Towards Evaluating the Robustness of Neural Networks

Nicholas Carlini, David Wagner

Presented by Xianda ‘Claude’ Zhou

Contribution

- Introduced a new class of attacks
  - Image Classification
  - White Box
  - Targeted
- Nullified defensive distillation (then state-of-the art defense)
- Proposed a transferability test to evaluate defenses

Key Points

- **Using gradient decent to find adversarial examples**
- The choice of the objective function is important

Using Gradient Descent - Formulation

- Classic formulation:

  Given input x, find x’ where:

  minimize D(x, x’)

  such that F(x’) = T

  x’ is a valid image

- Non-linear constraints not suitable for gradient descent

Using Gradient Descent - Reformulation

- Reformulation:

  Given input x, find x’ where:

  minimize $D(x, x') + c \cdot g(x', T)$ $(c>0)$

  such that x’ is a valid image

- Move the constraint into the objective
- g(x’, T) is some loss function on how close F(x’) is to T
  - $g(x', T) \leq 0$ when $F(x') = T$
  - $g(x', T) > 0$ when $F(x') \neq T$

Using Gradient Descent - Reformulation

- Reformulation:

  Given input x, find x’ where:

  minimize $D(x, x') + \mathbf{c} \cdot g(x', T)$ $(c>0)$

  such that **x’ is a valid image**

- Find the optimal ‘c’ by searching
- Simple tricks to force ‘box constraints’

- Post-Processing: Greedy procedure of rounding to integer

Key Points

- Using gradient decent to find adversarial examples
- **The choice of the objective function is important**

Choosing Objective Function

- $D(x, x') + c \cdot \mathbf{g}(x', t)$

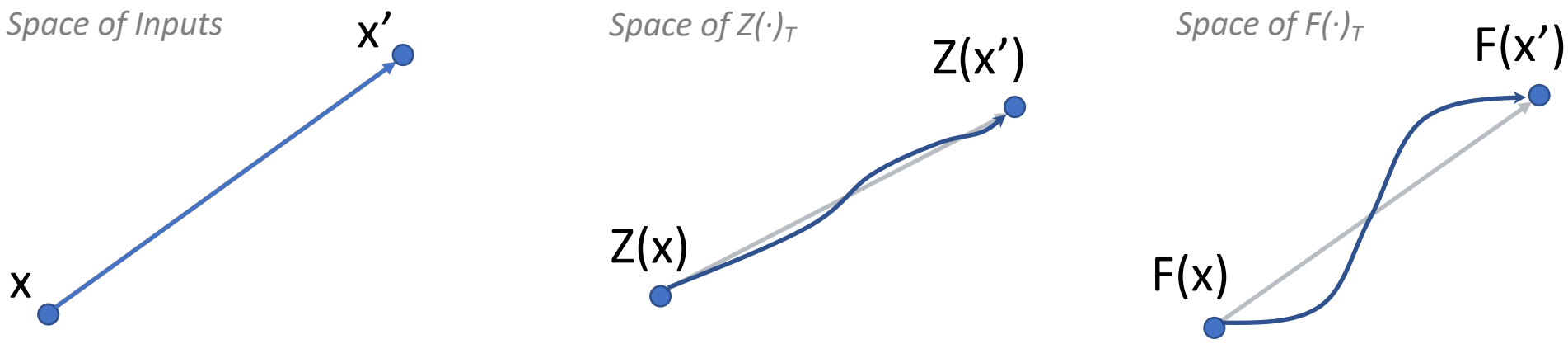
- Previously, **g** is mostly based on label probability or cross entropy loss
- *Eg.* $g(x', T) = 0.5 - F(x')_T$
  - If F(x') says the probability of T is 1:
    g(x', T) = -0.5
  - If F(x') says the probability of T is 0:
    g(x', T) = 0.5

Choosing Objective Function

- $F(\cdot) = \text{softmax}(Z(\cdot))$
  Z computes logits
- Investigated 7 objective functions
- Objective functions based on Z work best
  - 100% guaranteed to find an adversarial example
  - L_2 distance much smaller
- Objective functions based on F fail to find adversarial examples sometimes

Choosing Objective Function

- It is observed that
  - Given input x, adversarial example x’
  - $Z(\cdot)_T$ is **mostly linear** from x to x’. Therefore,
  - $F(\cdot)_T$ is **logistic** from x to x’

