AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method for Attacking Black-Box Neural Networks[1]

Yan Ju

Feb 15, 2023

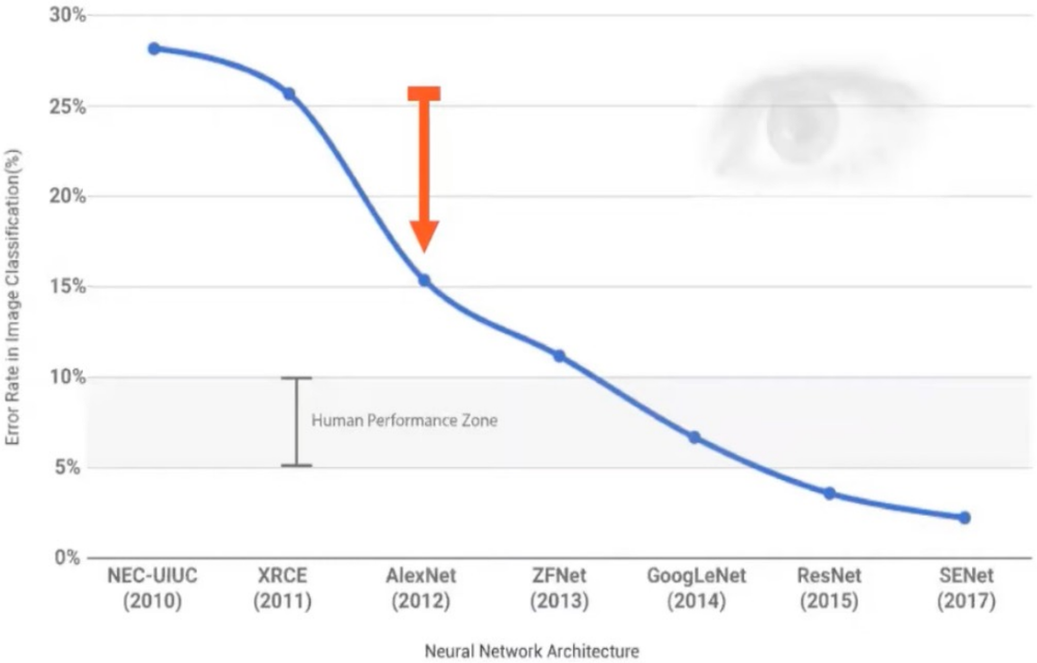
Content

- Background
  - Black-box Adversarial Attack
  - ZO (Zeroth Order) Optimization
- Related Works
- Method
- Experiment Results
- Conclusions

Background

- High Accuracy of DNN models

https://medium.com/syncedreview/sensetime-trains-imagenet-alexnet-in-record-1-5-minutes-e944ab049b2c



https://www.youtube.com/watch?v=17AL1mS3uxw&ab_channel=TrustworthyAI

Background

- However, what’s wrong with this classification model?[3]

original image

successful adversarial samples

https://ojs.aaai.org/index.php/AAAI/article/view/11302

Accuracy ≠ Adversarial Robustness

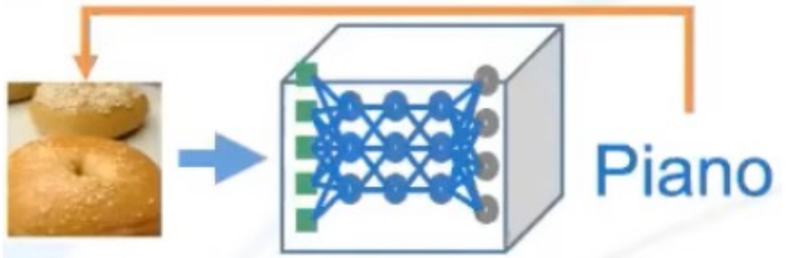
Background

- What is adversarial attack?
  - Generating adversarial examples to deceive machine-learning models

- Why studying adversarial attack?
  - Test and debug ML system: discover vulnerability of ML models before real attackers do so.
  - Rethink current models and training models for the new objective: accuracy +adversarial robustness.

