Adversarial examples (black-box)

Neil Gong

Slides credit: part of the slides were adapted from Xiaoyu Cao

Black-box attacks

- The attacker does not have access to the model parameters θ.

Attacker's goal

- Given C, x and t, find x' such that $C(x') = t$.
- x' should be semantically the same as x.
  - Approximation: $||x' - x||_p$ should be small.
- The cost should be acceptable
  - The number of queries to the prediction API should be small.
  - The computational cost should be limited.

Idea 1

- Learn a surrogate model C' and rely on the transferability of adversarial examples.

  - Surrogate model: a model learnt to mimic the target model, when the target model is not directly available.

  - Transferability: the ability of an adversarial example x' generated against C' transfering to C, i.e., x' is also an adversarial example against C.

Idea 2

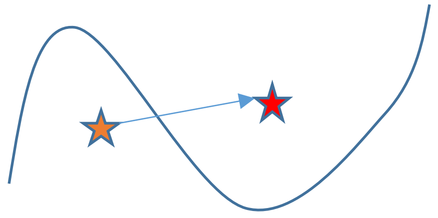
- Zeroth-order optimization
  - Optimizing an objective function f based only on access to function values $f(x)$.
  - Gradient estimation methods.
  - Trial-and-error methods.

Two ideas of black-box attacks

- Surrogate model based methods

- Zeroth-order optimization methods

Surrogate model based methods



- Classification boundary
  - A classifier can be uniquely identified by its classification boundary.

- Assumption
  - Different classifiers have similar classification boundary.

- However, the classification boundary for neural networks are too complicated to theoretically analyze.

Increase the probability of transfer

- Find adversarial examples that transfer to more surrogate classifiers.
    - It is more likely that such adversarial examples can transfer to the target classifier.
    - *Paper: Delving into Transferable Adversarial Examples and Black-box Attacks*

- Learn a surrogate model that better approximate the classification boundary of the target model.
    - *Paper: Practical Black-Box Attacks against Machine Learning*

Delving into Transferable Adversarial Examples and Black-box Attacks

- Threat model
  - The attacker has some training data.
  - The attacker can learn multiple surrogate models.

- Discovery: naïve transfer is not effective

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
ResNet-152	1.25	0%	86%	87%	93%	96%
ResNet-101	1.24	84%	0%	93%	95%	100%
ResNet-50	1.21	90%	91%	0%	91%	97%
VGG-16	1.55	89%	94%	92%	0%	84%
GoogLeNet	1.27	94%	97%	98%	91%	0%

Ensemble-based transfer attack

- Train multiple surrogate models.
- If an adversarial example transfers to all the surrogate models, then w.h.p. it can also transfer to the target model.
- Given k surrogate models $C_1, \dots, C_k$, an initial example x, a target label t and a target classifier C

$$\min_{x'} \; L(C_1(x'), \dots, C_k(x'); t) + \lambda d(x', x)$$

Ensemble-based transfer attack

	RMSD	ResNet-152	ResNet-101	ResNet-50	VGG-16	GoogLeNet
-ResNet-152	17.17	0%	0%	0%	0%	0%
-ResNet-101	17.25	0%	1%	0%	0%	0%
-ResNet-50	17.25	0%	0%	2%	0%	0%
-VGG-16	17.80	0%	0%	0%	6%	0%
-GoogLeNet	17.41	0%	0%	0%	0%	5%

Ensemble-based transfer attack

- Strength
  - No need to query.
  - The generated adversarial examples are applicable to any target classifier.