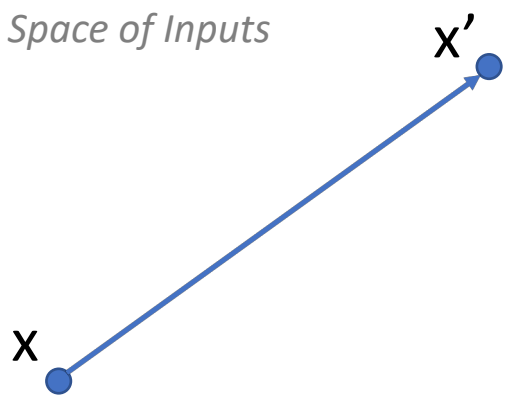


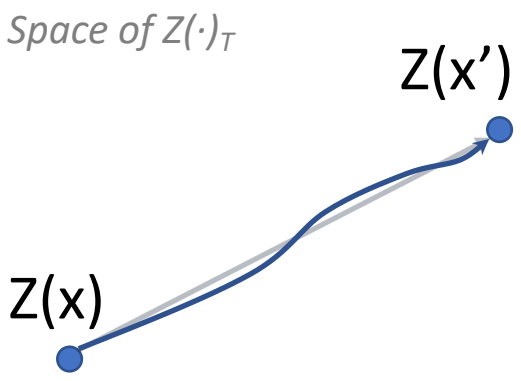
Choosing Objective Function

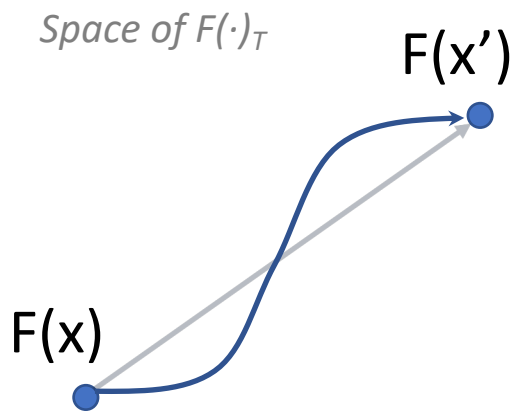
- If we use objective func based on $F(\cdot)_T$

  $D(x, x') + c \cdot g(x', t)$

  - Need a large ‘c’ to start moving in initial steps
  - Need a smaller ‘c’ to balance the two terms
  - **Impossible to find a proper constant ‘c’**







Formulation

- Formulation:

  Given input x, find x’ where:

  minimize $\mathbf{D}(x, x') + c \cdot g(x', t)$

  such that x’ is a valid image

- Investigated L_0, L_2, L_∞ for distance computing
  - Techniques to bypass the non-differentiability of L_0 and L_∞

Results

- New attacks almost 100% guaranteed to find an adversarial example
- Distance smaller than previous attacks

- Attacks on Defensive Distillation
  - Almost 100% guaranteed to find an adversarial example on defensively distilled networks
  - Distance marginally larger than on unsecured networks
  - **Nullified Defensive Distillation**

Summary

- Defensive Distillation:
  - NNs are highly non-linear and have ‘blind spots’ where adversarial examples lie
  - remove those blind spots by preventing over-fitting
- C&W Attacks:
  - NNs are locally-linear so that adversarial examples can exist (Explaining and Harnessing Adversarial Examples [Goodfellow 2014])
- Previous attacks
  - broke local-linearity
  - failed on Defense Distillation mostly because it caused $F(\cdot)$ vanish to zero

Pros

1. Efficient Formulation

- Reformulate the task to a more efficient optimization problem

  Given input x, find x’ where:

  minimize D(x, x’)
  such that F(x’) = T
  x’ is a valid image

  minimize D(x, x’) + c · g(x’, T)
  such that x’ is a valid image

- Seen in previous work, but not fully explored

2. Simple Mathematical Observation

- The linearity problem with softmax-based objectives

- Reminiscent of model ensemble in Yearbook Challenge
  - Averaging logits works better than averaging probability

3. Evaluating Robustness by Attacks

- Two ways to evaluate robustness
  - Construct a proof
  - **Demonstrate constructive attacks**

- Seemingly on NN defense from its title
  - Make a contribution from the rival's side

4. Comprehensiveness

- Experiments
  - Implemented/experimented on all major attack techniques (Almost a comprehensive literature review)
  - Designed/implemented L_0, L_2, L_∞ for comparison
- Evaluation
  - Three approached to evaluate a targeted attack: Average/Best/Worst Case
- Rounding to Integers
  - Almost Never explored before

5. Controllability of Strength

- Hyperparameter ‘κ’ in the objective to control the strength of adversarial examples
  - The larger κ, the stronger the classification of the adversarial example.

Original

Dog (83%)

Adversarial

Hummingbird (90%)
κ=0

Hummingbird (99%)
κ=20

- Build transferability test
  - Broke Defensive Distillation (again) in black-box setting

Thank You!