Background

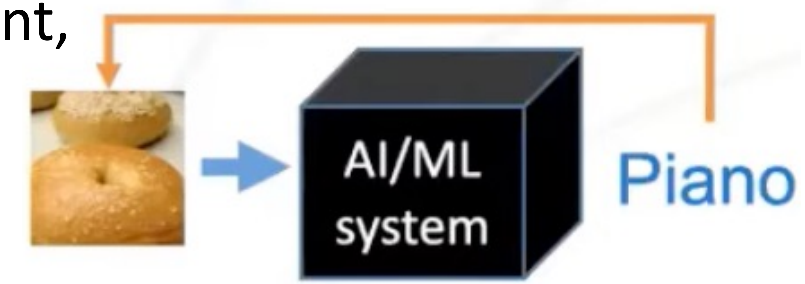
- Different Adversarial Attacks
  - White-box Attack: target model is transparent
    input gradients and BP can be used to attack the model



(a)

https://www.youtube.com/watch?v=17AL1mS3uxw&ab_channel=TrustworthyAI

  - Black-box Attack: target model is not transparent, only observe inputs and outputs (e.g., online APIs)
    input gradients is infeasible and inaccessible



(b)

https://www.youtube.com/watch?v=17AL1mS3uxw&ab_channel=TrustworthyAI

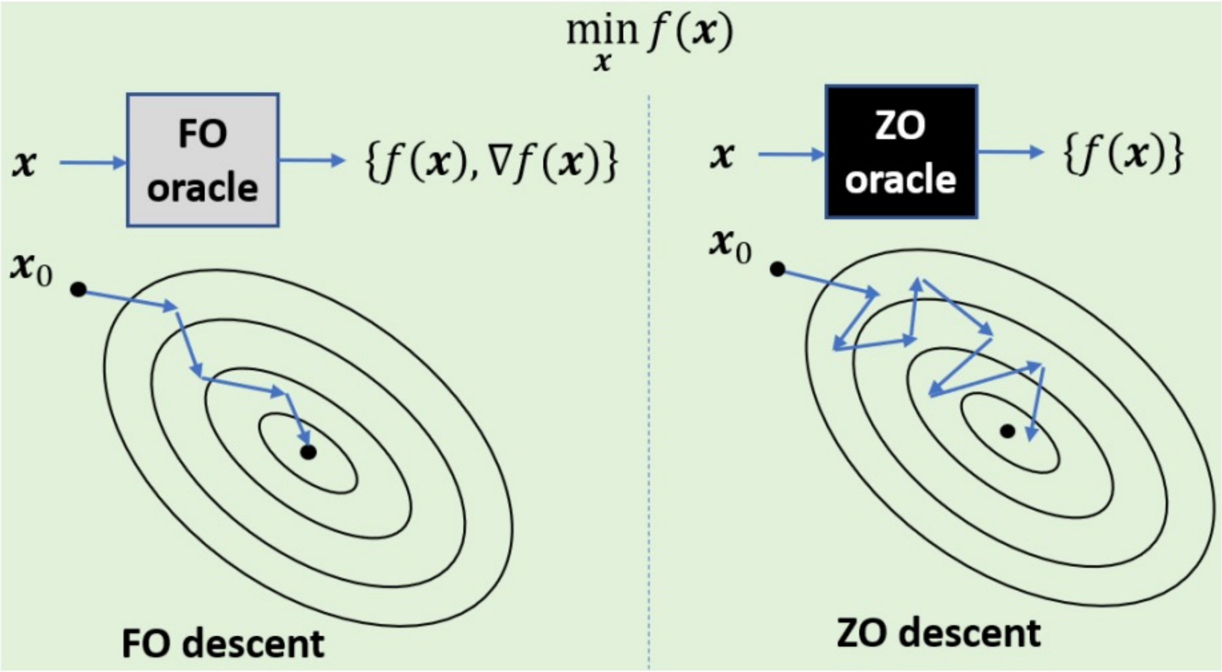
↓

Use ZO Optimization!!!

