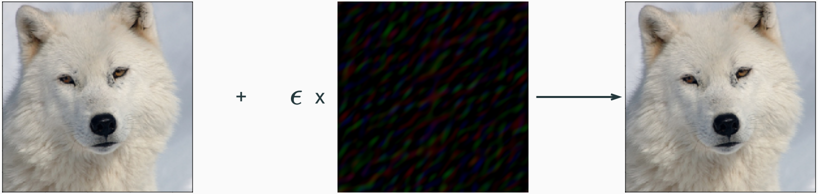
Simple and Efficient Hard Label Black-box Adversarial Attacks in Low Query Budget Regimes

Satya Narayan Shukla[1]

Joint work with Anit Kumar Sahu[2], Devin Willmott[3], Zico Kolter[3,4]

[1]University of Massachusetts Amherst, [2]Amazon Alexa AI, [3]Bosch Center for AI, [4]Carnegie Mellon University

Adversarial Attacks

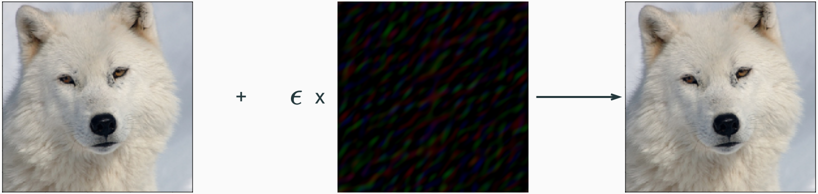


Original Image
Label: wolf

Adversarial Perturbation

Adversarial Image
Label: shower curtain

Adversarial Attacks



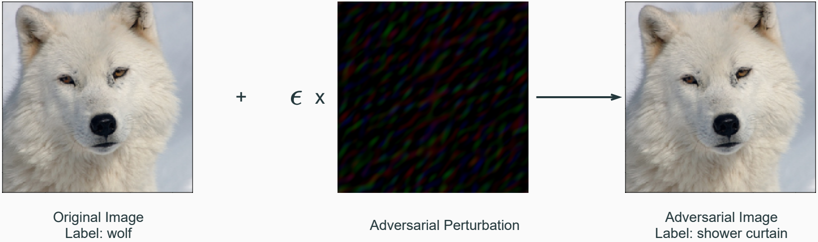
Original Image
Label: wolf

Adversarial Perturbation

Adversarial Image
Label: shower curtain

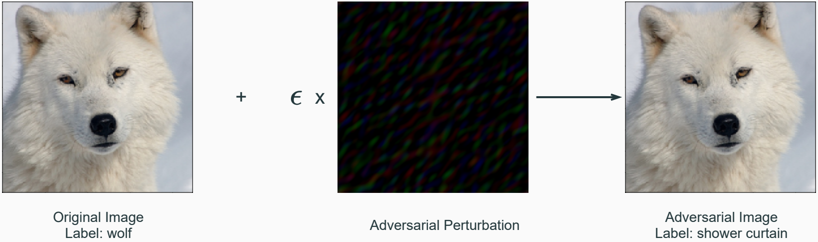
- **Attack Goal:** Untargeted and Targeted

Adversarial Attacks



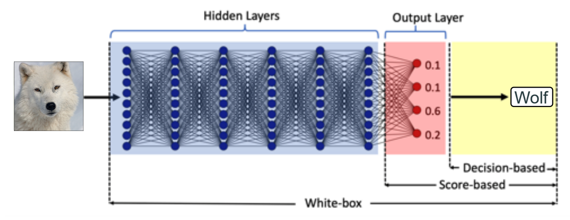
- **Attack Goal:** Untargeted and Targeted
- **Distance Metrics:** ℓ_2, ℓ_∞

Adversarial Attacks



- **Attack Goal:** Untargeted and Targeted
- **Distance Metrics:** ℓ_2, ℓ_∞
- **Threat Model:** White-box and Black-box

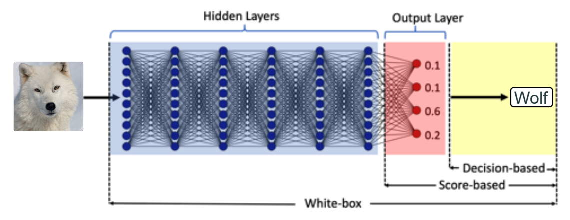
Threat Model



[1]

