# Tấn công Elastic-Net vào DNN qua một số mẫu đối nghịch

## Khai phá dữ liệu và học máy

Nguyễn Mạnh Linh, Đào Thị Thu Hồng

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

2022/12

# Outline

# Giới thiệu

Sự dễ tổn thương của DNNs


Hình 1: Các mẫu đối nghịch bị phân loại sai bởi mô hình Inception-V3

# Giới thiệu - Các loại tấn công

1. Tấn công nhắm đích - *targeted attacks*
2. Tấn công không nhắm đích - *untargeted attacks*
3. Tấn công chuyển giao - *transfer attacks*

# Giới thiệu: Huấn luyện đối nghịch

Huấn luyện đối nghịch (*adverarial 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

1. Tấn công phân loại ảnh dựa trên mạng neuron tích chập
2. Mẫu đối nghịch được tạo ra để làm sai lệch kết quả phân loại
3. Mẫu mới được tạo ra (gần) giống với mẫu gốc

# Giới thiệu: Độ méo

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

- $L_\infty$: 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) lồi 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
- CIFRA10
- ImageNet

Được so sánh với các tấn công dựa trên $L_2$ và $L_\infty$

# Outline

# Nghiên cứu liên quan

**This is a block**

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

**This is an alert block**

Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis.

# 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 $L_\infty$

$$\mathbf{x} = \mathbf{x_0} - \epsilon \times \text{sign}(\nabla J(\mathbf{x_0}, t)) \tag{1}$$

với $\epsilon$ là độ biến dạng $L_\infty$ giữa $\mathbf{x}$ và $\mathbf{x_0}$ và $\text{sign}(\nabla J)$ là dấu của gradient

- Tấn công dựa trên $L_1$ và $L_2$

$$\mathbf{x} = \mathbf{x_0} - \epsilon \frac{\nabla J(\mathbf{x_0}, t)}{\|\nabla J(\mathbf{x_0}, t)\|_q} \tag{2}$$

với $q = 1, 2$ và $\epsilon$ 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 toát EAD (sẽ được trình bày trong phần sau)

# Phòng thủ DNNs

- Defensive distillation - Chưng 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:

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

  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 elactic-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:

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

Trong đó $\mathbf{Logit}(\mathbf{x}) = [\mathbf{Logit}(\mathbf{x})_1, ..., \mathbf{Logit}(\mathbf{x})_K] \in \mathbb{R}^K$ 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 xuất dự đoán $x$ có nhãn $t$ theo luật phân loại của hàm softmax:

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

# EAD - Xây dựng

Do đó, hàm mất mát trong phương trình 4 có mục đích là để cho ra nhãn $t$ là lớp có xác xuất cao nhất của $x$ và tham số $\kappa$ 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 4 trở thành:

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

# 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:

$$\begin{aligned} &\text{minimize}_{\mathbf{x}}\ c \times f(\mathbf{x}, t) + \beta \|\mathbf{x} - x_0\|_1 + \|\mathbf{x} - \mathbf{x_0}\|_2^2 \\ &\text{st } \mathbf{x} \in [0,1]^p \end{aligned} \tag{7}$$

Với $f(x, t)$ được xác định trong phương trình (4), $c, \beta \geq 0$ lần lượt là các tham số hiệu chỉnh của hàm mất mát $f$ và hàm phạt $L_1$.

# Outline

# Outline

# Conclusion

**This is an example block**

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl.

# References