  - …

Background

- What is ZO Optimization?
  - A value-based optimization mimicking first-order (FO) methods using gradient estimates[2]



https://arxiv.org/pdf/2006.06224.pdf[2]

Background

- When should we use ZO Optimization?
  - Gradient information is infeasible to obtain

    e.g., Finding adversarial examples for black-box models, Machine learning given only model outputs.
  - Gradient information is difficult/expensive to compute

    e.g., gradient computation involves matrix inverse.
  - Black-box optimization involving high dimensions

Related Works

- [4] is an first attempt using ZO Optimization for black-box attack.
- In [4], gradient $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i}$ of the i-th component is calculated by:

$$g_i = \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h} \approx \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i}$$

- Limitation of [4]: need enormous amount of queries and hence not query-efficient.

For example, the ImageNet dataset: d = 299 × 299 × 3 ≈ 270, 000 input dimensions, each dimension needs two query counts, so it will be 540,000 query counts per iteration. Usually, it will take hundreds of iteration to generate a good sample, so this is unacceptable!!

Method

AutoZOOM: Autoencoder-Based Zeroth Order Optimization Method to improve the query efficiency for black-box attacks.

- Block 1: An adaptive random gradient estimation strategy to balance query counts and image distortion.
- Block 2: An auto-encoder/bilinear resizing to decrease dimension of attack space and accelerate attack.

Methods

- Block1: An adaptive random gradient estimation strategy
  - a scaled random full gradient estimator of $\nabla f(x)$

$$\mathbf{g} = b \cdot \frac{f(\mathbf{x} + \beta \mathbf{u}) - f(\mathbf{x})}{\beta} \cdot \mathbf{u} \qquad (1)$$

$\beta > 0$: small smoothing parameter $\mathbf{u}$:unit-length vector drawn randomly from a unit Euclidean sphere

b:tunable parameter balancing bias and variance

  - the final gradient estimate is averaged over q random directions $\{\mathbf{u}_j\}_{j=1}^{q}$.

$$\overline{\mathbf{g}} = \frac{1}{q} \sum_{j=1}^{q} \mathbf{g}_j \qquad (2)$$

where g_j is a gradient estimate defined in (1) with $\mathbf{u}$ = $\mathbf{u}_j$.

Methods

- Block1: An adaptive random gradient estimation strategy

$$\mathbb{E}\|\bar{\mathbf{g}} - \nabla f(\mathbf{x})\|_2^2 \leq 4\left(\frac{b^2}{d^2} + \frac{b^2}{dq} + \frac{(b-d)^2}{d^2}\right)\|\nabla f(\mathbf{x})\|_2^2 + \frac{2q+1}{q} b^2 \beta^2 L^2 \quad (3)$$

when q, d is fixed, optimal $b^* = \frac{dq}{2q+d}$ by minimizing this term

L: Lipschitz constant, fix b, q, small β

if q is small (query-efficient), d is large, $b^* \approx q$, $\left(\frac{b^2}{d^2} + \frac{b^2}{dq} + \frac{(b-d)^2}{d^2}\right) \approx 1$, larger estimation error