[1] picture taken from Chen et al. (2019)

Threat Model

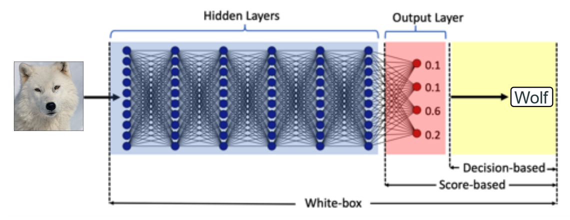


[1]

- White-box Attacks

[1] picture taken from Chen et al. (2019)

Threat Model

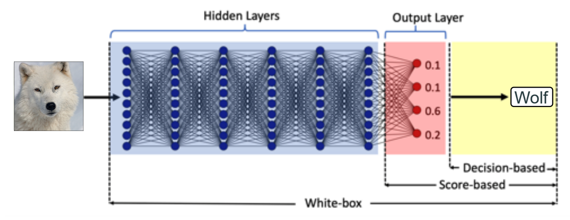


[1]

- White-box Attacks
- Score-based Black-box Attacks (Soft Label)

[1] picture taken from Chen et al. (2019)

Threat Model



[1]

- White-box Attacks
- Score-based Black-box Attacks (Soft Label)
- **Decision-based Black-box Attacks (hard label)**

[1] picture taken from Chen et al. (2019)

Contributions

- We propose *Bayes Attack* , a hard label black-box adversarial attack method in low query budget regimes.

Contributions

- We propose *Bayes Attack* , a hard label black-box adversarial attack method in low query budget regimes.
- Our proposed method uses Bayesian optimization for finding adversarial perturbations in low dimension subspace.

Contributions

- We propose *Bayes Attack* , a hard label black-box adversarial attack method in low query budget regimes.
- Our proposed method uses Bayesian optimization for finding adversarial perturbations in low dimension subspace.
- Our proposed approach achieves higher attack success rate compared to the current state-of-the-art methods while requiring much fewer queries.

Problem Formulation

- Notations
  - Target Model $F : \mathbb{R}^d \to \{1, \ldots K\}$
  - original image $x \in \mathbb{R}^d$
  - Original label $y \in \{1, \ldots, K\}$
  - Perturbation δ
  - Distance threshold ϵ

Problem Formulation

- Notations
  - Target Model $F : \mathbb{R}^d \to \{1, \ldots K\}$
  - original image $x \in \mathbb{R}^d$
  - Original label $y \in \{1, \ldots, K\}$
  - Perturbation δ
  - Distance threshold ϵ
- Objective

$$\max_{\boldsymbol{\delta}} f(\mathbf{x}, y, \boldsymbol{\delta})$$

$$\text{subject to} \;\; \|\boldsymbol{\delta}\|_p \leq \epsilon \;\text{ and }\; (\mathbf{x} + \boldsymbol{\delta}) \in [0, 1]^d,$$

$$\text{where} \;\; f(\mathbf{x}, y, \boldsymbol{\delta}) = \begin{cases} 0 & \text{if } F(\mathbf{x} + \boldsymbol{\delta}) \neq y \\ -1 & \text{if } F(\mathbf{x} + \boldsymbol{\delta}) = y \end{cases}$$

Bayesian Optimization

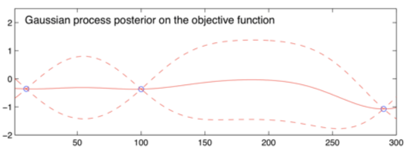
- A black-box optimization method particularly suited to problems with low dimension and expensive queries

Bayesian Optimization

- A black-box optimization method particularly suited to problems with low dimension and expensive queries
- Consists of a surrogate model and an acquisition function

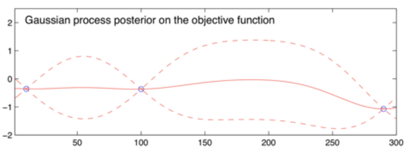
Bayesian Optimization

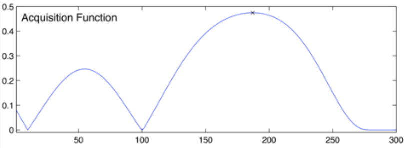
- A black-box optimization method particularly suited to problems with low dimension and expensive queries
- Consists of a surrogate model and an acquisition function



