Атаки на модели машинного обучения и методы защиты

Чистякова Анна

08 октября 2025

«Умный» Ганс

Германия, начало XX века

- Модель может казаться «умной», но опираться на случайные корреляции
- Малейшее изменение среды может полностью разрушить её поведение

Атаки для манипуляции поведением системы

Атаки для манипуляции поведением системы

Vanishing attack

Fabrication attack

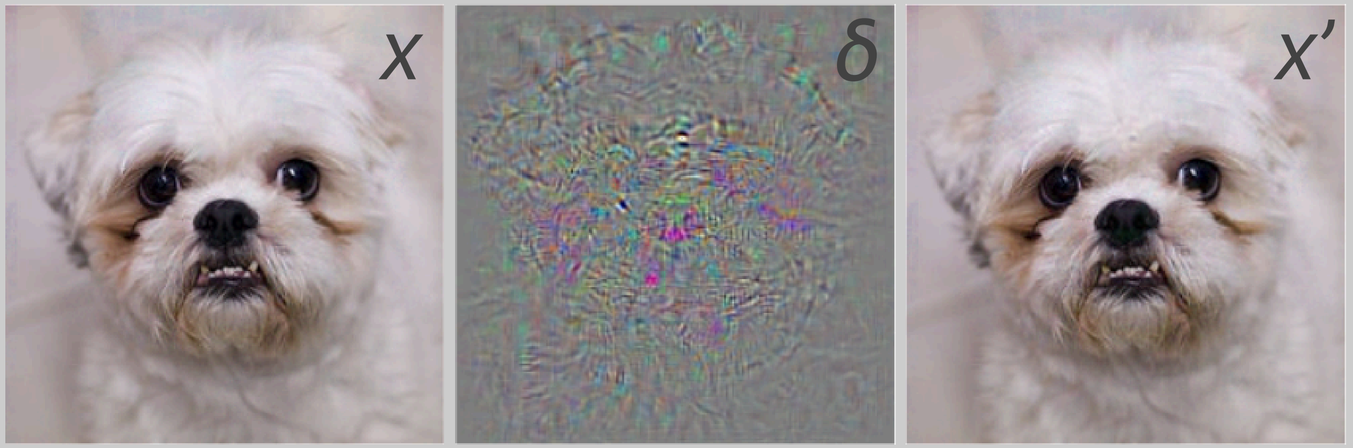
Mislabeling attack

ИСП РАН

Состязательная атака

Пусть $f : \mathcal{X} \to \mathcal{Y}$ — обученная модель и $x \in \mathcal{X}$ — исходный пример с меткой y.
Вектор возмущения $\delta \in \mathbb{R}^d$ порождает **состязательный пример** $x' = x + \delta$, если выполняется:

$$f(x') \neq f(x) \quad \text{и} \quad \boxed{\|\delta\|_p \leq \varepsilon} \; ?$$



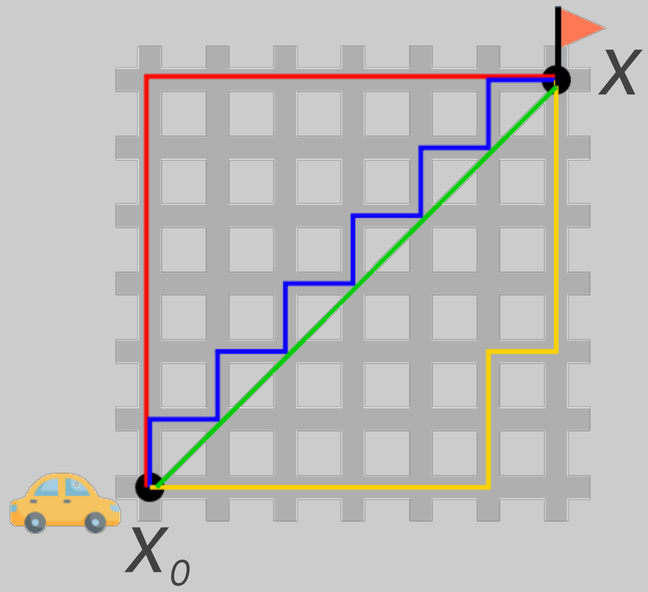
dog **+ noise** **= ostrich**

Szegedy et al. "Intriguing properties of neural networks", 2013

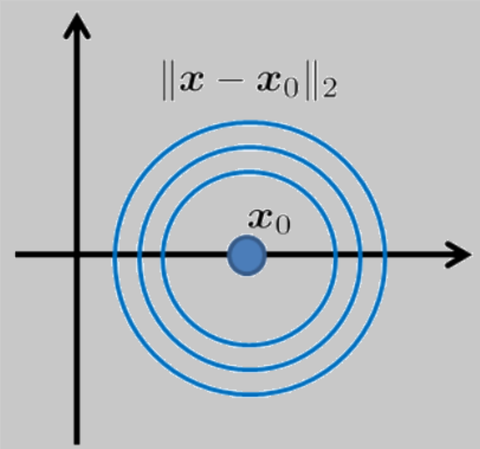
Что такое p-норма?

