liệu khổng lồ được sinh ra mỗi ngày

Tính toán song song và phân tán là một cách tiếp cận hiệu quả để khai phá dữ liệu

Giới thiệu - Các loại tấn công

- 1 Tấn công nhắm đích - *targeted attacks*
- 2 Tấn công không nhắm đích - *untargeted attacks*

Giới thiệu: Huấn luyện đối nghịch

Huấn luyện đối nghịch (*adversarial training*) (Madry et al. 2017): Sử dụng mẫu đối nghịch để huấn luyện một mô hình mạnh có khả năng chống chịu với các nhiễu của mẫu đối nghịch.

Giới thiệu: Tấn công phân loại ảnh

- ❶ Tấn công phân loại ảnh dựa trên mạng neuron tích chập
- ❷ Mẫu đối nghịch được tạo ra để làm sai lệch kết quả phân loại
- ❸ Mẫu mới được tạo ra (gần) giống với mẫu gốc

Giới thiệu: Độ nhiễu

$\|x\|_q = (\sum_{i=1}^p |x_i|^q)^{\frac{1}{q}}$ kí hiệu chuẩn L_q của vector p chiều
 $x = [x_1, \dots, x_p]$ với $q \geq 1$

- L_∞ : Đánh giá sự thay đổi tối đa các pixel (oodfellow, Shlens, and Szegedy 2015)
- L_2 : Cải thiện chất lượng hình ảnh (Carlini and Wagner 2017b)
- L_1 : Sử dụng trong các bài toán phục hồi ảnh (Fu et al. 2006)

Trong bài toán mẫu đối nghịch, độ biến dạng L_1 đánh giá tổng các thay đổi trong nhiễu loạn và đóng vai trò là một thành phần (hàm) lỗi đo lường số lượng pixel thay đổi (độ thưa) gây ra bởi nhiễu

Giới thiệu - Các tập dữ liệu thử nghiệm

Tấn công dựa trên L_1 với các tập dữ liệu

- MNIST
- CIFAR-10
- ImageNet

Được so sánh với các tấn công dựa trên L_2 và L_∞

Outline

Tấn công DNNs - FGM, I-FGM

- Kí hiệu \mathbf{x}_0 và \mathbf{x} lần lượt là mẫu gốc và mẫu đối nghịch, t là lớp mục tiêu cần tấn công.
- Tấn công dựa trên L_∞

$$\mathbf{x} = \mathbf{x}_0 - \epsilon \times \text{sign}(\nabla J(\mathbf{x}_0, t)) \quad (1)$$

với ϵ là độ biến dạng L_∞ giữa \mathbf{x} và \mathbf{x}_0 và $\text{sign}(\nabla J)$ là dấu của gradient

- Tấn công dựa trên L_1 và L_2

$$\mathbf{x} = \mathbf{x}_0 - \epsilon \frac{\nabla J(\mathbf{x}_0, t)}{\|\nabla J(\mathbf{x}_0, t)\|_q} \quad (2)$$

với $q = 1, 2$ và ϵ là độ méo tương quan

Tấn công DNNs - C&W

- Thay vì sử dụng hàm mất mát trên tập huấn luyện Carlini và Wagner đã thiết kế một hiệu chỉnh L_2 trong hàm mất mát dựa trên lớp logit trong DNNs để sinh ra các mẫu đối nghịch (Carlini and Wagner 2017b)
- Công thức này hóa ra là một trường hợp riêng của thuật toán EAD (sẽ được trình bày trong phần sau)

Phòng thủ DNNs

- Defensive distillation - Chứng cất phòng thủ (Papernot et al.2016b)
- Adversarial training - Huấn luyện đối nghịch (Zheng et al. 2016; Madry et al. 2017; Tram'er et al. 2017; Zantedeschi, Nicolae, and Rawat 2017)
- Detection methods - Phương pháp dò tìm (Feinman et al. 2017; Grosse et al. 2017; Lu, Issaranon, and Forsyth 2017; Xu, Evans, and Qi 2017)