Bayesian Optimization

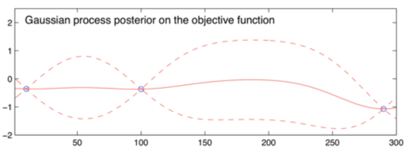
- A black-box optimization method particularly suited to problems with low dimension and expensive queries
- Consists of a surrogate model and an acquisition function

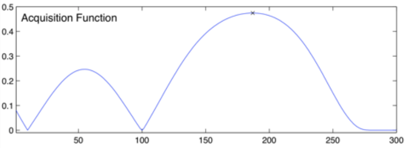




Bayesian Optimization

- A black-box optimization method particularly suited to problems with low dimension and expensive queries
- Consists of a surrogate model and an acquisition function

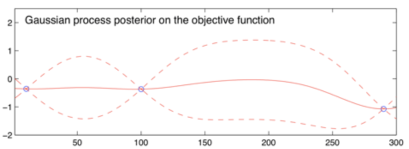


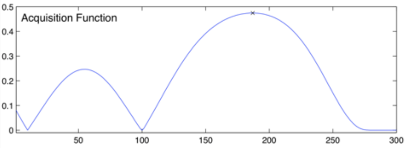


- Gaussian Processes as the surrogate model.

Bayesian Optimization

- A black-box optimization method particularly suited to problems with low dimension and expensive queries
- Consists of a surrogate model and an acquisition function

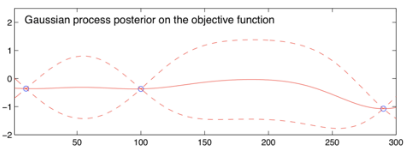


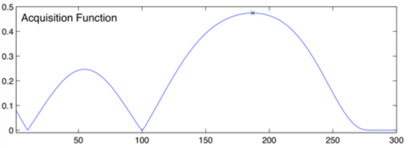


- Gaussian Processes as the surrogate model.
- Expected Improvement as the acquisition function.

Bayesian Optimization

- A black-box optimization method particularly suited to problems with low dimension and expensive queries
- Consists of a surrogate model and an acquisition function





- Gaussian Processes as the surrogate model.
- Expected Improvement as the acquisition function.
- Running BO over high dimensional inputs such as ImageNet ($3 \times 299 \times 299$) practically infeasible

Low Dimensional Subspace for ℓ_2

- Bayes Attack utilizes low-frequency FFT basis vectors to generate ℓ_2 norm constrained adversarial perturbations.

$$\text{FFT:} \qquad \mathrm{X}[u,v] = \frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1} \mathrm{x}[i,j] \exp\left[-j\frac{2\pi}{d}(u \cdot i + v \cdot j)\right]$$

$$\text{IFFT:} \qquad \mathrm{x}[i,j] = \frac{1}{d}\sum_{u=0}^{d-1}\sum_{v=0}^{d-1} \mathrm{X}[u,v] \exp\left[j\frac{2\pi}{d}(u \cdot i + v \cdot j)\right]$$

Low Dimensional Subspace for ℓ_2

- Bayes Attack utilizes low-frequency FFT basis vectors to generate ℓ_2 norm constrained adversarial perturbations.

$$\text{FFT:} \qquad \mathrm{X}[u,v] = \frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1}\mathrm{x}[i,j]\exp\left[-j\frac{2\pi}{d}(u\cdot i + v\cdot j)\right]$$

$$\text{IFFT:} \qquad \mathrm{x}[i,j] = \frac{1}{d}\sum_{u=0}^{d-1}\sum_{v=0}^{d-1}\mathrm{X}[u,v]\exp\left[j\frac{2\pi}{d}(u\cdot i + v\cdot j)\right]$$

- Equivalent norm: $\|\mathrm{x}\|_2 = \|\mathrm{FFT}(\mathrm{x})\|_2$, $\|\mathrm{X}\|_2 = \|\mathrm{IFFT}(\mathrm{X})\|_2$

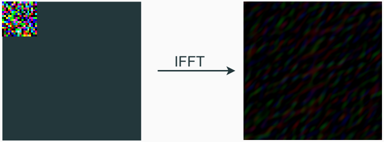
Low Dimensional Subspace for ℓ_2

- Bayes Attack utilizes low-frequency FFT basis vectors to generate ℓ_2 norm constrained adversarial perturbations.