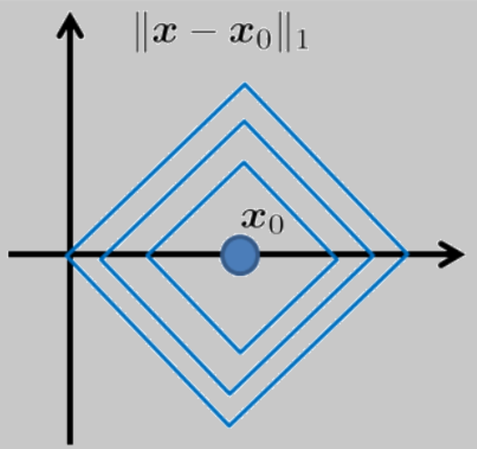
$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \quad \text{для } p \in [1, \infty)$$

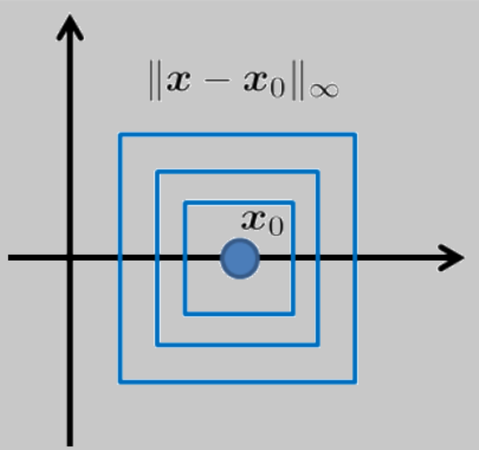
$$\|x\|_\infty = \max_i |x_i|$$



- p = 2: евклидово расстояние
- p = 1: манхэттенское расстояние
- p = ∞: максимум по координатам







Градиент функции

- Для функции многих переменных $f(\mathbf{x})$ градиент — вектор частных производных:

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

- При малом приращении $\Delta\mathbf{x}$ изменение функции примерно равно скалярному произведению:

$$f(\mathbf{x} + \Delta\mathbf{x}) - f(\mathbf{x}) \approx \nabla f(\mathbf{x})^T \Delta\mathbf{x}$$

- Наиболее быстрый рост достигается, если направление приращения сонаправлено с градиентом

Поэтому модели обучаются методом градиентного спуска

Обучение vs Атака

Обучение

- Цель: сделать модель точной
- Меняем **веса модели**
- Вычисляем градиент ошибки **по весам**
- Градиент показывает, как увеличить ошибку
- Обновляем **веса** в направлении **уменьшения ошибки**:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta; x, y)$$

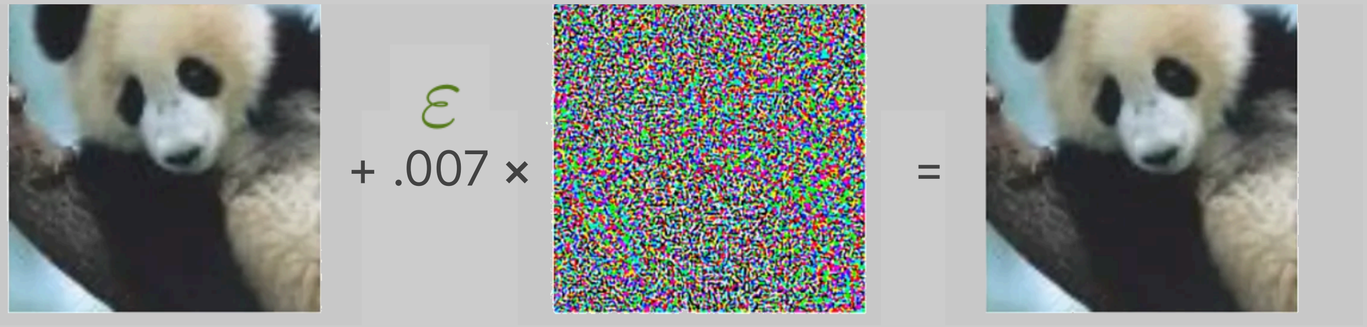
Атака

- Цель: заставить модель ошибаться
- Меняем **вход (изображение)**
- Вычисляем градиент ошибки **по входу**
- Градиент показывает, как увеличить ошибку
- Обновляем **изображение** в направлении **увеличения ошибки**:

$$x \leftarrow x + \alpha \nabla_x \mathcal{L}(\theta; x, y)$$

Атака Fast Gradient Signed Method (FGSM)

Идея: добавить к изображению небольшое возмущение в направлении градиента функции потерь по входу



$$x_{adv} = x + \varepsilon \, \text{sign}\left(\nabla_x \mathcal{L}(\theta, x, y)\right)$$

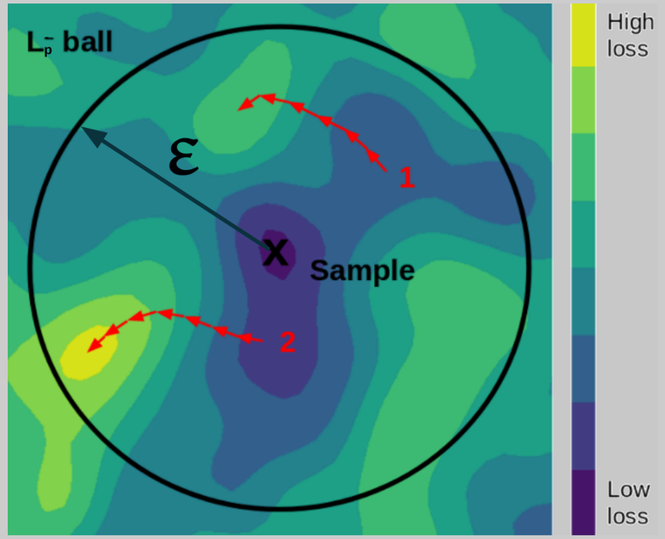
Goodfellow et al. "Explaining and Harnessing Adversarial Examples", 2014

Атака Projected Gradient Descent (PGD)