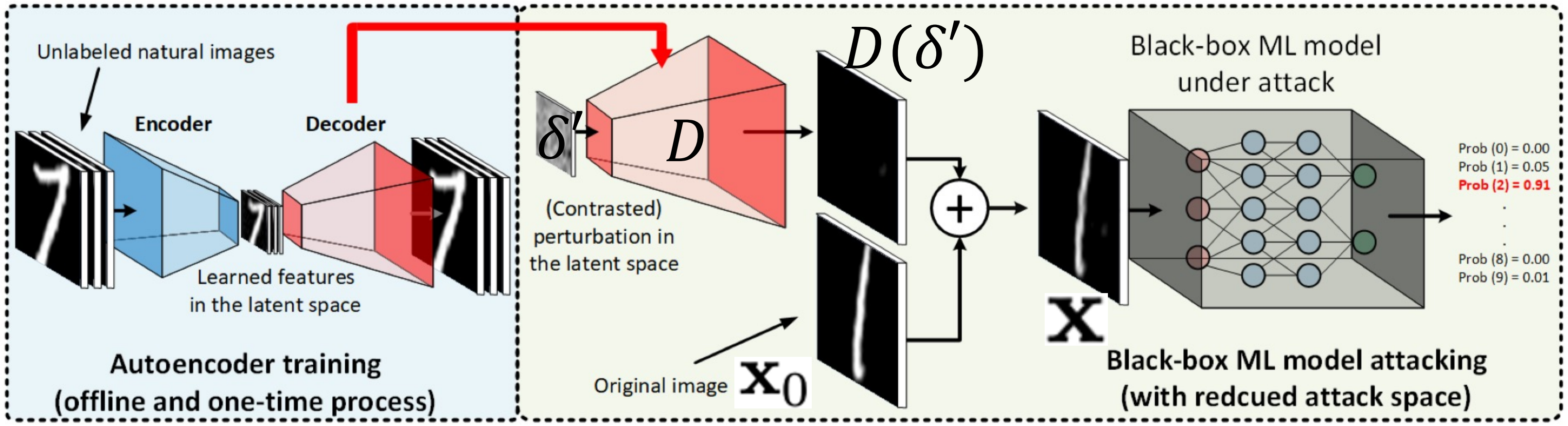
if q is large (query-inefficient), $b^* \approx \frac{d}{2}$, $\left(\frac{b^2}{d^2} + \frac{b^2}{dq} + \frac{(b-d)^2}{d^2}\right) \approx \frac{1}{2}$, smaller estimation error

In AutoZOOM, we set $q = b$, and use an adaptive strategy for selecting q

- $q = 1$ query-efficient but rough estimation
- $q > 1$ query-inefficient but more accurate estimation

Methods

- Block 2: An auto-encoder/bilinear resizing to decrease the dimension of attack space and accelerate attack.
  - motivations:
    - the convergence rate is $O\left(\sqrt{d/T}\right)$, where T is the number of iterations, d is a dimension-dependent factor[5,6,7,8].
    - perform random gradient estimation from a reduced dimension $d' < d$ to improve query efficiency.
  - Method: generate adversarial perturbation δ' from a dimension-reduced space then use Decoder D to map reduced dimension to original dimension



adversarial sample

$$\mathbf{x} = \mathbf{x}_0 + D(\delta')$$

Methods

- Black-box targeted attacks Formulation

$$\min_{\mathbf{x}\in[0,1]^d} \mathrm{Dist}(\mathbf{x},\mathbf{x}_0) + \lambda \cdot \mathrm{Loss}(\mathbf{x}, M(F(\mathbf{x})), t) \quad (4)$$

$F : [0,1]^d \mapsto \mathbb{R}^K$ Classification function
- input: d-dimensional scaled image
- output: a vector of prediction scores of all K image classes

M: monotonic transformation: preserve the ranking of the predictions score and alleviate large score variation

$\mathbf{x}_0$: a natural image, class label is t_0

$\mathbf{x}$: adversarial example, target class label is $t \neq t_0$, $\mathbf{x} \in [0,1]^d$: confine it to the valid image space