$$\textbf{FFT:} \quad \mathsf{X}[u,v] = \frac{1}{d}\sum_{i=0}^{d-1}\sum_{j=0}^{d-1} \mathsf{x}[i,j] \exp\left[-j\frac{2\pi}{d}(u \cdot i + v \cdot j)\right]$$

$$\textbf{IFFT:} \quad \mathsf{x}[i,j] = \frac{1}{d}\sum_{u=0}^{d-1}\sum_{v=0}^{d-1} \mathsf{X}[u,v] \exp\left[j\frac{2\pi}{d}(u \cdot i + v \cdot j)\right]$$

- Equivalent norm: $\|\mathsf{x}\|_2 = \|\mathrm{FFT}(\mathsf{x})\|_2$, $\|\mathsf{X}\|_2 = \|\mathrm{IFFT}(\mathsf{X})\|_2$
- To allow only low-frequencies, the top-left $\lfloor rd \rfloor \times \lfloor rd \rfloor$, $r \in (0,1]$ square of X have nonzero entries

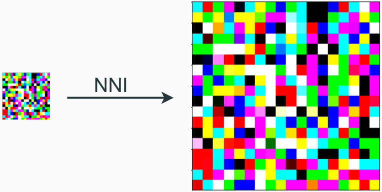


Low Resolution Subspace for ℓ_∞

- Bayes Attack utilizes spatial local similarity in images to generate ℓ_∞ norm constrained adversarial perturbations.

Low Resolution Subspace for ℓ_∞

- Bayes Attack utilizes spatial local similarity in images to generate ℓ_∞ norm constrained adversarial perturbations.
- We search for perturbations in a lower resolution image space $\lfloor rd \rfloor \times \lfloor rd \rfloor, r \in (0, 1]$ and use nearest neighbor interpolation.
- Equivalent norms: $\|\mathsf{X}\|_\infty = \|\mathrm{NNI}(\mathsf{X})\|_\infty$



Algorithm Black-box Adversarial Attack using Bayesian Optimization

1: **procedure** BAYES-ATTACK(x_0, y_0)
2: $\mathcal{D} = \{(\boldsymbol{\delta}_1, v_1), \cdots, (\boldsymbol{\delta}_{n_0}, v_{n_0})\}$ ▷ Quering randomly chosen n_0 points.

Bayes Attack

Algorithm Black-box Adversarial Attack using Bayesian Optimization

```
procedure BAYES-ATTACK(x_0, y_0)
    D = {(δ_1, v_1), ..., (δ_{n_0}, v_{n_0})}          ▷ Quering randomly chosen n_0 points.
    Update the GP on D                                 ▷ Updating posterior distribution
    t ← n_0                                            ▷ Updating number of queries
    while t ≤ T do
        δ_t ← arg max_δ A(δ | D)                       ▷ Optimizing the acquisition function



        t ← t + 1
        if v_t < 0 then
            D ← D ∪ (δ_t, v_t) and update the GP       ▷ Updating posterior distribution
```

Bayes Attack

Algorithm Black-box Adversarial Attack using Bayesian Optimization

procedure BAYES-ATTACK(x_0, y_0)
    $\mathcal{D} = \{(\boldsymbol{\delta}_1, v_1), \cdots, (\boldsymbol{\delta}_{n_0}, v_{n_0})\}$ ▷ Quering randomly chosen n_0 points.
    Update the GP on $\mathcal{D}$ ▷ Updating posterior distribution
    $t \leftarrow n_0$ ▷ Updating number of queries
    while $t \leq T$ **do**
        $\boldsymbol{\delta}_t \leftarrow \arg\max_{\boldsymbol{\delta}} \mathcal{A}(\boldsymbol{\delta} \mid \mathcal{D})$ ▷ Optimizing the acquisition function
        $\boldsymbol{\delta}_t \leftarrow \Pi^{p}_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta}_t)$ ▷ Projecting perturbation on ℓ_p-ball
        $\boldsymbol{\Delta}_t \leftarrow \text{map}(\boldsymbol{\delta}_t)$ ▷ Mapping perturbation to full input space
        $v_t \leftarrow f(\mathbf{x}_0, y_0, \boldsymbol{\Delta}_t)$ ▷ Querying the model
        $t \leftarrow t + 1$
        if $v_t < 0$ **then**
            $\mathcal{D} \leftarrow \mathcal{D} \cup (\boldsymbol{\delta}_t, v_t)$ and update the GP ▷ Updating posterior distribution