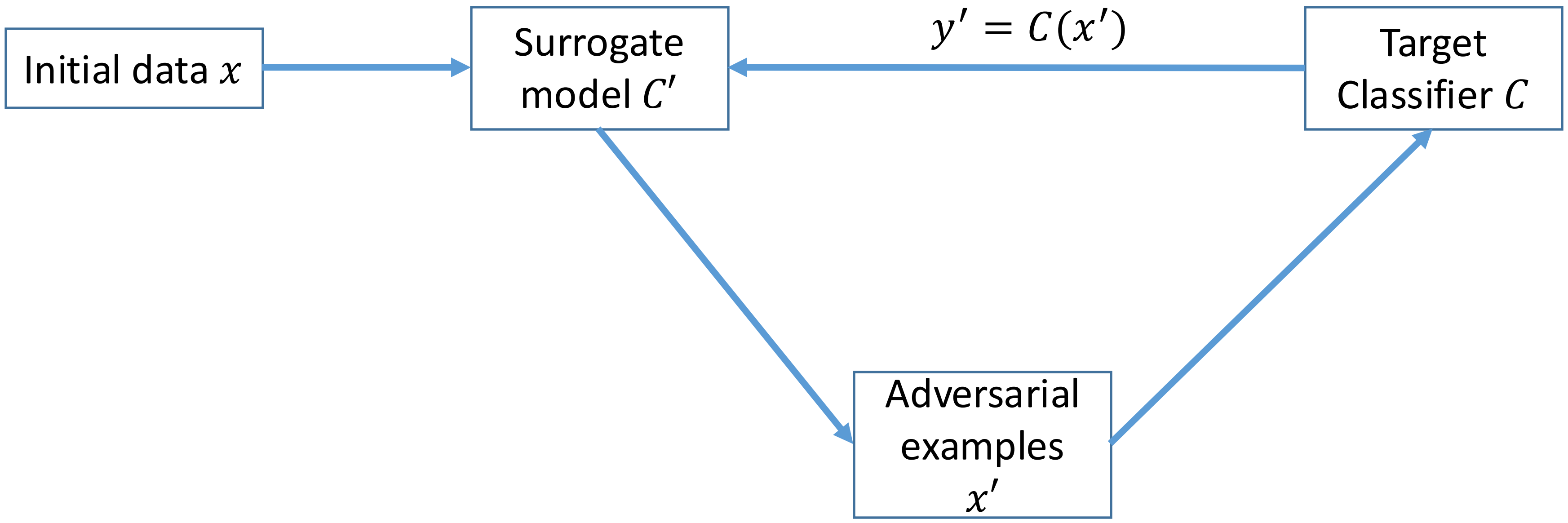
- Weakness
  - There is no theoretical guarantee on the transferability.
    - The success rate for targeted attacks is not as good as untargeted attacks.

Practical Black-Box Attacks against Machine Learning

- Threat model
  - The attacker feeds synthetic data points x into C and fetch the output label y.
  - The attacker can train a surrogate model using (x, y).

- Learn a surrogate model that better approximates the local classification boundary of the target classifier.
  - Iterative methods.

Surrogate model Training



Practical black-box attack

- Strength
    - The trained surrogate model can be used to generate future adversarial examples. No further queries are needed.

- Weakness
    - If we only care about few adversarial examples, the cost is huge – a lot of queries and training.
    - No theoretical guarantee on transferability.

Zeroth-order optimization

- Threat model
  - The attacker queries the target classifier for some output, which can be either a label y or a probability vector $\boldsymbol{p}$.
- General approaches
  - Gradient estimation
  - Trial and error

Gradient estimation

Black-box Adversarial Attacks with Limited Queries and Information

- In the white-box setting, the attacker solves an optimization problem to generate adversarial examples.

- They use first-order optimization methods, i.e., the derivative is known.

- However, in the black-box setting, we cannot directly obtain the gradients.

Natural Evolution Strategies (NES) gradient estimation

Algorithm 1 NES Gradient Estimate

Input: Classifier $P(y|x)$ for class y, image x
Output: Estimate of $\nabla P(y|x)$
Parameters: Search variance σ, number of samples n, image dimensionality N

$g \leftarrow \mathbf{0}_n$
for $i = 1$ **to** n **do**
  $u_i \leftarrow \mathcal{N}(\mathbf{0}_N, \boldsymbol{I}_{N \cdot N})$
  $g \leftarrow g + P(y|x + \sigma \cdot u_i) \cdot u_i$
  $g \leftarrow g - P(y|x - \sigma \cdot u_i) \cdot u_i$
end for
return $\frac{1}{2n\sigma} g$

Gradient estimation methods

- Strength
  - Query-efficient when the number of adversarial examples is small.
  - Builds a bridge between white-box and black-box attacks. With the estimated gradients, one can apply any white-box technics.

- Weakness
  - Needs access to the probability vector. If only a predicted label is available, there is a solution but the attack becomes less efficient.
  - When a lot of adversarial examples are needed, it becomes costly.

Trial-and-error methods

Simple Black-box Adversarial Attacks

Algorithm 1 SimBA in Pseudocode

---

procedure $\textsc{SimBA}(\mathbf{x}, y, Q, \epsilon)$
    $\delta = \mathbf{0}$
    $\mathbf{p} = p_h(y \mid \mathbf{x})$
    while $\mathbf{p}_y = \max_{y'} \mathbf{p}_{y'}$ **do**
        Pick randomly without replacement: $\mathbf{q} \in Q$
        for $\alpha \in \{\epsilon, -\epsilon\}$ **do**
            $\mathbf{p}' = p_h(y \mid \mathbf{x} + \delta + \alpha\mathbf{q})$
            if $\mathbf{p}'_y < \mathbf{p}_y$ **then**
                $\delta = \delta + \alpha\mathbf{q}$
                $\mathbf{p} = \mathbf{p}'$
                break
  return δ

---

orthonormal candidate vectors

SimBA

- Strength
  - Query-efficient when the number of adversarial examples is small.
  - Simple but effective.
- Weakness
  - Needs access to the probability vector.
  - When a lot of adversarial examples are needed, it becomes costly.