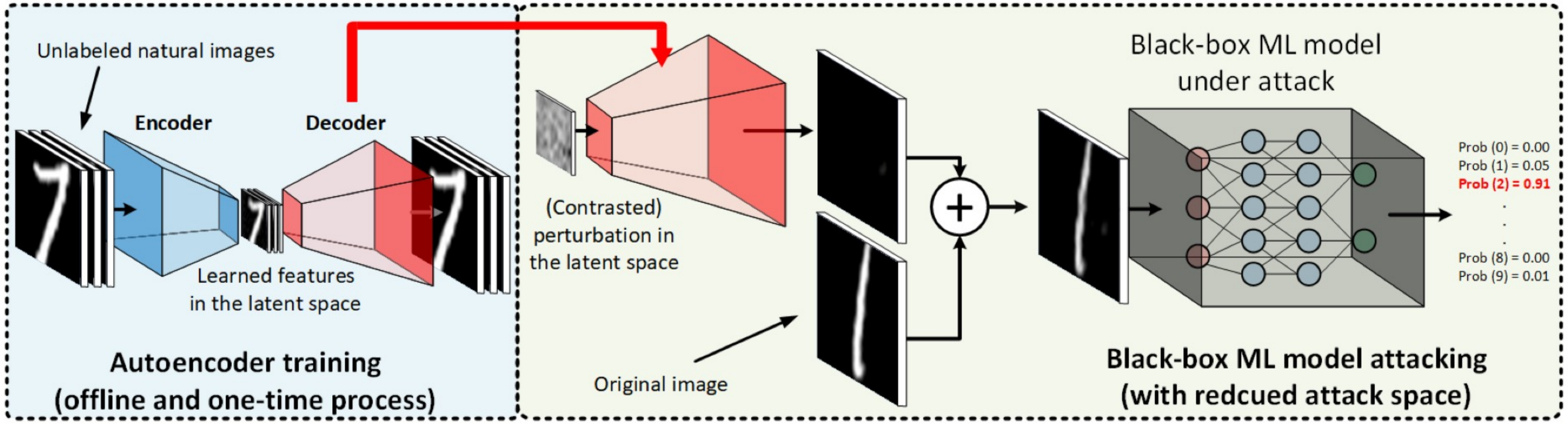
$\mathrm{Dist}(\mathbf{x},\mathbf{x}_0)$:measures the distortion between $\mathbf{x}_0$ and $\mathbf{x}$, using L_P norm $\mathrm{Dist}(\mathbf{x},\mathbf{x}_0) = \|\mathbf{x}-\mathbf{x}_0\|_p = \|\boldsymbol{\delta}\|_p = \sum_{i=1}^{d} |\boldsymbol{\delta}_i|^{1/p}$ for $p \geq 1$

$Loss(\cdot)$ is an attack objective reflecting the likelihood of predicting $t = argmax_{k\epsilon(1,\ldots K)} M(F(X))_k$, Can be the training loss of DNNs or some designed loss based on model predictions.

λ is a regularization coefficient;

Methods

- AutoZOOM Algorithm



Algorithm 1 AutoZOOM for black-box attacks on DNNs

Input: Black-box DNN model F, original example $\mathbf{x}_0$, distortion measure Dist$(\cdot)$, attack objective Loss$(\cdot)$, monotonic transformation $M(\cdot)$, decoder $D(\cdot) \in \{\text{AE}, \text{BiLIN}\}$, initial coefficient λ_{ini}, query budget Q

while query count $\leq Q$ **do**

1. Exploration: use $\mathbf{x} = \mathbf{x}_0 + D(\boldsymbol{\delta}')$ and apply the random gradient estimator in (2) with $q = 1$ to the downstream optimizer (e.g., ADAM) for solving (4) until an initial attack is found.