Bayes Attack

Algorithm Black-box Adversarial Attack using Bayesian Optimization

procedure BAYES-ATTACK(x_0, y_0)
  $\mathcal{D} = \{(\boldsymbol{\delta}_1, v_1), \cdots, (\boldsymbol{\delta}_{n_0}, v_{n_0})\}$ ▷ Quering randomly chosen n_0 points.
  Update the GP on $\mathcal{D}$ ▷ Updating posterior distribution
  $t \leftarrow n_0$ ▷ Updating number of queries
  while $t \leq T$ **do**
    $\boldsymbol{\delta}_t \leftarrow \arg\max_{\boldsymbol{\delta}} \mathcal{A}(\boldsymbol{\delta} \mid \mathcal{D})$ ▷ Optimizing the acquisition function
    $\boldsymbol{\delta}_t \leftarrow \Pi^{p}_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta}_t)$ ▷ Projecting perturbation on ℓ_p-ball
    $\boldsymbol{\Delta}_t \leftarrow \text{map}(\boldsymbol{\delta}_t)$ ▷ Mapping perturbation to full input space
    $v_t \leftarrow f(\mathrm{x}_0, y_0, \boldsymbol{\Delta}_t)$ ▷ Querying the model
    $t \leftarrow t + 1$
    if $v_t < 0$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup (\boldsymbol{\delta}_t, v_t)$ and update the GP ▷ Updating posterior distribution
    else
      return $\boldsymbol{\delta}_t$ ▷ Adversarial attack successful

Bayes Attack

Algorithm Black-box Adversarial Attack using Bayesian Optimization

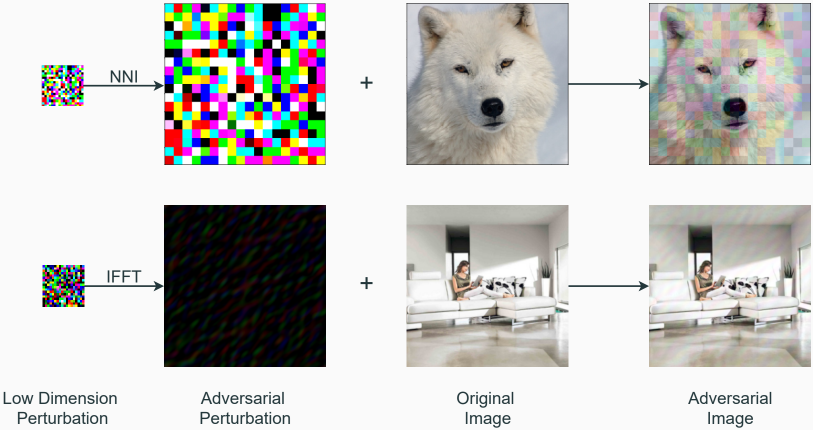
procedure BAYES-ATTACK(x_0, y_0)
  $\mathcal{D} = \{(\boldsymbol{\delta}_1, v_1), \cdots, (\boldsymbol{\delta}_{n_0}, v_{n_0})\}$ ▷ Quering randomly chosen n_0 points.
  Update the GP on $\mathcal{D}$ ▷ Updating posterior distribution
  $t \leftarrow n_0$ ▷ Updating number of queries
  while $t \leq T$ **do**
    $\boldsymbol{\delta}_t \leftarrow \arg\max_{\boldsymbol{\delta}} \mathcal{A}(\boldsymbol{\delta} \mid \mathcal{D})$ ▷ Optimizing the acquisition function
    $\boldsymbol{\delta}_t \leftarrow \Pi^{p}_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta}_t)$ ▷ Projecting perturbation on ℓ_p-ball
    $\boldsymbol{\Delta}_t \leftarrow \text{map}(\boldsymbol{\delta}_t)$ ▷ Mapping perturbation to full input space
    $v_t \leftarrow f(\mathrm{x}_0, y_0, \boldsymbol{\Delta}_t)$ ▷ Querying the model
    $t \leftarrow t + 1$
    if $v_t < 0$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup (\boldsymbol{\delta}_t, v_t)$ and update the GP ▷ Updating posterior distribution
    else
      return $\boldsymbol{\delta}_t$ ▷ Adversarial attack successful
  return $\boldsymbol{\delta}_t$ ▷ Adversarial attack unsuccessful