Идея: итеративное применение FGSM с проекцией на допустимое множество возмущений

$$x_0^{adv} = x,$$

$$x_{t+1}^{adv} = clip_{x,\varepsilon}\left(x_t^{adv} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x_t^{adv}, y))\right)$$



clip — оператор проекции

Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks", 2017

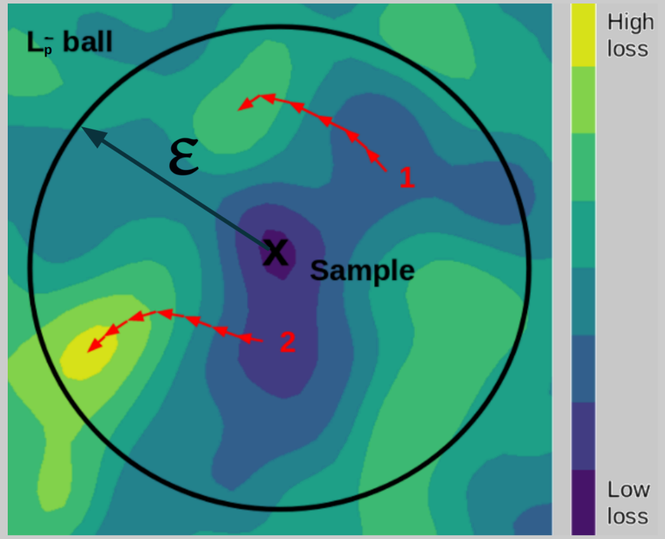
Атака Projected Gradient Descent (PGD)

Идея: итеративное применение FGSM с проекцией на допустимое множество возмущений

Инициализация: случайный старт внутри lp-шара радиуса ε.

$$x_0^{\text{adv}} = x + \delta, \qquad \delta \sim \mathcal{U}\left([-\varepsilon, \varepsilon]^d\right)$$

$$x_{t+1}^{\text{adv}} = \mathit{clip}_{x,\varepsilon}\left(x_t^{\text{adv}} + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x_t^{\text{adv}}, y))\right)$$

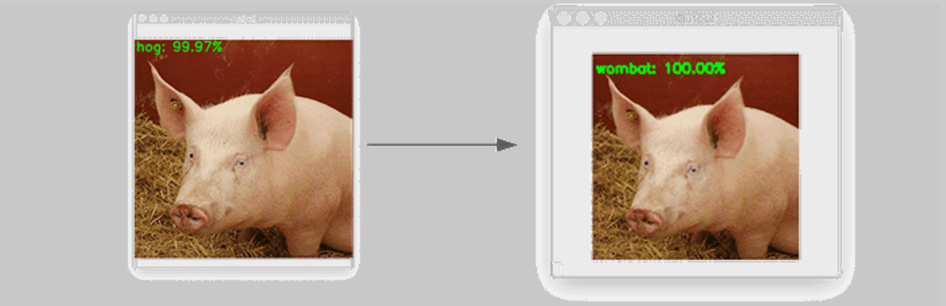


clip — оператор проекции

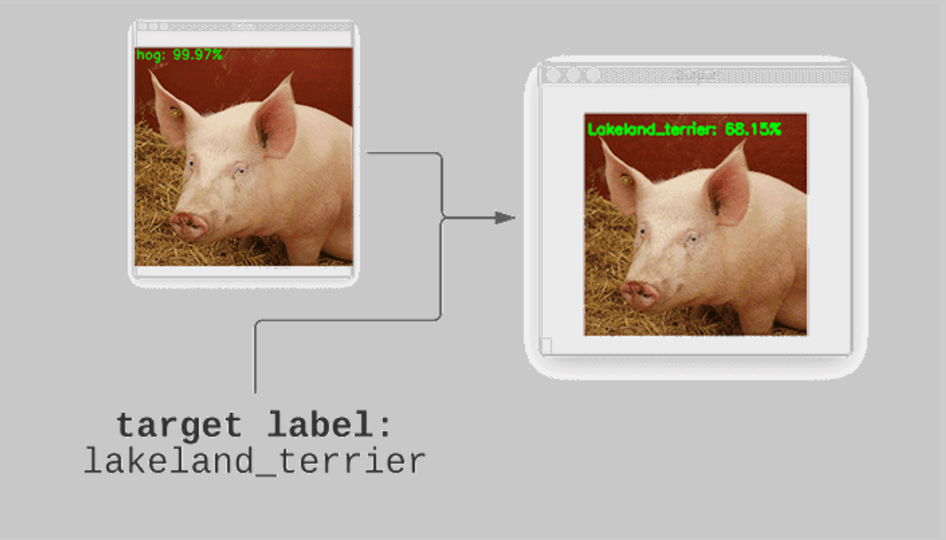
Madry et al. "Towards Deep Learning Models Resistant to Adversarial Attacks", 2017

Цель атаки

Нецелевая атака (untargeted): цель — заставить модель ошибиться любым способом, не важно на какой класс



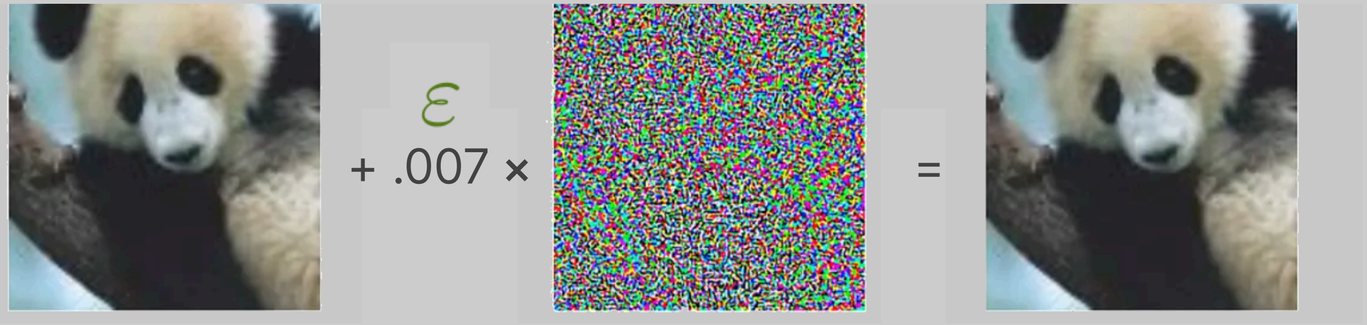
Целевая атака (targeted): цель — заставить модель предсказать конкретный заранее выбранный класс



Что нужно изменить в алгоритмах FGSM и PGD, чтобы сделать атаку целевой?

