https://qdata.github.io/deep2Read/

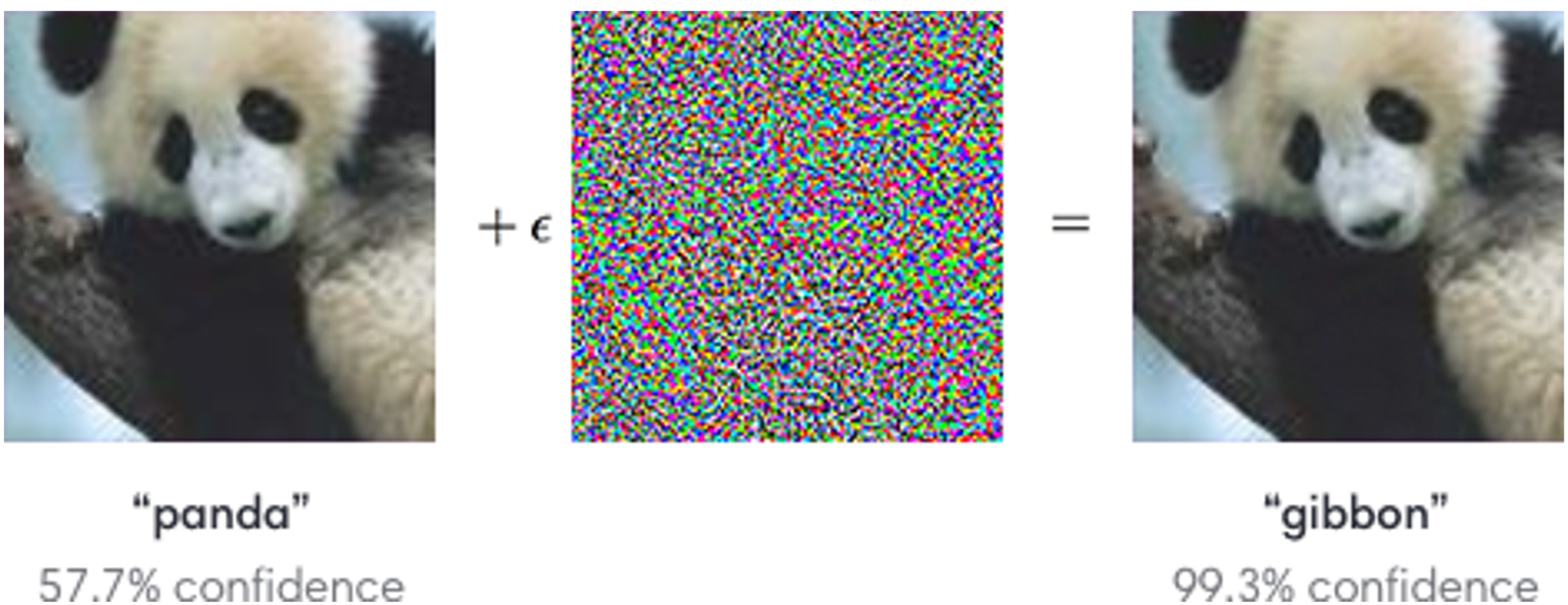
Parsimonious Black-Box Adversarial Attacks Via Efficient Combinatorial Optimization

Seungyong Moon, Gaon An, Hyun Oh Song

ICML 2019

Presented by Eli Lifland, 8/30/2019

Adversarial Perturbations



White vs Black Box Attacks

- White Box: access to parameters and therefore gradient
  - Fast Gradient Sign Method (FGSM): perturb in direction of gradient
  - Projected Gradient Descent (PGD): multiple iterations of FGSM
- Black Box: only query access
  - Substitute networks: train a network to match predictions of target network
  - Gradient estimation: directly estimate gradient via queries
    - Outperforms substitute networks

Motivation

- Focus on black-box, which is more realistic in practice
- Problems with current black-box methods
  - Substitute network attacks don’t always transfer to target networks
  - Robustness of gradient estimation affected by choice of hyperparameters
    - E.g. learning rate, decay rates, update rule

Problem formulation

- Create imperceptible perturbations x_{adv} under L_∞ radius with limited query budget to maximize loss
- Attacker only has access to loss function, l(x,y)

$$\underset{\|x_{adv}-x\|_\infty \le \epsilon}{\text{maximize}} \quad \ell(x_{adv})$$