Bayes Attack

Algorithm Black-box Adversarial Attack using Bayesian Optimization

procedure BAYES-ATTACK(x_0, y_0)
  $\mathcal{D} = \{(\boldsymbol{\delta}_1, v_1), \cdots, (\boldsymbol{\delta}_{n_0}, v_{n_0})\}$ ▷ Quering randomly chosen n_0 points.
  Update the GP on $\mathcal{D}$ ▷ Updating posterior distribution
  $t \leftarrow n_0$ ▷ Updating number of queries
  while $t \leq T$ **do**
    $\boldsymbol{\delta}_t \leftarrow \arg\max_{\boldsymbol{\delta}} \mathcal{A}(\boldsymbol{\delta} \mid \mathcal{D})$ ▷ Optimizing the acquisition function
    $\boldsymbol{\delta}_t \leftarrow \Pi^p_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta}_t)$ ▷ Projecting perturbation on ℓ_p-ball
    $\boldsymbol{\Delta}_t \leftarrow \text{map}(\boldsymbol{\delta}_t)$ ▷ Mapping perturbation to full input space
    $v_t \leftarrow f(\mathbf{x}_0, y_0, \boldsymbol{\Delta}_t)$ ▷ Querying the model
    $t \leftarrow t + 1$
    if $v_t < 0$ **then**
      $\mathcal{D} \leftarrow \mathcal{D} \cup (\boldsymbol{\delta}_t, v_t)$ and update the GP ▷ Updating posterior distribution
    else
      return $\boldsymbol{\delta}_t$ ▷ Adversarial attack successful
  return $\boldsymbol{\delta}_t$ ▷ Adversarial attack unsuccessful

Bayes Attack



Experiments

- Untargeted and targeted attacks

Experiments

- Untargeted and targeted attacks
- ℓ_2 and ℓ_∞ threat models

Experiments

- Untargeted and targeted attacks
- ℓ_2 and ℓ_∞ threat models
- MNIST, CIFAR-10 - 4 convolution and 2 fully-connected layers
- ImageNet - ResNet50, VGG16-bn, Inception-v3

Experiments

- Untargeted and targeted attacks
- ℓ_2 and ℓ_∞ threat models
- MNIST, CIFAR-10 - 4 convolution and 2 fully-connected layers
- ImageNet - ResNet50, VGG16-bn, Inception-v3
- Compare with Boundary attack[2], OPT[3], Sign-OPT[4] and HSJA[5]

---

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248 (2017).

[3] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. 2019. Query-efficient hard-label black-box attack: An optimization-based approach. In International Conference on Learning Representations.

[4] Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. 2020. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. In International Conference on Learning Representations.

[5] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. 2019. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. ArXiv abs/1904.02144 (2019).

Untargeted ℓ_∞ attacks on ImageNet

$\epsilon = 0.05$, Query budget = 1000

	ResNet50		Inception-v3		VGG16-bn	
Method	success	avg. query	success	avg. query	success	avg. query
OPT attack	5.73	246.31	2.87	332.17	7.53	251.21
Sign-OPT	10.31	660.37	7.51	706.3	15.85	666.87
Bayes attack	**67.48**	**45.94**	**44.29**	**72.31**	**78.47**	**33.7**

Untargeted ℓ_∞ attacks on MNIST and CIFAR-10

MNIST ($\epsilon = 0.3$) and CIFAR-10 ($\epsilon = 0.05$), Query budget = 1000.

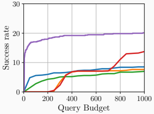
Method	MNIST		CIFAR-10	
	success	avg. query	success	avg. query
OPT attack	2.91	657.93	12.55	271.24
Sign-OPT	7.02	682.36	31.87	679.39
Bayes attack	**90.35**	**27.56**	**70.38**	**75.88**

Targeted ℓ_∞ attacks on MNIST and CIFAR-10

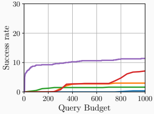
MNIST ($\epsilon = 0.3$) and CIFAR-10 ($\epsilon = 0.1$), Query budget = 1000.