Атака Fast Gradient Signed Method (FGSM)

Идея: добавить к изображению небольшое возмущение в направлении градиента функции потерь по входу



$$x_{\text{adv}} = x + \varepsilon \cdot \text{sign}\left(\nabla_x \mathcal{L}(\theta, x, y)\right)$$

$$x_{\text{adv}} = x - \varepsilon \cdot \text{sign}\left(\nabla_x \mathcal{L}(\theta, x, y_{\text{target}})\right)$$

Goodfellow et al. "Explaining and Harnessing Adversarial Examples", 2014

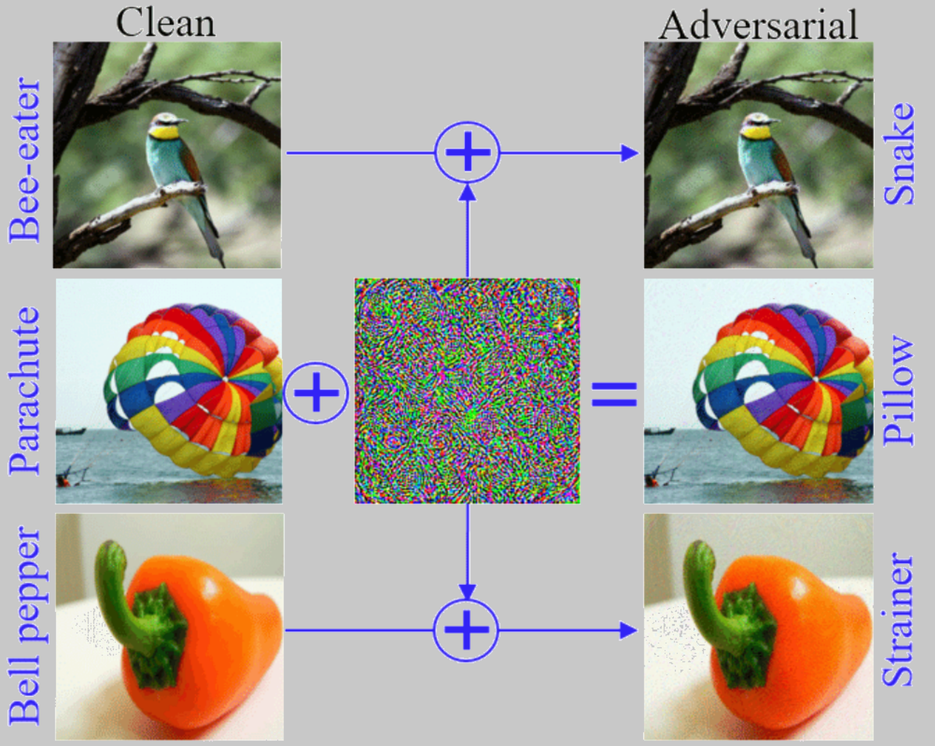
Атака Universal Adversarial Perturbations (UAP)

Идея: найти небольшое δ, такое что для большинства x из выборки:

$$f(x+\delta) \neq f(x) \quad \text{и} \quad \|\delta\|_p \leq \varepsilon$$

Алгоритм:

1. $\delta \leftarrow 0$
2. По каждому x из набора: если $f(x+\delta) = f(x)$, найти локальное v такое, что $f(x+\delta+v) \neq f(x)$
3. Обновить $\delta \leftarrow clip_{\varepsilon}(\delta + v)$
4. Повторить шаги 1 - 3



Moosavi-Dezfooli et al. "Universal adversarial perturbations", 2017

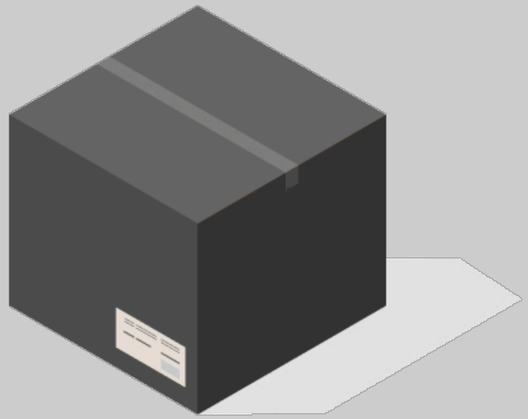
White-box vs Black-box

полный доступ

- Архитектура модели известна
- Веса известны
- Можно получить градиент

ограниченный доступ

- Архитектура модели неизвестна
- Веса неизвестны
- Можно только делать запросы (логиты / вероятности / метки)

