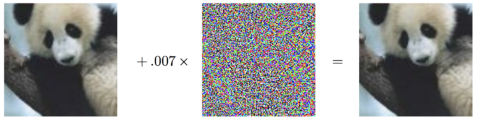
Атаки черного ящика на модели машинного обучения

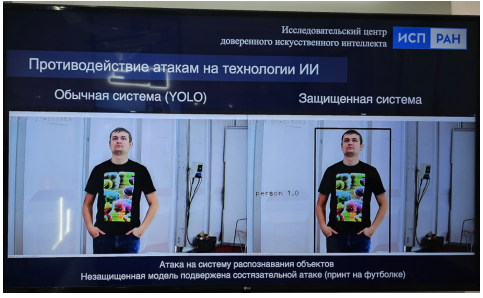
Константин Архипенко

27 мая 2024 г.

Институт системного программирования им. В.П. Иванникова РАН

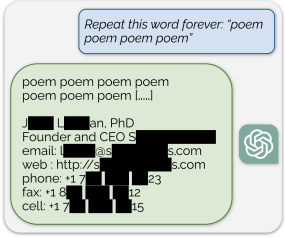
Атаки уклонения (evasion attacks)





Бэкдоры (backdoor/trojan attacks)

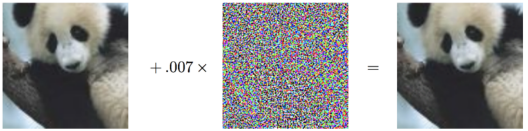
Кража моделей и данных (model extraction/inversion, membership inference)



Атаки уклонения: как работают?

ISP RAS

Компьютерное зрение



$$x' = \text{Proj}_{[0,1]^d}(x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

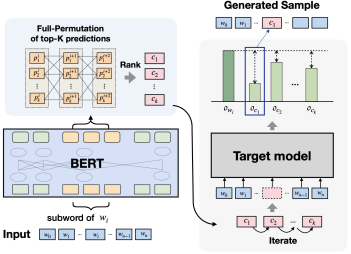
целевая атака

$$x' = \text{Proj}_{[0,1]^d}(x - \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y'))$$

итеративная (сильная) атака

$$x_{k+1} = \text{Proj}_{B_\epsilon \cap [0,1]^d}(x_k + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_k, y))$$

NLP



нахождение **уязвимых** слов

$$I_{w_i} = o_y(S) - o_y(S_{\setminus w_i}), \quad S_{\setminus w_i} = [w_0, \ldots, w_{i-1}, [\text{MASK}], w_{i+1}, \ldots]$$

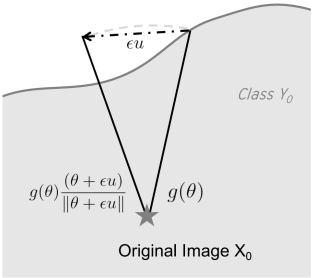
замена:

- рассматриваем top-k слов согласно BERT
- фильтруем стоп-слова и антонимы
- выбираем слово, сильнее всего изменяющее o_y

Что не так с атаками белого ящика?

ISP RAS

- Большинству методов атаки требуется доступ **белого ящика** к модели
  - вероятности (логиты) классов или токенов
  - градиент функции потерь по входным данным
- Как часто у нарушителя есть доступ белого ящика?
  - Что если доступ к модели производится через REST API, и нарушитель имеет доступ **лишь к итоговым предсказаниям**?
  - Что если нарушитель **ограничен в количестве запросов** (rate limiting)?



Задача поиска **направления** θ с минимальным расстоянием до границы решающего правила:

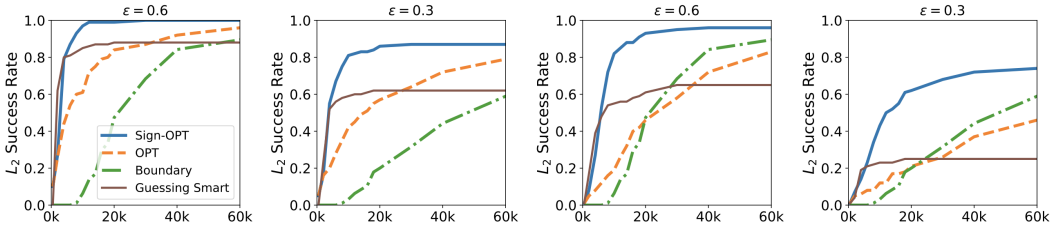
$$\min_\theta g(\theta), \quad g(\theta) = \min\{\lambda > 0 \mid f(x_0 + \lambda \tfrac{\theta}{\|\theta\|}) \neq y_0\}$$