Outline

EAD - Tóm lược

- Hiệu chỉnh elastic-net là công nghệ được sử dụng rộng rãi trong việc giải quyết các bài toán lựa chọn thuộc tính nhiều chiều (Zou and Hastie 2005)
- Nhìn chung, hiệu chỉnh elastic-net được sử dụng trong bài toán cực tiểu hóa sau đây:

$$\text{minimize}_{\mathbf{z} \in \mathcal{Z}} f(\mathbf{z}) + \lambda_1 \|\mathbf{z}\|_1 + \lambda_2 \|\mathbf{z}\|_2^2 \quad (3)$$

Trong đó \mathbf{z} là vector của p biến tối ưu, \mathcal{Z} là tập nghiệm chấp nhận được, $f(\mathbf{z})$ là hàm mất mát, $\|\mathbf{z}\|_q$ là chuẩn q của \mathbf{z} và $\lambda_1, \lambda_2 \geq 0$ tương ứng là các tham số hiệu chỉnh L_1 và L_2

- Biểu thức $\lambda_1 \|\mathbf{z}\|_1 + \lambda_2 \|\mathbf{z}\|_2^2$ được gọi là hiệu chỉnh elastic-net của \mathbf{z}

EAD - Xây dựng

Cho trước 1 ảnh \mathbf{x}_0 và nhãn đúng của nó là t_0 , gọi \mathbf{x} là mẫu đối nghịch của \mathbf{x}_0 với lớp đích nhắm đến là $t \neq t_0$. Hàm mất mát $f(\mathbf{x})$ cho tấn công nhắm đích là:

$$f(\mathbf{x}, t) = \max \left(\max_{j \neq t} [\mathbf{Logit}(\mathbf{x})]_j - [\mathbf{Logit}(\mathbf{x})]_t, -\kappa \right) \quad (4)$$

Trong đó $\mathbf{Logit}(\mathbf{x}) = [\mathbf{Logit}(\mathbf{x})_1, \dots, \mathbf{Logit}(\mathbf{x})_K] \in \mathbb{R}^K$ là lớp logit (lớp trước softmax) biểu diễn cho \mathbf{x} trong mạng DNN, K là số lượng lớp cần phân loại, $\kappa > 0$ là tham số tin cậy, nó đảm bảo một khoảng cách cố định giữa $\max_{j \neq t} [\mathbf{Logit}(\mathbf{x})]_j$ và $[\mathbf{Logit}(\mathbf{x})]_t$.

EAD - Xây dựng

Thành phần $[\mathbf{Logit}(x)]_t$ là xác suất dự đoán x có nhãn t theo luật phân loại của hàm softmax:

$$\text{Prob}(\text{Label}(\mathbf{x}) = t) = \frac{\exp([\mathbf{Logit}(\mathbf{x})]_t)}{\sum_{j=1}^K \exp([\mathbf{Logit}(\mathbf{x})]_j)} \quad (5)$$

EAD - Xây dựng

Do đó, hàm mất mát trong phương trình (??) có mục đích là để cho ra nhãn t là lớp có xác suất cao nhất của \mathbf{x} và tham số κ đảm bảo sự phân biệt giữa lớp t và lớp dự đoán gần nhất khác với t . Với tấn công không nhắm mục tiêu, hàm mất mát trong phương trình ?? trở thành:

$$f(\mathbf{x}) = \max \left([\mathbf{Logit}(\mathbf{x})]_{t_0} - \max_{j \neq t} [\mathbf{Logit}(\mathbf{x})]_j, -\kappa \right) \quad (6)$$

EAD - Xây dựng

Hiệu chỉnh elastic-net còn tạo ra mẫu đối nghịch tương tự với ảnh gốc. Công thức tấn công elastic-net vào mạng DNNs (EAD) để tạo ra mẫu đối nghịch (\mathbf{x}, t) cho ảnh gốc (\mathbf{x}_0, t_0) như sau:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad c \times f(\mathbf{x}, t) + \beta \|\mathbf{x} - \mathbf{x}_0\|_1 + \|\mathbf{x} - \mathbf{x}_0\|_2^2 \\ & \text{st } \mathbf{x} \in [0, 1]^p \end{aligned} \tag{7}$$

Với $f(\mathbf{x}, t)$ được xác định trong phương trình (??), $c, \beta \geq 0$ lần lượt là các tham số hiệu chỉnh của hàm mất mát f và hàm phạt L_1 .

