Adversarial attack methods

- White box attacks
  : The network is "transparent" to the attacker. Both the architecture and weights are known
- Black box attacks
  : The attackers has only access to the input and output of the network
- Gray box attacks
  : The attacker knows the network architectures but the weights

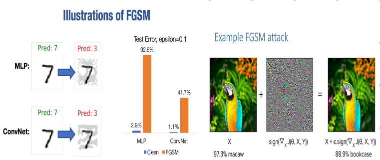
Goal: $\max_{\delta} loss(\theta, x+\delta, y)$, where θ: model parameters
x: input
y: correct label

Examples : δ that is small w.r.t
- ℓ_p norm
- Rotation and/or translation
- VGG (Visual Geometry group)

Evasion Attack 1: Fast Gradient Sign Method (FGSM) : White box attack

- Classifier (e.g. ResNet-50) : $\tilde{y} = f(\theta, x)$
- Find adversarial image x' that maximizes the loss: $L(x', y) = L(f(\theta, x'), y)$
- Bounded perturbation : $\|x' - x\|_\infty \leq \varepsilon$, ε : the attack strength

Optimal adversarial image: $x' = x + \varepsilon \cdot sign(\nabla_x L(x, y))$



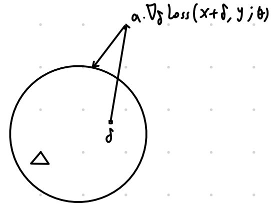
Evasion Attack 2: Iterative Fast Gradient Sign Method (IFGSM) : Whitebox attacks

- Similar to FGSM
- Generates enhanced attacks
  : $x^{(m)} = x^{(m-1)} + \varepsilon \cdot sign(\nabla_x L(x^{(m-1)}, y))$
  With $x^{(0)} = x$, and $y' = x^{(M)}$, where M is the number of iterations
- Both FGSM, and IFGSM are fix-perturbation attacks

Evasion Attack 3: Least Likely (LL) and Iterative Least Likely (ILL) Attacks : White box

- Similar to FGSM
  $x' = x - \varepsilon \cdot sign(\nabla_x L(x, y_{ll}))$, where y_{ll} is the least likely class predicted by the network on clean image x
- Strong attack as it emphasizes least likely class

Evasion Attack 4: Projected Gradient Descent (PGD) : White box



- Recall that we are optimizing : $\max_{\delta \in \Delta} Loss(x+\delta, y; \theta)$
- We can employ a projected gradient descent method, take gradient step and project back into feasible set Δ

  : $\delta := P_\Delta[\delta + \nabla_\delta Loss(x+\delta, y; \theta)]$
- Projected gradient descent applied to ℓ_∞ ball, repeat: $\delta := Clip_\epsilon[\delta + a\nabla_\delta J(\delta)]$
- Slower than FGSM (requires multiple iterations), but typically able to find better optima

Evasion Attack 5: Carlini and Wagner attack (CW) : White box

- zero confidence attack
- $\forall\ t \neq y$ find the adversarial image that will be classified as t by solving the problem : $\min_\delta \|\delta\|_2^2$

  subject to

  $f(x+\delta) = t,\ x+\delta \in [0,1]^n$
- Finding the exact solution is difficult

- CW-L2 attack
  - Relaxed version: $\min_\delta \|\delta\|_2^2 + c \cdot g(x+\delta)$

    s.t.

    $x+\delta \in [0,1]^n, c \geq 0$
  - Letting $Z(x)$ be the neural net activations before the output layer (logits)

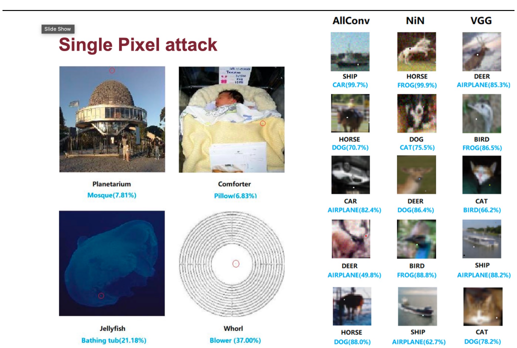
    $g(x) = \max\left(\max_{i \neq t}(Z(x)_i) - Z(x)_t, 0\right)$
  - resists many defense methods

Evasion Attack 6: Universal Adversarial Perturbation

: A single perturbation to fool a network on any image with a high probability (e.g. 0.8+)

- Perturbations generalize well across different models, posing a threat to Deep Learning in practice

Evasion Attack 7: Single Pixel Attack



Su et al. (2017). One Pixel Attack for Fooling Deep Neural Networks. *IEEE Trans. Ev. Comp.*

Adversarially Robust Neural Networks - Adversarial Training

Standard generalization: $\mathbb{E}_{(x,y)\sim D}[\text{loss}(\theta, x, y)]$

Adversarially robust generalization: $\mathbb{E}_{(x,y)\sim D}[\max_{\delta \in \Delta} \text{loss}(\theta, x+\delta, y)]$

The outer minimization problem

Inner Maximization:

$$\max_{\delta \in \Delta} \text{Loss}(x+\delta, y; \theta)$$

① Local search (lower bound on objective)

② Combinatorial Optimization

③ Convex relaxation (upper bound on objective)

Outer minimization

$$\min_{\theta} \sum_{x,y \in S} \max_{\delta \in \Delta} \text{Loss}(x+\delta, y, \theta)$$

① Adversarially training

② Robust training

Danskin's Theorem:

A fundamental result in optimization:

$$\nabla_\theta \max_{\delta \in \Delta} \text{Loss}(x+\delta, y; \theta) = \nabla_\theta \text{Loss}(x+\delta^*, y; \theta)$$

where $\delta^* = \max_{\delta \in \Delta} \text{Loss}(x+\delta, y; \theta)$

Seems obvious, but it is a very subtle result; means we can optimize through the max by just finding its maximizing value

Adversarial training (Goodfellow et al., 2014)

Repeat

1. Select minibatch B
2. For each $(x,y) \in B$, compute adversarial example $\delta^*(x)$
3. Update θ

$$\theta := \theta - \frac{a}{|B|} \sum_{x,y \in B} \nabla_\theta \text{Loss}(x+\delta^*(x), y; \theta)$$

Another core idea on Adversarial Robustness : Denoising

Defence Mechanisms - Defense GANs

Main idea : Train a GAN that generates unperturbed images. Instead of classifying a given input image, use the closest image generated by the GAN

Pros: Effective against white-box and black-box attacks

No accuracy drop (theoretically)

Cons: Complex method

difficult to train GAN.

other types of attack : Adversarial Noise

Adversarial Rotation

Adversarial Photographer.