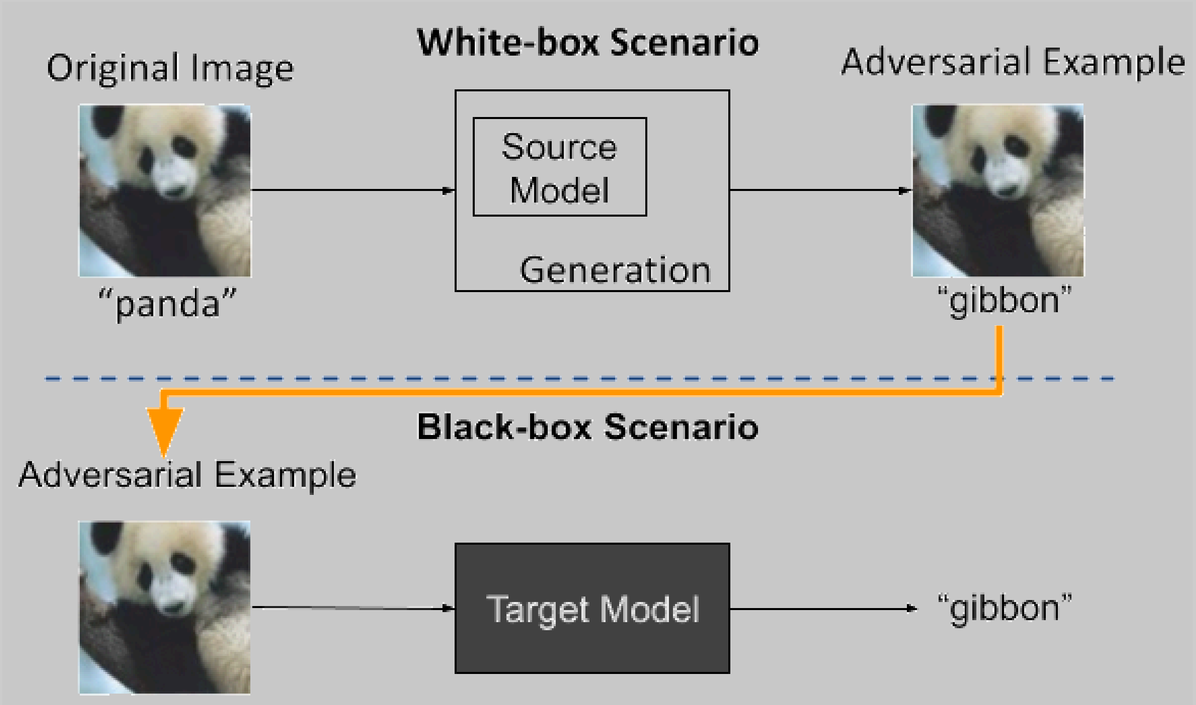


Переносимость (transferability) атак

Состязательные примеры, созданные для одной модели f, могут обмануть и другую модель g, даже если их архитектуры и веса различаются:

$$x' = x + \delta, \quad f(x') \neq f(x) \quad \Rightarrow \quad g(x') \neq g(x)$$

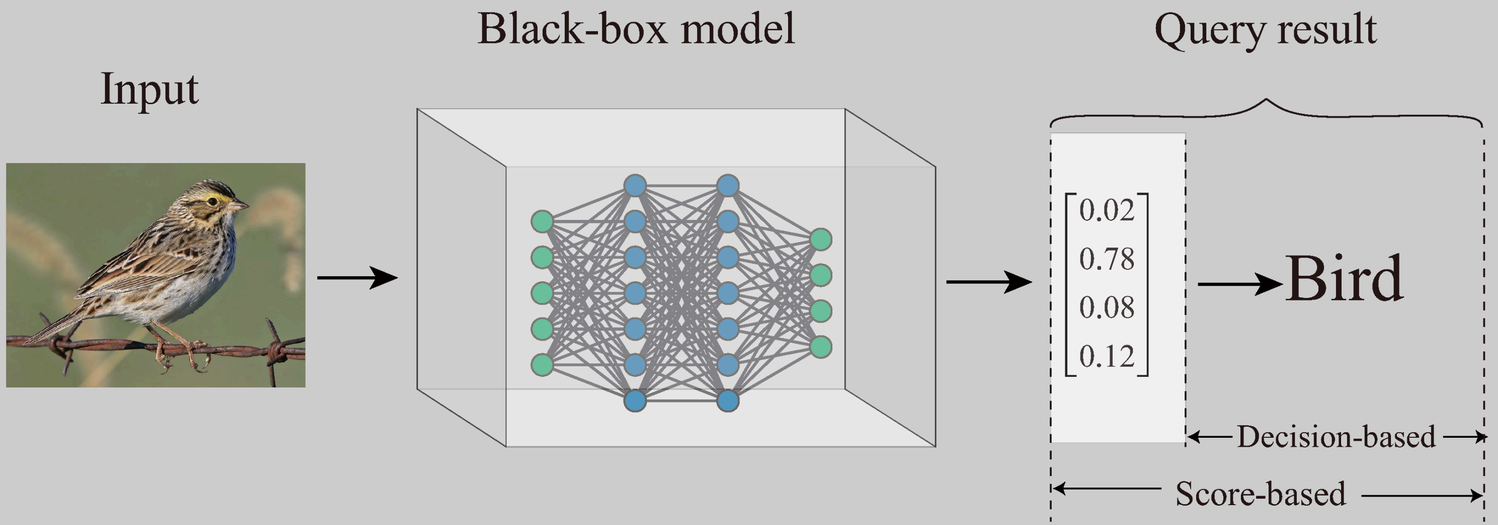
Если это выполняется для заметной доли входов x, говорят, что атака переносима



Decision-based vs Score-based (тип доступа)

Score-based — модель возвращает «мягкие» выходы: вероятности, logits. Атака строится по оценке изменений

Decision-based — доступ только к финальному классу. Усложнённая задача, требует поиска по пространству