2. Exploitation (post-success stage): continue to fine-tune the adversarial perturbation $D(\boldsymbol{\delta}')$ for solving (4) while setting $q \geq 1$ in (2).

end while

Output: Least distorted successful adversarial example

$Dist(\cdot)$: the squared L_2 norm

objective for targeted black-box attack:

$$\text{Loss} = \max\{\max_{j \neq t} \log[F(\mathbf{x})]_j - \log[F(\mathbf{x})]_t, 0\}$$

$$M = \log(\cdot)$$

→ query-efficient but rough estimation

→ query-inefficient but more accurate estimation

Experiments

- Datasets
  - MNIST (LeCun et al. 1998), CIFAR-10 (Krizhevsky 2009) and ImageNet (Russakovsky et al. 2015).
  - Reduction rate: MNIST: 28X28X1 -> 14X14X1(25%); CIFAR-10: 32X32X3 -> 8X8X3(6.25%); ImageNet: 299X299X3 -> 32X32X3(1.15%)
- Results

Table 1: Performance evaluation of black-box targeted attacks on MNIST

Method	λ_{ini}	Attack success rate (ASR)	Mean query count (initial success)	Mean query count reduction ratio (initial success)	Mean per-pixel L_2 distortion (initial success)	True positive rate (TPR)	Mean query count with per-pixel L_2 distortion ≤ 0.004
ZOO	0.1	99.44%	35,737.60	0.00%	3.50×10^{-3}	96.76%	47,342.85
	1	99.44%	16,533.30	53.74%	3.74×10^{-3}	97.09%	31,322.44
	10	99.44%	13,324.60	62.72%	4.85×10^{-3}	96.31%	41,302.12
ZOO+AE	0.1	99.67%	34,093.95	4.60%	3.43×10^{-3}	97.66%	44,079.92
	1	99.78%	15,065.52	57.84%	3.72×10^{-3}	98.00%	29,213.95
	10	99.67%	12,102.20	66.14%	4.66×10^{-3}	97.66%	38,795.98
AutoZOOM-BiLIN	0.1	99.89%	2,465.95	93.10%	4.51×10^{-3}	96.55%	3,941.88
	1	99.89%	879.98	97.54%	4.12×10^{-3}	97.89%	2,320.01
	10	99.89%	612.34	98.29%	4.67×10^{-3}	97.11%	4,729.12
AutoZOOM-AE	0.1	**100.00%**	2,428.24	**93.21%**	4.54×10^{-3}	96.67%	3,861.30
	1	**100.00%**	729.65	**97.96%**	4.13×10^{-3}	96.89%	1,971.26
	10	**100.00%**	510.38	**98.57%**	4.67×10^{-3}	97.22%	4,855.01