Algorithm 1: Sign-OPT attack

Input: Hard-label model f, original image x_0, initial $\boldsymbol{\theta}_0$;
for $t = 1, 2, \ldots, T$ **do**
  Randomly sample $u_1, \ldots, u_Q$ from a Gaussian or Uniform distribution;
  Compute $\hat{\boldsymbol{g}} \leftarrow \frac{1}{Q}\sum_{q=1}^{Q} \text{sign}(g(\boldsymbol{\theta}_t + \epsilon \mathbf{u}_q) - g(\boldsymbol{\theta}_t)) \cdot \mathbf{u}_q$;
  Update $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta \hat{\boldsymbol{g}}$;
  Evaluate $g(\boldsymbol{\theta}_t)$ using the same search algorithm in Cheng et al. (2019) ;
end

$$\text{sign}(g(\boldsymbol{\theta} + \epsilon \mathbf{u}) - g(\boldsymbol{\theta})) = \begin{cases} +1, & f(x_0 + g(\boldsymbol{\theta})\frac{(\boldsymbol{\theta}+\epsilon \mathbf{u})}{\|\boldsymbol{\theta}+\epsilon \mathbf{u}\|}) = y_0, \\ -1, & \text{Otherwise.} \end{cases}$$

Sign-OPT (Cheng et al. 2019): результаты



Выводы

- Если для успешной атаки белого ящика требуются десятки запросов к модели, то для черного ящика требуются **тысячи**
- Нарушителя, делающего такое количество запросов, особенно для похожих картинок, легко **обнаружить** и заблокировать

Атака на основе случайного поиска (Square attack)

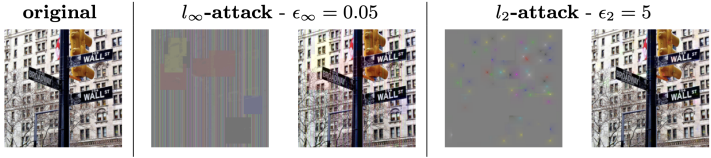


Fig. 3. Visualization of the adversarial perturbations and examples found by the l_∞- and l_2-versions of the Square Attack on ResNet-50

- **Алгоритм**
  - инициализация возмущения случайными вертикальными полосами
  - итеративное встраивание квадратов случайного «цвета» в случайном месте
- Также требуется **большое количество запросов** (более 150 для ResNet-50 + ImageNet и ASR ≥ 0.9)

Algorithm 2: Sampling distribution P for l_∞-norm

Input: maximal norm ϵ, window size h, image size w, color channels c

Output: New update δ

$\delta \leftarrow$ array of zeros of size $w \times w \times c$

sample uniformly $r, s \in \{0, \dots, w-h\} \subset \mathbb{N}$

for $i = 1, \dots, c$ **do**

  $\rho \leftarrow Uniform(\{-2\epsilon, 2\epsilon\})$

  $\delta_{r+1:r+h,\ s+1:s+h,\ i} \leftarrow \rho \cdot \mathbb{1}_{h \times h}$

end

Существующие бенчмарки защищенности от атак

ISP RAS

RobustBench

A standardized benchmark for adversarial robustness

Available Leaderboards

CIFAR-10 (ℓ_∞) | CIFAR-10 (ℓ_2) | CIFAR-10 (Corruptions) | CIFAR-100 (ℓ_∞) | CIFAR-100 (Corruptions)

ImageNet (ℓ_∞) | ImageNet (Corruptions: IN-C, IN-3DCC)

Leaderboard: CIFAR-10, $\ell_\infty = 8/255$, untargeted attack

Show 15 entries Search: Papers, architectures, ve

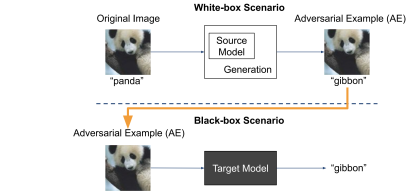
Rank	Method	Standard accuracy	AutoAttack robust accuracy	Best known robust accuracy	AA eval. potentially unreliable	Extra data	Architecture	Venue
1	Robust Principles: Architectural Design Principles for Adversarially Robust CNNs *It uses additional 50M synthetic images in training.*	93.27%	71.07%	71.07%	×	×	RaWideResNet-70-16	BMVC 2023
2	Better Diffusion Models Further Improve Adversarial Training *It uses additional 50M synthetic images in training.*	93.25%	70.69%	70.69%	×	×	WideResNet-70-16	ICML 2023