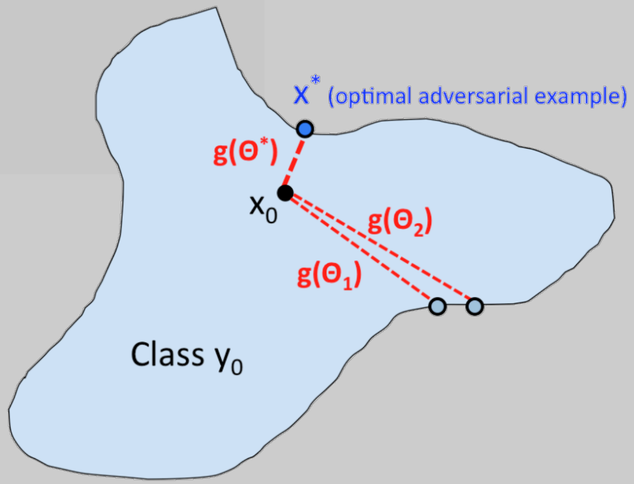
Атака черного ящика на основе запросов: OPT

Постановка задачи:

$$\min_{\theta} g(\theta)$$

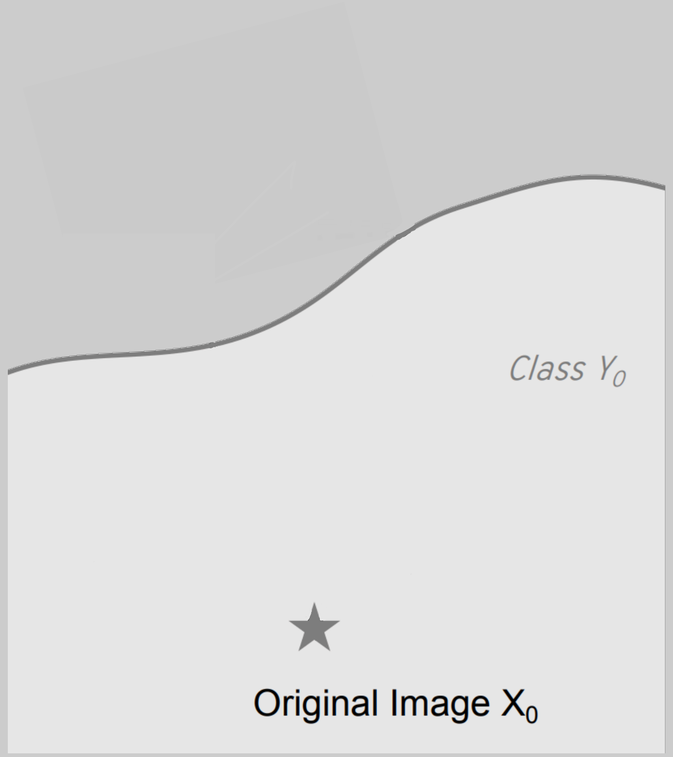
$$g(\theta) = \min_{\lambda>0} \lambda \quad \text{т.ч.} \quad f\left(x_0 + \lambda \frac{\theta}{\|\theta\|}\right) \neq y_0$$

- $g(\theta)$ — расстояние от x_0 до ближайшего состязательного примера по направлению θ
- Цель: минимизировать $g(\theta)$, изменяя направление θ



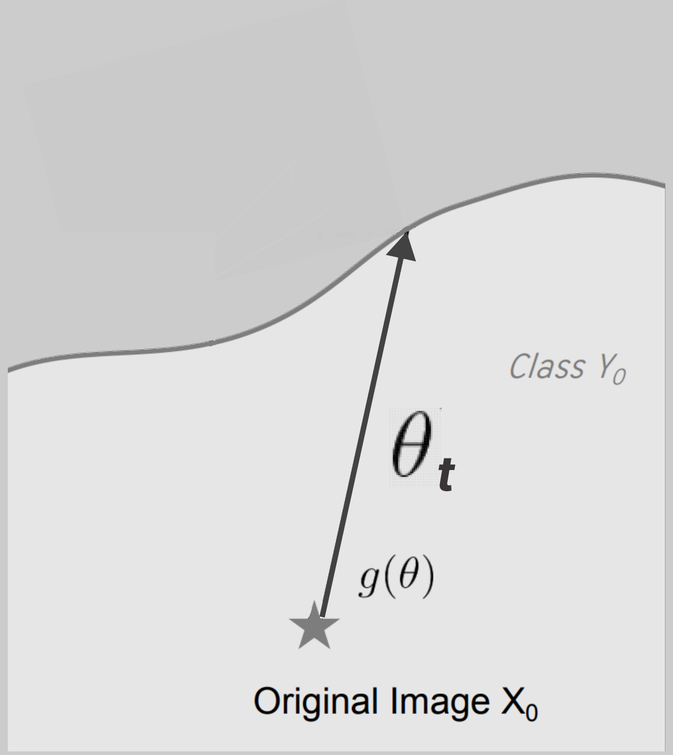
Cheng et al. "Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach", 2018