Experiments

- Overall performance

Table 2: Performance evaluation of black-box targeted attacks on CIFAR-10

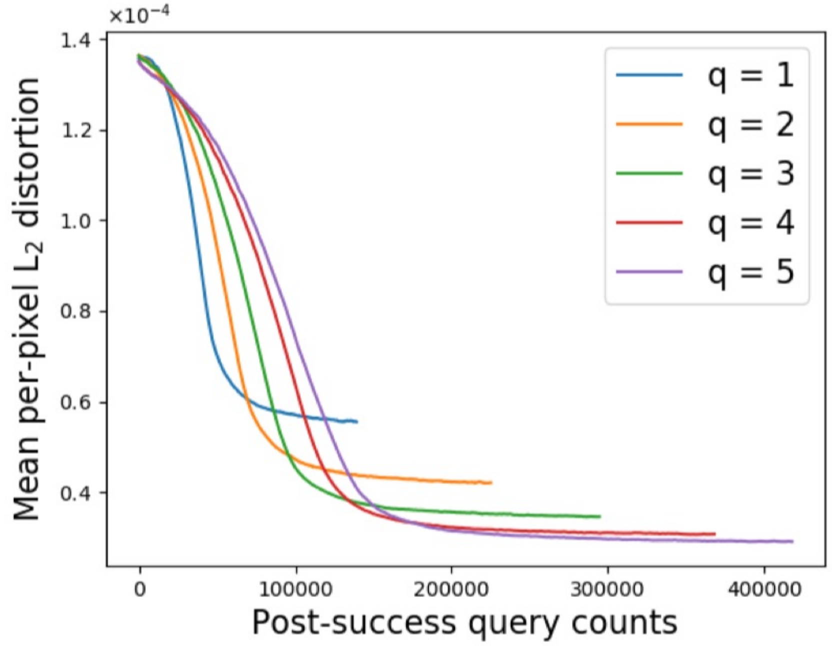
Method	λ_{ini}	Attack success rate (ASR)	Mean query count (initial success)	Mean query count reduction ratio (initial success)	Mean per-pixel L_2 distortion (initial success)	True positive rate (TPR)	Mean query count with per-pixel L_2 distortion ≤ 0.0015
ZOO	0.1	97.00%	25,538.43	0.00%	5.42×10^{-4}	100.00%	25,568.33
	1	97.00%	11,662.80	54.33%	6.37×10^{-4}	100.00%	11,777.18
	10	97.00%	10,015.08	60.78%	8.03×10^{-4}	100.00%	10,784.54
ZOO+AE	0.1	99.33%	19,670.96	22.98%	4.96×10^{-4}	100.00%	20,219.42
	1	99.00%	5,793.25	77.32%	6.83×10^{-4}	99.89%	5,773.24
	10	99.00%	4,892.80	80.84%	8.74×10^{-4}	99.78%	5,378.30
AutoZOOM-BiLIN	0.1	99.67%	2,049.28	91.98%	1.01×10^{-3}	98.77%	2,112.52
	1	99.67%	813.01	96.82%	8.25×10^{-4}	99.22%	1,005.92
	10	99.33%	623.96	97.56%	9.09×10^{-4}	98.99%	835.27
AutoZOOM-AE	0.1	**100.00%**	1,523.91	**94.03%**	1.20×10^{-3}	99.67%	1,752.45
	1	**100.00%**	332.43	**98.70%**	1.01×10^{-3}	99.56%	345.62
	10	**100.00%**	259.34	**98.98%**	1.15×10^{-3}	99.67%	990.61

Table 3: Performance evaluation of black-box targeted attacks on ImageNet

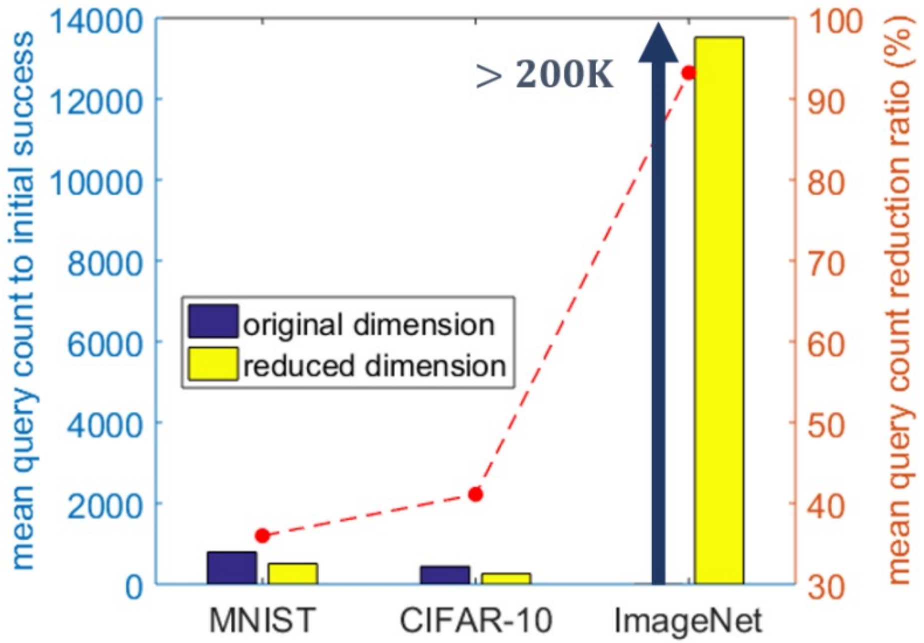
Method	Attack success rate (ASR)	Mean query count (initial success)	Mean query count reduction ratio (initial success)	Mean per-pixel L_2 distortion (initial success)	True positive rate (TPR)	Mean query count with per-pixel L_2 distortion ≤ 0.0002
ZOO	76.00%	2,226,405.04 (2.22M)	0.00%	4.25×10^{-5}	100.00%	2,296,293.73
ZOO+AE	92.00%	1,588,919.65 (1.58M)	28.63%	1.72×10^{-4}	100.00%	1,613,078.27
AutoZOOM-BiLIN	**100.00%**	14,228.88	99.36%	1.26×10^{-4}	100.00%	15,064.00
AutoZOOM-AE	**100.00%**	13,525.00	**99.39%**	1.36×10^{-4}	100.00%	14,914.92