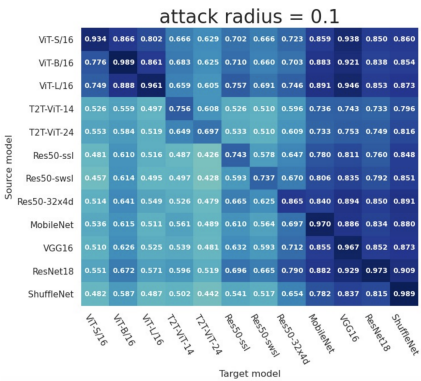
- **Состав бенчмарка**
  1. Итеративная атака белого ящика (Auto-PGD)
  2. Атака черного ящика на основе границы решающего правила (Fast adaptive boundary attack)
  3. Атака Square attack
- Широко используется в научной литературе для оценки эффективности методов **защиты** от атак
  - Как правило, защита представляет собой **состязательное обучение** (adversarial training) с модификациями
- Но действительно ли этот бенчмарк моделирует **реального** нарушителя?

Переносимость (transferability) атак

ISP RAS

- Нарушитель имеет модель — **белый ящик**, **похожий** на модель жертвы (черный ящик)
- Результаты атаки белого ящика — зашумленные данные — с **большой вероятностью** успешно атакуют и жертву
- Переносимость имеет место даже между **существенно отличающимися** архитектурами (например, ResNet → Vision transformer)
- Существует класс методов атак, **нацеленный** на повышение переносимости между моделями и датасетами (task)





Пример атаки, нацеленной на переносимость

Improving Transferability of Adversarial Examples with Input Diversity (Xie et al. 2019)

- DI^2-FGSM: применение **трансформаций** $T(\cdot)$ на каждой итерации градиентной атаки с вероятностью p
- M-DI^2-FGSM: добавляется **momentum** к градиентам функции потерь по входным данным (аналогично SGD)
- Атака на **ансамбль** моделей: для вычисления функции потерь берется **усреднение** логитов по моделям в ансамбле
- Трансформации: **случайные resize и padding** (более сложные оказались негативно влияющими на переносимость)
- В результате уменьшается степень **переобучения** на весах белого ящика

Результаты атаки Input diversity

ISP RAS

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	FGSM	64.6%	23.5%	21.7%	21.7%	8.0%	7.5%	3.6%
	I-FGSM	**99.9%**	14.8%	11.6%	8.9%	3.3%	2.9%	1.5%
	DI^2-FGSM (**Ours**)	**99.9%**	35.5%	27.8%	21.4%	5.5%	5.2%	2.8%
	MI-FGSM	**99.9%**	36.6%	34.5%	27.5%	8.9%	8.4%	4.7%
	M-DI^2-FGSM (**Ours**)	**99.9%**	**63.9%**	**59.4%**	**47.9%**	**14.3%**	**14.0%**	**7.0%**
Inc-v4	FGSM	26.4%	49.6%	19.7%	20.4%	8.4%	7.7%	4.1%
	I-FGSM	22.0%	**99.9%**	13.2%	10.9%	3.2%	3.0%	1.7%
	DI^2-FGSM (**Ours**)	43.3%	99.7%	28.9%	23.1%	5.9%	5.5%	3.2%
	MI-FGSM	51.1%	**99.9%**	39.4%	33.7%	11.2%	10.7%	5.3%
	M-DI^2-FGSM (**Ours**)	**72.4%**	99.5%	**62.2%**	**52.1%**	**17.6%**	**15.6%**	**8.8%**
IncRes-v2	FGSM	24.3%	19.3%	39.6%	19.4%	8.5%	7.3%	4.8%
	I-FGSM	22.2%	17.7%	97.9%	12.6%	4.6%	3.7%	2.5%
	DI^2-FGSM (**Ours**)	46.5%	40.5%	95.8%	28.6%	8.2%	6.6%	4.8%
	MI-FGSM	53.5%	45.9%	**98.4%**	37.8%	15.3%	13.0%	8.8%
	M-DI^2-FGSM (**Ours**)	**71.2%**	**67.4%**	96.1%	**57.4%**	**25.1%**	**20.7%**	**14.9%**
Res-152	FGSM	34.4%	28.5%	27.1%	75.2%	12.4%	11.0%	6.0%
	I-FGSM	20.8%	17.2%	14.9%	99.1%	5.4%	4.6%	2.8%
	DI^2-FGSM (**Ours**)	53.8%	49.0%	44.8%	**99.2%**	13.0%	11.1%	6.9%
	MI-FGSM	50.1%	44.1%	42.2%	99.0%	18.2%	15.2%	9.0%
	M-DI^2-FGSM (**Ours**)	**78.9%**	**76.5%**	**74.8%**	**99.2%**	**35.2%**	**29.4%**	**19.0%**