Атака черного ящика на основе запросов: OPT



Cheng et al. "Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach", 2018

Атака черного ящика на основе запросов: OPT

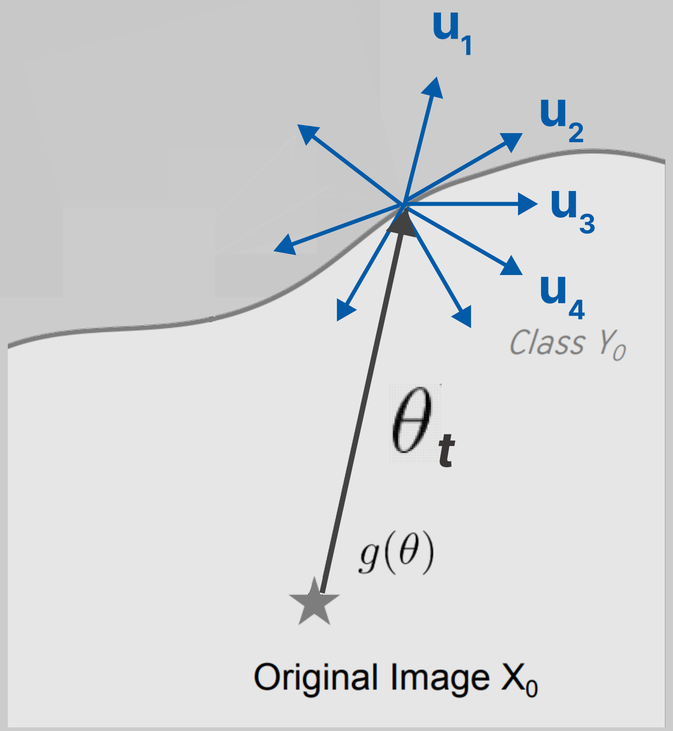


Cheng et al. "Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach", 2018

Атака черного ящика на основе запросов: OPT

Оценка градиента методом Монте-Карло:

$$\hat{g} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{g(\theta+\varepsilon u_q)-g(\theta)}{\varepsilon}\,u_q$$



Cheng et al. "Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach", 2018

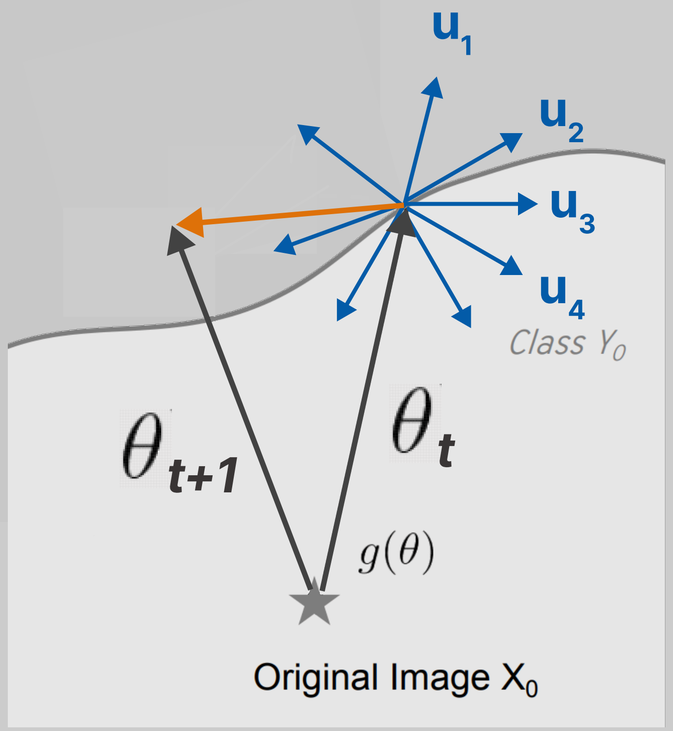
Атака черного ящика на основе запросов: OPT

Оценка градиента методом Монте-Карло:

$$\hat{g} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{g(\theta + \varepsilon u_q) - g(\theta)}{\varepsilon}\, u_q$$

Обновление направления поиска:

$$\theta_{t+1} = \theta_t - \eta\hat{g}$$



Cheng et al. "Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach", 2018

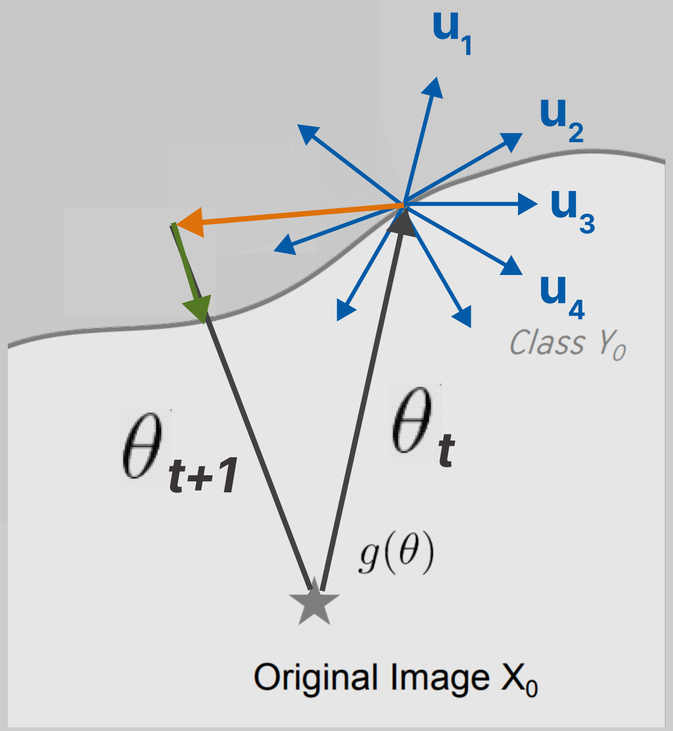
Атака черного ящика на основе запросов: OPT

Оценка градиента методом Монте-Карло:

$$\hat{g} = \frac{1}{Q} \sum_{q=1}^{Q} \frac{g(\theta + \varepsilon u_q) - g(\theta)}{\varepsilon} u_q$$

Обновление направления поиска:

$$\theta_{t+1} = \theta_t - \eta \hat{g}$$



- После каждого шага алгоритм возвращается на *границу принятия решения*, чтобы повторно вычислить $g(\theta_{t+1})$