Thuật toán 1 Tấn công Elastic-net vào DNNs (EAD)

Input: Ảnh gốc và nhãn của nó (\mathbf{x}_0, t_0) , lớp mục tiêu t , tham số chuyển giao κ , tham số hiệu chỉnh β , độ dài bước α_k , số bước lặp I

Output: mẫu đối nghịch \mathbf{x}

Khởi tạo: $\mathbf{x}^{(0)} = \mathbf{y}^{(0)} = \mathbf{x}_0$

for $k = 0$ to $I - 1$ **do**

$$\begin{aligned}\mathbf{x}^{(k+1)} &= S_{\beta}(\mathbf{y}^{(k)} - \alpha_k \nabla g(\mathbf{y}^{(k)})) \\ \mathbf{y}^{(k+1)} &= \mathbf{x}^{(k+1)} + \frac{k}{k+3}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})\end{aligned}$$

end for

Luật quyết định: tìm \mathbf{x} từ tập các mẫu thành công trong $\{\mathbf{x}^k\}_{k=1}^I$ (luật EN, luật L_1).

EAD - Thuật toán

Trong đó

$$[S_{\beta}(\mathbf{z})]_i = \begin{cases} \min\{\mathbf{z}_i - \beta, 1\} & \text{nếu } \mathbf{z}_i - \mathbf{x}_{0i} > \beta; \\ \mathbf{x}_{0i} & \text{nếu } |\mathbf{z}_i - \mathbf{x}_{0i}| \leq \beta; \\ \max\{\mathbf{z}_i + \beta, 0\} & \text{nếu } \mathbf{z}_i - \mathbf{x}_{0i} < -\beta \end{cases} \quad (8)$$

Với $i \in \{1, \dots, p\}$. Nếu $|\mathbf{z}_i - \mathbf{x}_{0i}| > \beta$, thành phần \mathbf{z}_i được co lại với hệ số β và chiếu thành phần kết quả lên miền ràng buộc chấp nhận được thuộc đoạn $[0, 1]$.

Outline

Thực nghiệm: Dữ liệu và phương pháp

- Tập dữ liệu: MNIST, CIFAR10, ImageNet
 - MNIST và CIFAR10 được huấn luyện trên mô hình DNN bởi Carlini và Wagner
 - ImageNet sử dụng mô hình InceptionV3
- Phương pháp tấn công:
 - EAD
 - C&W
 - FGM
 - I-FGM
- Phần cứng: Intel E5-2690 v3 CPU, 40 GB RAM, NVIDIA K80 GPU


Thực nghiệm: Độ đo

- ASR: Tỷ lệ tần công thành công
- L_1 , L_2 và L_∞ : Các khoảng cách giữa mẫu đối nghịch và ảnh gốc

Thực nghiệm: Các trường hợp quan tâm

- **Trường hợp tốt nhất (best case):** tấn công dễ nhất về phương diện nhiều, trong số các tấn công nhắm tới tất cả các lớp sai nhãn.
- **Trường hợp trung bình (average case):** tấn công nhắm ngẫu nhiên vào 1 lớp sai nhãn.
- **Trường hợp xấu nhất (worst case):** tấn công khó nhất về phương diện nhiều, trong số những tấn công nhắm tới tất cả các lớp sai nhãn.