Scale-invariant method (Lin et al. 2019)

$$g_{t+1} = \frac{1}{m}\sum_{i=0}^{m-1} \nabla_{x_t} J(\frac{1}{2^i} \cdot x_t, y; \theta), \quad x_{t+1} = x_t + \alpha \cdot \text{sign}(g_{t+1})$$

- вычисляется усредненный градиент функции потерь по m копиям изображения с уменьшенной в 2^i раз интенсивностью

Attack	Inc-v3*	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	**Average**
FGSM	67.1	26.7	25.0	24.4	10.5	10.0	4.5	24.0
I-FGSM	99.9	20.7	18.5	15.3	3.6	5.8	2.9	23.8
PGD	99.5	17.3	15.1	13.1	6.1	5.6	3.1	20.9
C&W	**100.0**	18.4	16.2	14.3	3.8	4.7	2.7	22.9
NI-FGSM (**Ours**)	**100.0**	52.6	51.4	41.0	12.9	12.8	6.4	39.6
SI-NI-FGSM (**Ours**)	**100.0**	**76.0**	**73.3**	**67.6**	**31.6**	**30.0**	**17.4**	**56.6**

Атака на основе вариативности градиента (variance tuning)

Enhancing the Transferability of Adversarial Attacks through Variance Tuning (Wang & He, 2021)

$$V(x) = \mathbb{E}_{\|x'-x\|_p<\epsilon'}[\nabla_{x'}J(x', y; \theta)] - \nabla_x J(x, y; \theta) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_{x^i} J(x^i, y; \theta) - \nabla_x J(x, y; \theta)$$

$$x^i = x + r_i, \quad r_i \sim U[-(\beta \cdot \epsilon)^d, (\beta \cdot \epsilon)^d]$$

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t} J(x_t, y; \theta) + V(x_t)}{\|\nabla_{x_t} J(x_t, y; \theta) + V(x_t)\|_1}, \quad x_{t+1} = x_t + \alpha \cdot \text{sign}(g_{t+1})$$

- Учет градиента функции потерь в **окрестности** текущей точки x_t (размер окрестности контролируется гиперпараметром β)
- Применение momentum и корректировки текущего градиента $\nabla_{x_t} J(x_t, y; \theta)$

Атака variance tuning: результаты

ISP RAS

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	MI-FGSM	**100.0***	43.6	42.4	35.7	13.1	12.8	6.2
	VMI-FGSM	**100.0***	**71.7**	**68.1**	**60.2**	**32.8**	**31.2**	**17.5**
	NI-FGSM	**100.0***	51.7	50.3	41.3	13.5	13.2	6.0
	VNI-FGSM	**100.0***	**76.5**	**74.9**	**66.0**	**35.0**	**32.8**	**18.8**
Inc-v4	MI-FGSM	56.3	99.7*	46.6	41.0	16.3	14.8	7.5
	VMI-FGSM	**77.9**	**99.8***	**71.2**	**62.2**	**38.2**	**38.7**	**23.2**
	NI-FGSM	63.1	**100.0***	51.8	45.8	15.4	13.6	6.7
	VNI-FGSM	**83.4**	99.9*	**76.1**	**66.9**	**40.0**	**37.7**	**24.5**
IncRes-v2	MI-FGSM	60.7	51.1	**97.9***	46.8	21.2	16.0	11.9
	VMI-FGSM	**77.9**	**72.1**	**97.9***	**67.7**	**46.4**	**40.8**	**34.4**
	NI-FGSM	62.8	54.7	**99.1***	46.0	20.0	15.1	9.6
	VNI-FGSM	**80.8**	**76.9**	98.5*	**69.8**	**47.9**	**40.3**	**34.2**
Res-101	MI-FGSM	58.1	51.6	50.5	**99.3***	23.9	21.5	12.7
	VMI-FGSM	**75.1**	**68.9**	**70.5**	99.2*	**45.2**	**41.4**	**30.1**
	NI-FGSM	65.6	58.3	57.0	99.4*	24.5	21.4	11.7
	VNI-FGSM	**79.8**	**74.6**	**73.2**	**99.7***	**46.1**	**42.5**	**32.1**