Method	MNIST		CIFAR-10	
	success	avg. query	success	avg. query
OPT attack	0.0	–	0.0	–
Sign-OPT	2.41	975.67	3.50	937.65
Bayes attack	**26.23**	**130.03**	**48.93**	**149.15**

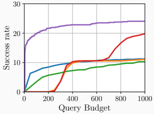
Untargeted ℓ_2 attacks on ImageNet



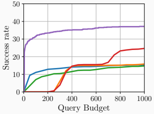
(a) ResNet50, $\epsilon = 5.0$



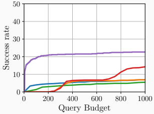
(b) Inception-v3, $\epsilon = 5.0$



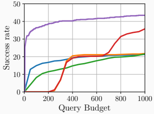
(c) VGG16-bn, $\epsilon = 5.0$



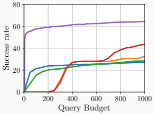
(d) ResNet50, $\epsilon = 10.0$



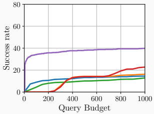
(e) Inception-v3, $\epsilon = 10.0$



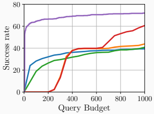
(f) VGG16-bn, $\epsilon = 10.0$



(g) ResNet50, $\epsilon = 20.0$

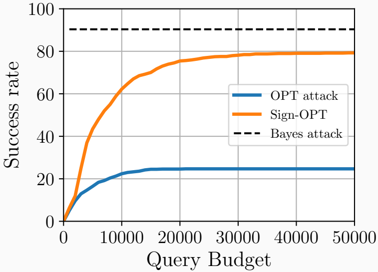


(h) Inception-v3, $\epsilon = 20.0$



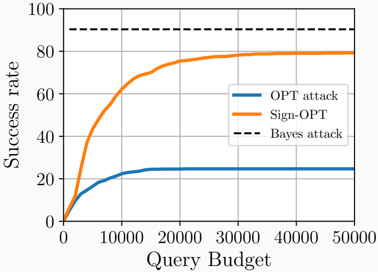
(i) VGG16-bn, $\epsilon = 20.0$

Query Efficiency Comparison

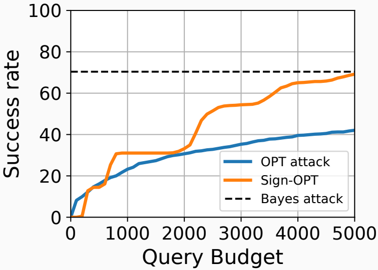


(a) MNIST

Query Efficiency Comparison



(a) MNIST



(b) CIFAR-10

Low Dimension Subspaces

Performance comparison of FFT basis vectors and random vectors sampled from the standard normal distribution for ℓ_2 attack with $\epsilon = 20.0$ on ResNet50.

Basis	Success	Avg Queries
Cosine FFT	64.38%	54.25
Sine FFT	63.74%	45.72
Cosine and sine FFT	66.67%	54.97
Standard Normal	33.33%	48.25

Conclusions

- We consider the problem of hard-label black-box adversarial attacks in low query budget regimes which is an important practical consideration.

Conclusions

- We consider the problem of hard-label black-box adversarial attacks in low query budget regimes which is an important practical consideration.
- We show that BO presents as a scalable, query-efficient alternative for black-box adversarial attacks when combined with searching in structured low dimensional subspaces.

Conclusions

- We consider the problem of hard-label black-box adversarial attacks in low query budget regimes which is an important practical consideration.
- We show that BO presents as a scalable, query-efficient alternative for black-box adversarial attacks when combined with searching in structured low dimensional subspaces.
- We successfully demonstrate the efficacy of our method in attacking multiple deep learning architectures in both untargeted and targeted settings, and ℓ_∞ and ℓ_2 norms.

Thank You.

- Poster Session: 18th August 2021, 05:30 pm - 08:30 pm EST
- Code: `https://github.com/satyanshukla/bayes_attack`
- Paper: `https://arxiv.org/pdf/2007.07210.pdf`
- Contact: `snshukla@cs.umass.edu`