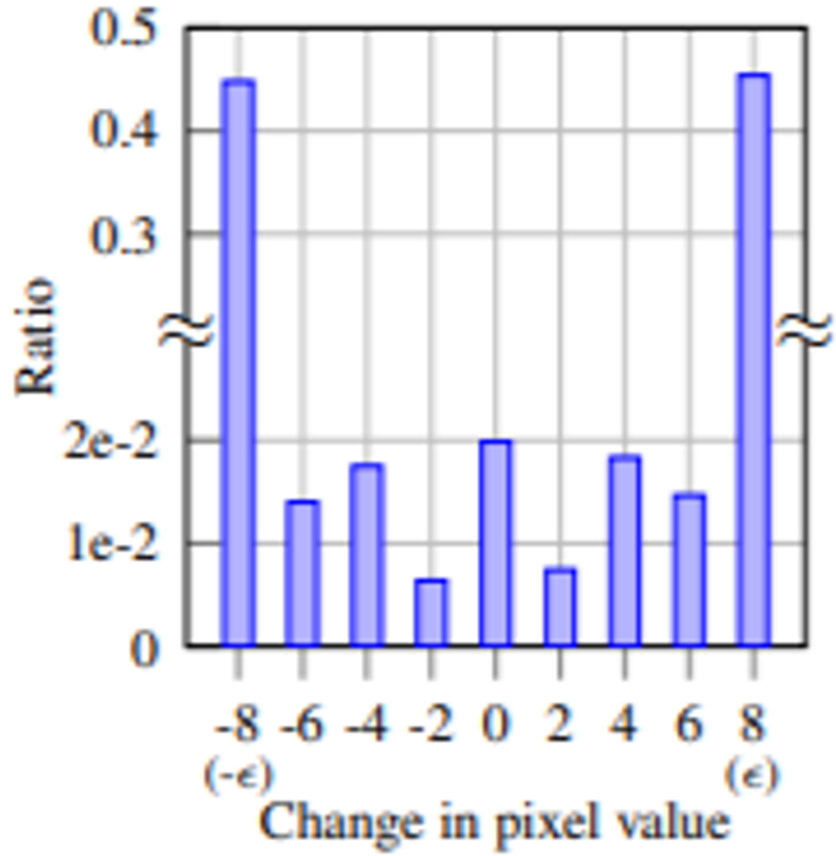
FGSM Approximation

$$\underset{\|x_{adv}-x\|_\infty \le \epsilon}{\text{maximize}} \; \ell(x_{adv}) \implies \underset{x_{adv}}{\text{maximize}} \; {x_{adv}}^\intercal \nabla_x \ell(x, y) \qquad (1)$$

$$\text{subject to} \quad -\epsilon\mathbf{1} \preceq x_{adv} - x \preceq \epsilon\mathbf{1},$$

- Where the ≤ is element-wise inequality and **1** is a vector of ones
- Optimal solution will be obtained at extreme point of feasible set, or a vertex of the L_∞ ball
- PGD on Cifar-10 does give solutions close to vertices of L_∞ ball

PGD Pixel-level Perturbations



Discrete formulation

- Only consider pixel perturbations of +/- ϵ

$$\underset{x_{adv}\in\mathbb{R}^p}{\text{maximize}}\ f(x_{adv}) \quad \Longrightarrow \quad \underset{x_{adv}}{\text{maximize}}\ f(x_{adv}) \qquad (2)$$

$$\text{subject to } \|x_{adv}-x\|_\infty \le \epsilon \qquad \text{subject to } x_{adv}-x \in \{\epsilon, -\epsilon\}^p,$$

- $f(x) = l(x, y_{gt})$ for untargeted attacks, $-l(x, y_{target})$ for targeted attacks
- Set maximization problem in which we choose from all pixels V a set S with +ϵ perturbations, with the rest having -ϵ perturbations

Submodularity

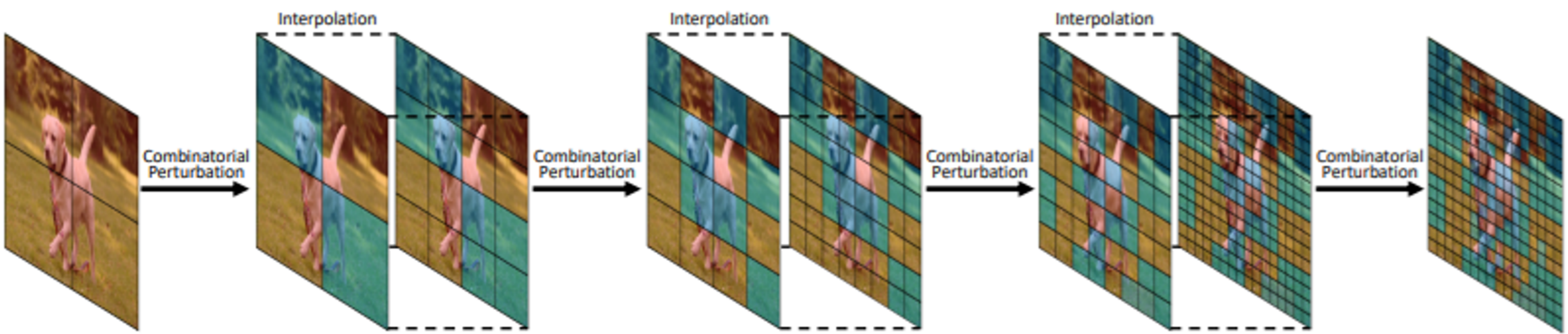
- Define F(S U {e}) - F(S) as the marginal gain from adding pixel e to S
  - Submodularity implies that this marginal gain will be smaller when S has more elements
  - “Diminishing returns”
- This is not completely true but algorithm assumes it is approximately true to cut save queries