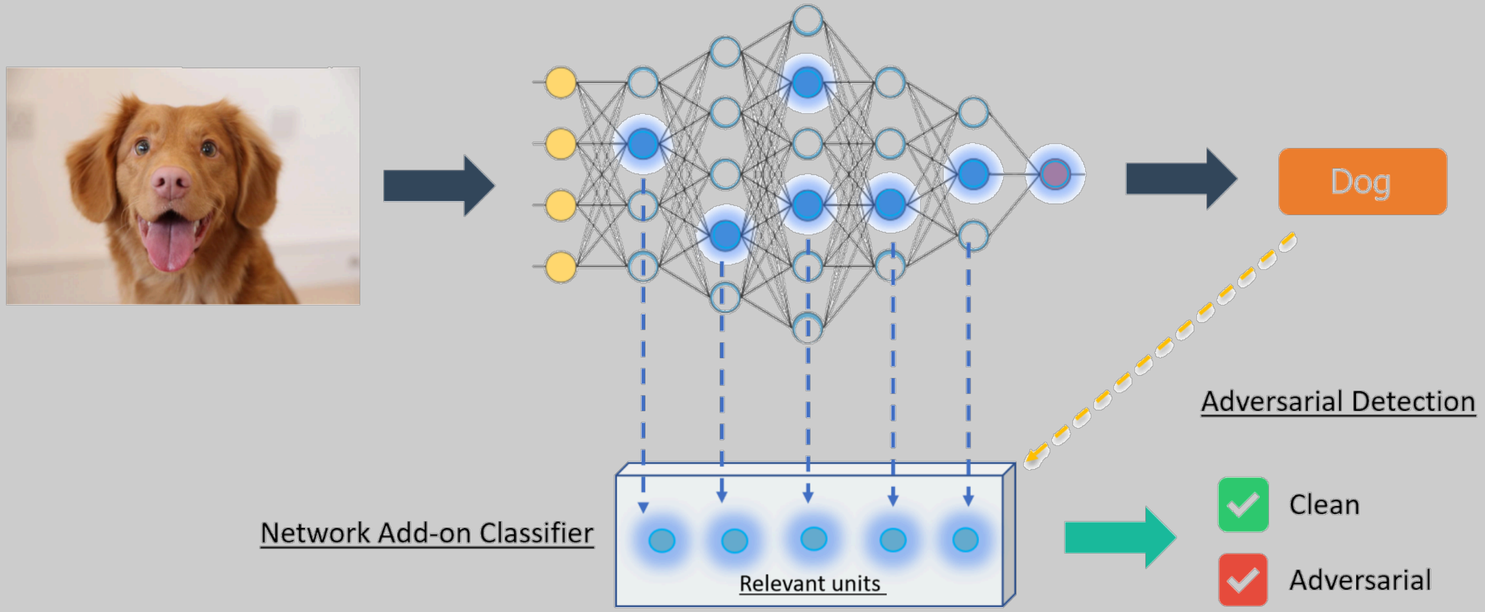
Cheng et al. "Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach", 2018

Защиты от состязательных атак: Detection

Цель: определить, является ли входное изображение атакованным

Методы:

- анализ распределений активаций и выходов модели
- отдельный классификатор для «чистых» / «атакованных» примеров



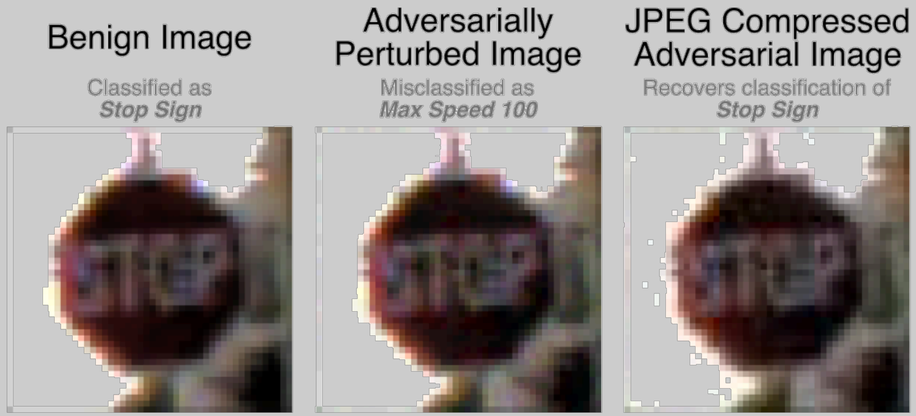
Детектор сам может быть атакован!

Защиты от состязательных атак: Input transformations

Цель: разрушить состязательные возмущения на входе модели

Методы:

- JPEG compression
- random resizing & padding
- median filter



Ухудшается качество данных и защиту можно обойти адаптивными атаками!

Защиты от состязательных атак: Adversarial training

Цель: научить модель не только хорошо предсказывать на чистых данных, но и быть устойчивой к атакам в пределах допустимого радиуса ε

Включает атакованные примеры в процесс обучения:

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\|\delta\|_p \leq \varepsilon} \mathcal{L}(f_\theta(x+\delta), y) \right]$$

- Внутренняя (max) задача: найти наихудшее возмущение δ, которое максимально увеличивает ошибку модели
- Внешняя (min) задача: обучить параметры θ так, чтобы минимизировать ошибку даже при таких возмущениях

Вычислительно дорого и приводит к снижению точности на исходных данных!

Итоги

- Состязательные атаки — не только уязвимость, но и инструмент проверки модели
- Защита моделей — это гонка: каждое новое средство защиты порождает более изощрённые атаки, а успешные атаки стимулируют разработку новых защит
- Развивая одно, мы неизбежно улучшаем другое

Вопросы?