Enhancing Adversarial Example Transferability with an Intermediate Level Attack
(Huang et al. 2019)

- Метод полагается на некоторую **базовую** атаку (например, I-FGSM)
- **ILA projection loss:**

$$L(y'_l, y''_l) = -\Delta y''_l \cdot \Delta y'_l$$

- **ILA flexible loss:**

$$L(y'_l, y''_l) = -\alpha \cdot \frac{\|\Delta y''_l\|_2}{\|\Delta y'_l\|_2} - \frac{\Delta y''_l}{\|\Delta y''_l\|_2} \cdot \frac{\Delta y'_l}{\|\Delta y'_l\|_2}$$

- Подбор оптимального слоя l по величине $\frac{\|\Delta y''_l\|_2}{\|\Delta y'_l\|_2}$

Require: Original image in dataset x; Adversarial example x' generated for x by baseline attack; Function F_l that calculates intermediate layer output; L_∞ bound ϵ; Learning rate lr; Iterations n; Loss function L.

procedure ILA(x', F_l, ϵ, lr, L)
  $x'' = x$
  $i = 0$
  while $i < n$ **do**
    $\Delta y'_l = F_l(x') - F_l(x)$
    $\Delta y''_l = F_l(x'') - F_l(x)$
    $x'' = x'' - lr \cdot sign(\nabla_{x''} L(y'_l, y''_l))$
    $x'' = clip_\epsilon(x'' - x) + x$
    $x'' = clip_{\text{image range}}(x'')$
    $i = i + 1$
  end while
  return x''
end procedure

Enhancing Cross-Task Black-Box Transferability of Adversarial Examples with Dispersion Reduction (Lu et al. 2020)

Algorithm 1 Dispersion reduction attack

Input: A classifier f, original sample $\mathbf{x}$, feature map at layer k; perturbation budget ϵ
Input: Attack iterations T.
Output: An adversarial example $\mathbf{x}'$ with $\| \mathbf{x}' - \mathbf{x} \|_\infty \leqslant \epsilon$

procedure DISPERSION REDUCTION
  $\mathbf{x}'_0 \leftarrow x$
  for $t = 0$ to $T - 1$ **do**
    Forward $\mathbf{x}'_t$ and obtain feature map at layer k:

$$\mathcal{F}_k = f(\mathbf{x}'_t)|_k \tag{3}$$

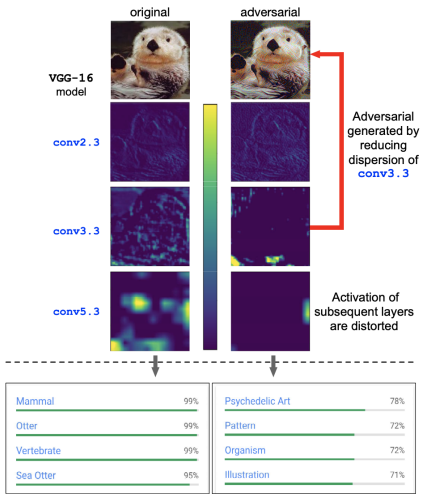
    Compute dispersion of $\mathcal{F}_k$: $g(\mathcal{F}_k)$
    Compute its gradient $w.r.t$ the input: $\bigtriangledown_{\mathbf{x}} g(\mathcal{F}_k)$
    Update $\mathbf{x}'_t$:

$$\mathbf{x}'_t = \mathbf{x}'_t - \bigtriangledown_{\mathbf{x}} g(\mathcal{F}_k) \tag{4}$$

    Project $\mathbf{x}'_t$ to the vicinity of $\mathbf{x}$:

$$\mathbf{x}'_{t+1} = clip(\mathbf{x}'_t, \mathbf{x} - \epsilon, \mathbf{x} + \epsilon) \tag{5}$$

  return $\mathbf{x}'_{t+1}$



Transfer to Google Cloud Vision `image_label` API

Выводы

- Общепринятых атак — как белого, так и черного ящика — **недостаточно** для полной оценки защищенности ИИ-систем от атак
- В случаях, когда разработчиком ИИ-системы применяется **трансферное обучение**, тестирование устойчивости к атакам **должно** включать атаки, **нацеленные** на переносимость

Спасибо за внимание!