Combination of both

HopSkipJumpAttack: A Query-Efficient Decision-Based Attack

- Combines trial-and-error and gradient estimation.

- An improved version of Boundary Attack.

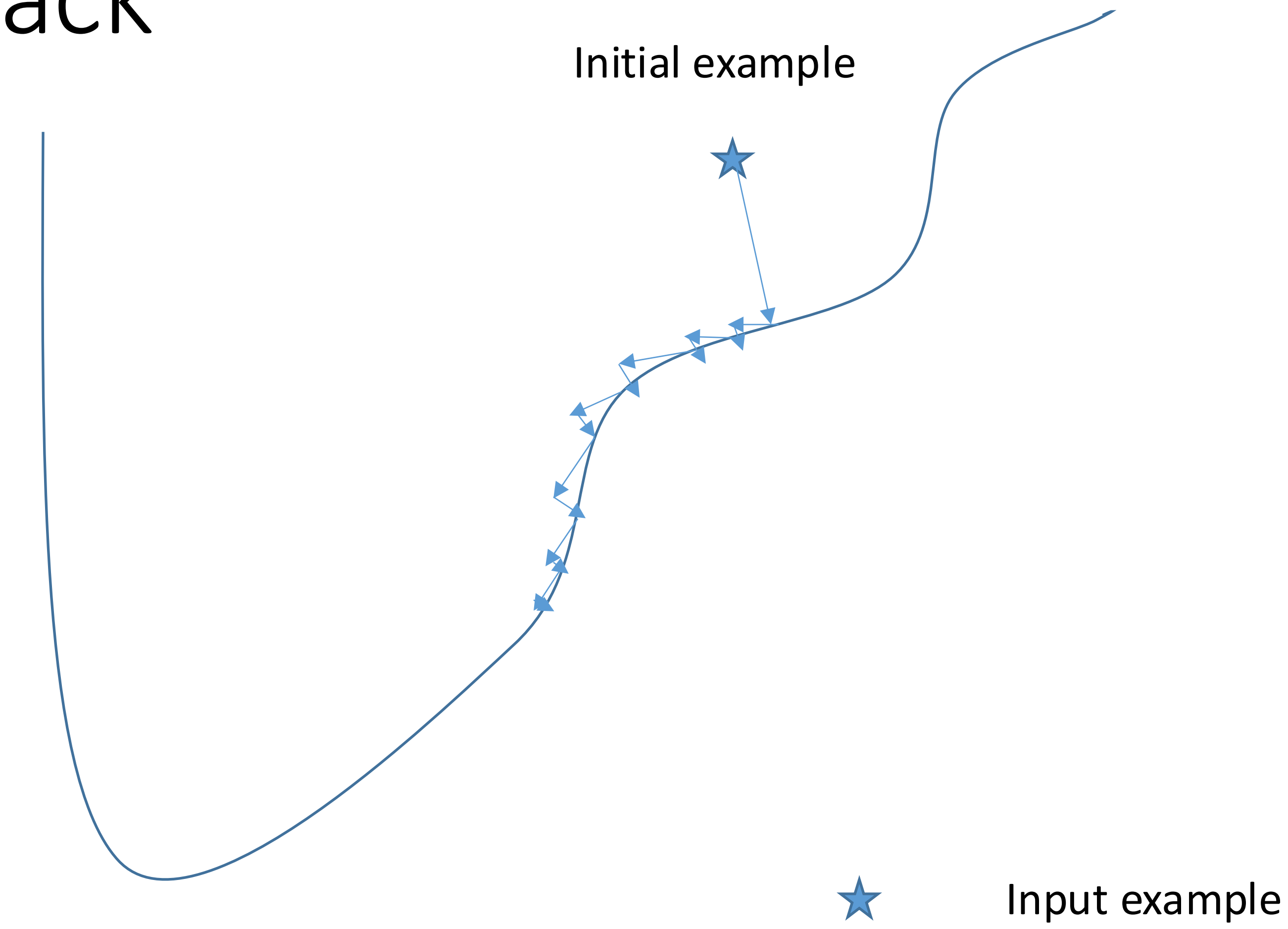
- What is Boundary Attack?