Experiments

- Others



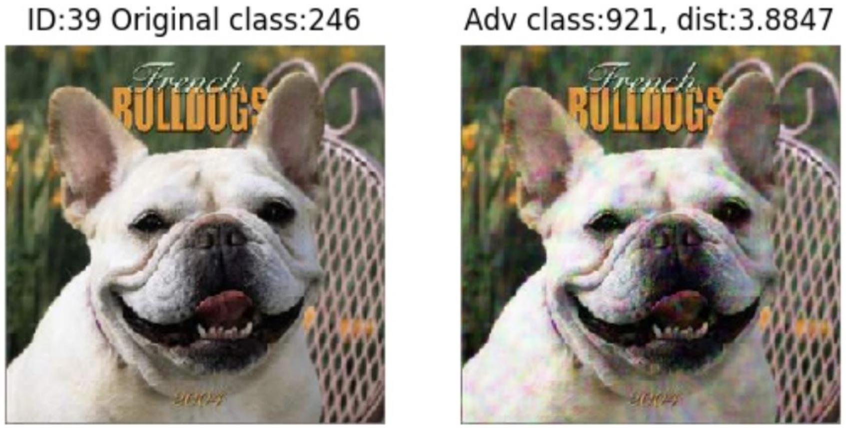
(a)



(b)

Experiments

• Visual performance



(a) “French bulldog” to “traffic light”

(b) “purse” to “bagel”

(c) “bagel” to “ grand piano”

(d) “traffic light” to “ iPod”

Conclusions

- AutoZOOM: a generic attack acceleration framework that uses ZO Optimization for black-box attack.
- It adopts a new and adaptive random full gradient estimation strategy to strike a balance between query counts and estimation errors.
- A decoder (AE or BiLIN) is used for attack dimension reduction and convergence acceleration.

References

[1] Tu, Chun-Chen, et al. "Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. No. 01. 2019.

[2] Liu, Sijia, et al. "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications." *IEEE Signal Processing Magazine* 37.5 (2020): 43-54.

[3] Chen, Pin-Yu, et al. "Ead: elastic-net attacks to deep neural networks via adversarial examples." Proceedings of the AAAI conference on artificial intelligence. Vol. 32. No. 1. 2018.

[4] Chen, Pin-Yu, et al. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models." Proceedings of the 10th ACM workshop on artificial intelligence and security. 2017.

[5] Nesterov, Yurii, and Vladimir Spokoiny. "Random gradient-free minimization of convex functions." Foundations of Computational Mathematics 17 (2017): 527-566.

[6] Liu, Sijia, et al. "Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications." International Conference on Artificial Intelligence and Statistics. PMLR, 2018.

[7] Gao, Xiang, Bo Jiang, and Shuzhong Zhang. "On the information-adaptive variants of the ADMM: an iteration complexity perspective." Journal of Scientific Computing 76 (2018): 327-363.

Thank You!
Q & A