Lazy Greedy Insertion

- First, query marginal gain for all elements not in S, and insert these elements into a max heap
  - this is treated as an upper bound because of submodularity
- While the heap isn't empty:
  - Pop the top element, update its upper bound
  - If it's greater than the new top element
    - If it's > 0, add it to S
    - else, end
  - If it's less than the new top element
    - Add it back to the heap

Implementation

- Exploit locally regular structure to do hierarchical evaluation
  - At each level, do one iteration of lazy insertion then lazy deletion
  - Terminate when converges or query limit reached



Diminishing gains

Submodular set functions are set functions who exhibit **diminishing returns**

for all $A \subseteq B$

$$F(A \cup e) - F(A) \geq F(B \cup e) - F(B)$$

Basically: as the size of the input set <u>increases</u>, the value that a single element adds <u>decreases</u>

Our problem: approximate submodularity

- As it turns out, our problem is not technically submodular
- However, as long as submodularity is not “severely deteriorated” (Zhou & Spanos, 2016), submodular maximization algorithms still work very well
- This means that we can compute an approximately optimal solution with a greedy algorithm!

Local-search optimization

- Notation
  - **P** is all pixels,
  - **S** is pixels to add **+ε** to
  - **P\S** are pixels to add **-ε**

- Basically, we can greedily choose to insert a pixel into S if the marginal gain is strictly positive, and remove it from S if the marginal gain is strictly negative
  - Then once the algorithm converges, it will converge to a **local optimum**

- End up with a set of pixels S to perturb the input image with +ε, and P\S to perturb with -ε

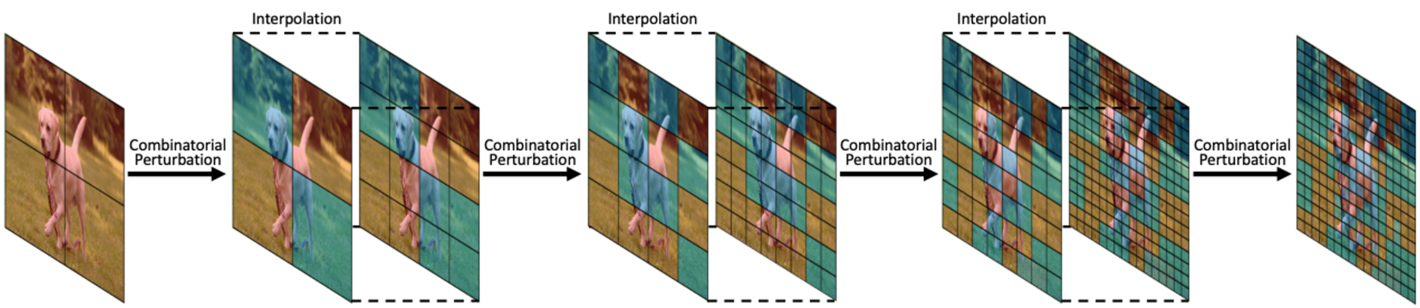
Speedup #1: Acceleration with lazy evaluations

- At each step we have to find the element that maximizes the marginal gain
  - Therefore, our greedy algorithm has to make **O(|P| |S|)** queries
  - This may be impractical for query-limited black-box attacks

- Speed this up: use the Lazy-Greedy algorithm (Minoux, 1978)
  - Instead of re-computing the marginal gain for each pixel at each iteration, keep the upper bounds on the marginal gains in a **max-heap**
  - Theoretically has the same worst-case number of function evaluations but provides a speedup of <u>several orders of magnitude</u> in practice!
  - Why? Because of submodularity! (wow)

Speedup #2:
Hierarchical lazy evaluation

- Exploit the locally regular structure of most images and do this on a hierarchical scale for another speed boost



Blue squares are in S, red squares are in P\S

Experimental Results

Method	Success rate	Avg. queries	Med. queries	Avg. queries (NES success)
PGD (white-box)	47.2%	20	-	-
NES	29.5%	2872	900	2872
Bandits	38.6%	1877	459	520
Ours	**48.0%**	**1261**	**356**	**247**

Table 1. Results for ℓ_∞ untargeted attacks on Cifar-10. Maximum number of queries set to 20,000.

Method	Success rate	Avg. queries	Med. queries	Avg. queries (NES success)
PGD (white-box)	99.9 %	20	-	-
NES†	77.8%	1735	-	1735
NES	80.3%	1660	900	1660
Bandits†	95.4%	1117	-	703
Bandits	94.9%	1030	286	603
Ours	**98.5%**	**722**	**237**	**376**

Table 2. Results for ℓ_∞ untargeted attacks on ImageNet. Maximum number of queries set to 10,000.

Method	Success rate	Avg. queries	Med. queries	Avg. queries (NES success)
PGD (white-box)	100%	200	-	-
NES†	99.2%	-	11550	-
NES	99.7%	16284	12650	16284
Bandits	92.3%	26421	18642	26421
Ours	**99.9%**	**7485**	**5373**	**7371**

Table 3. Results for ℓ_∞ targeted attacks on ImageNet. Maximum number of queries set to 100,000.

Conclusion

- Practical method for black-box adversarial attacks
- No gradient estimation required
  - No update hyperparameters
- State of the art success and query rates for both targeted and untargeted attacks