Boundary Attack

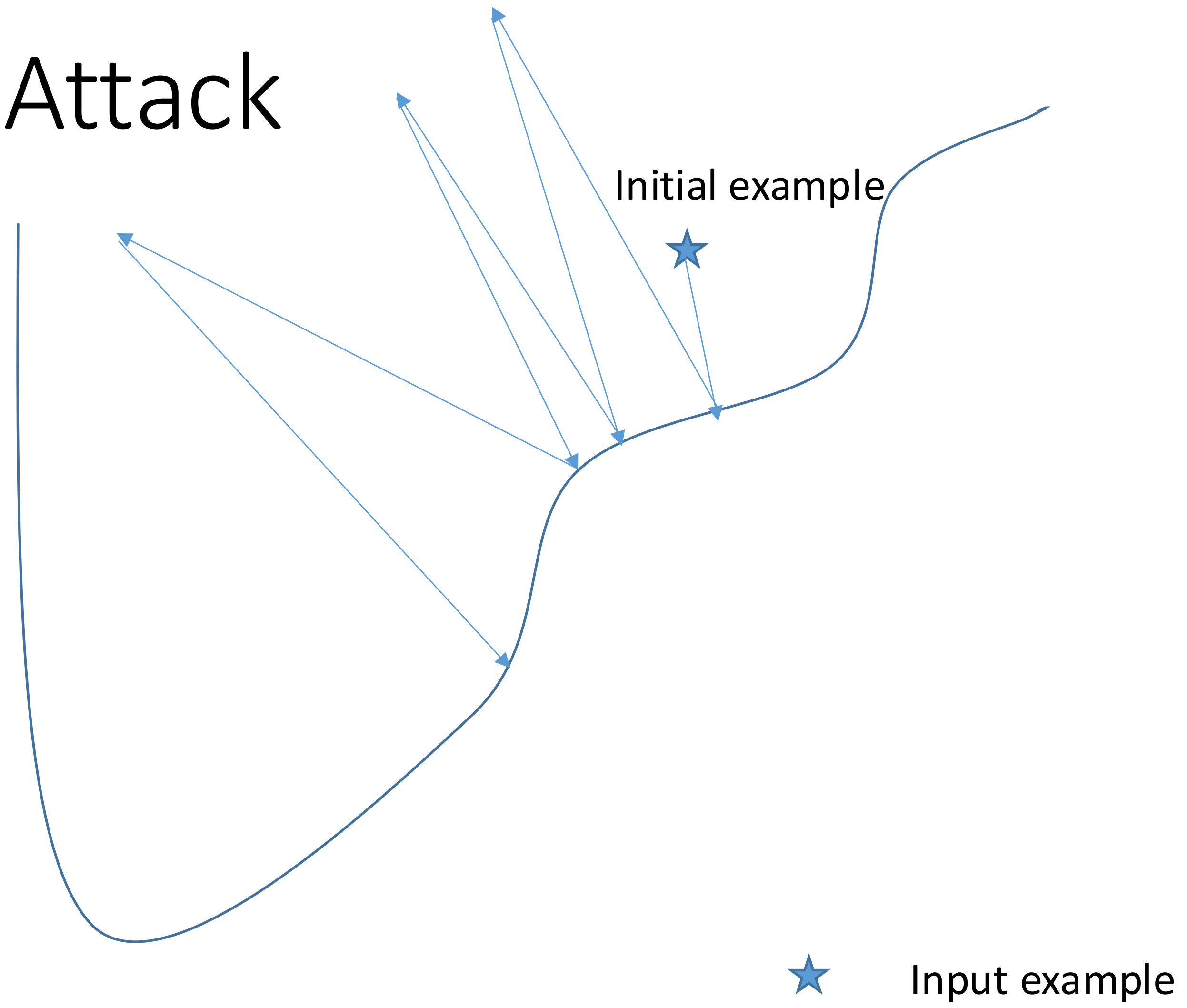
- Essentially a trial-and-error method.

- Different from other methods of the same type.
  - Starts from an initial example with the target label.
  - Iteratively move towards the input example along the decision boundary, until the distance to the input example does not decrease.

Boundary attack



HopSkipJumpAttack



HopSkipJumpAttack

- Strength
  - Query-efficient when the number of adversarial examples is small.
  - Only needs access to the predicted labels.
  - Theoretical analysis on the gradient estimation.

- Weakness
  - When a lot of adversarial examples are needed, it becomes costly.