Thực nghiệm: Độ nhạy




images/tab_4_1.png

Thực nghiệm: Luật quyết định


images/fig_02.png

Thực nghiệm: ASR, nhiễu trên MNIST, CIFAR10, ImageNet




images/tab_4_2.png

Thực nghiệm: MNIST

images/fig_05.png


Hình 2: Các mẫu đối nghịch được sinh bởi thuật toán EAD trên tập MNIST

Thực nghiệm: CIFAR10

images/fig_06.png

Hình 3: Các mẫu đối nghịch được sinh bởi thuật toán EAD trên tập CIFAR10

Thực nghiệm: ImageNet

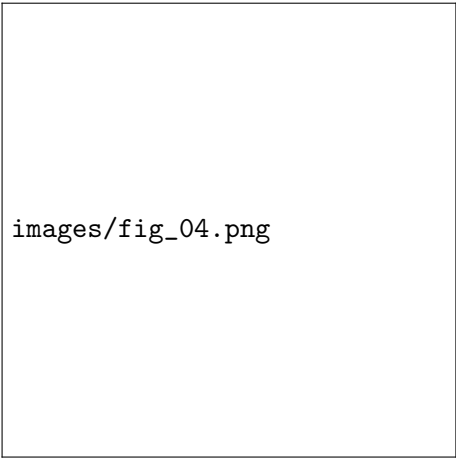
 images/fig_07.png

Hình 4: Các mẫu đối nghịch được sinh bởi thuật toán EAD trên tập ImageNet

Thực nghiệm: Phá vỡ chất lọc phòng thủ

images/fig_3.png


Thực nghiệm: Chuyển giao tấn công



images/fig_04.png

Hình 6: Khả năng chuyển giao tấn công (trường hợp trung bình) từ mạng không phòng thủ sang mạng chất lọc phòng thủ trên tập dữ liệu MNIST với các tham số κ khác nhau. EAD có thể đạt ASR gần 99% khi $\kappa = 50$, trong

Thực nghiệm: Huấn luyện đối nghịch bổ sung



images/tab_4_3.png

Nhận xét: Thời gian tấn công

Phương pháp	EAD-EN	FGM-L1	FGM-L2	IFGM-L1	IFGM-L2
Thời gian (s)	42810	893	1034	3366	9922

Bảng 1: Thời gian tấn công của các thuật toán trên tập CIFAR10

Thời gian tấn công của thuật toán EAD với luật EN là khoảng 11 giờ và gấp khoảng 48 lần khi so sánh với thuật toán nhanh nhất là FGM-L1!

Mở rộng: Thuật toán tổng quát

- Thuật toán FISTA: cực tiểu hóa hàm $f(\mathbf{x}, t)$ trong phương trình ???. Để tính được gradient ∇g ta cần có ràng buộc hàm mất mát của mô hình gốc f phải trơn
- Với một hàm f (lồi) bất kì mà không cần ràng buộc về tính trơn của nó. (Shao, Weijia, Fikret Sivrikaya, & Sahin Albayrak 2022) đã giới thiệu một thuật toán hiệu quả để cực tiểu hóa hàm mục tiêu tổ hợp bằng cập nhật mũ

Mở rộng: Tấn công hệ thống điểm danh GHTK

- EAD tấn công hệ thống nhận diện bằng khuôn mặt
- Tấn công leo thang đặc quyền khi sử dụng khuôn mặt của một nhân viên với quyền thấp hơn để đánh lừa mô hình nhận diện thành một nhân viên với quyền cao hơn
- Framework foolbox¹ đã tích hợp sẵn tấn công EAD và C&W để sinh ra các mẫu đối nghịch

¹<https://github.com/bethgelab/foolbox>

Outline

Kết luận

Nhóm tác giả đã đề xuất mô hình tấn công bằng hiệu chỉnh elastic-net để tạo ra các mẫu đối nghịch trong tấn công DNN. Các kết quả thực nghiệm trên các tập dữ liệu MNIST, CIFAR10 và ImageNet cho thấy các mẫu L_1 tạo bởi EAD có thể đạt được tỷ lệ thành công tương đương với các phương pháp tấn công tiên tiến dựa trên L_2 và L_∞ khi phá vỡ mạng không phòng thủ và phòng thủ cứng cáp. Ngoài ra, EAD có thể cải thiện khả năng chuyển giao tấn công và huấn luyện đối nghịch bổ sung. Các kết quả của nhóm tác giả đã chứng minh hiệu quả của EAD và đưa ra hướng mới sử dụng mẫu đối nghịch L_1 trong việc huấn luyện đối nghịch và tăng cường an ninh cho DNN.