Adversarial Examples in Machine Learning CS489/698

Rey Reza Wiyatno
Email: reywiyatno@gmail.com

Agenda

- Intro
- Adversarial Attacks
- Adversarial Defenses
- Takeaways

Agenda

- Intro
- Adversarial Attacks
- Adversarial Defenses
- Takeaways

Adversarial Examples

Inputs specifically designed to cause a model to make mistakes in its prediction, although they look like valid inputs to a human.

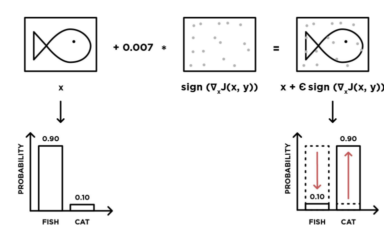


Illustration of adversarial example.

Source: https://medium.com/element-ai-research-lab/tricking-a-machine-into-thinking-youre-milla-jovovich-b19bf322d55c

Standard Definition in Classification Task

Generate adversarial example ***x'*** that looks like an original example ***x***, but is mispredicted by a model. Formally:

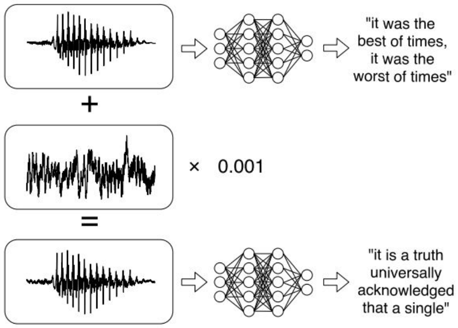
Given a model ***f*** and an input ***x***, find ***x'*** where $\|x - x'\|_p < \epsilon$, such that $f(x) \neq f(x')$

Note: other perceptual similarity metrics other than the L_p norm can also be used.

Adversarial Examples: Adversarial Patch

Adversarial Patch (Brown et al., 2017)

Adversarial Examples: Adversarial Audio



Demo: https://nicholas.carlini.com/code/audio_adversarial_examples/
Audio Adversarial Example: Targeted Attacks on Speech-to-Text (Carlini & Wagner, 2018)

Common Terms

Adversarial attacks: methods to generate adversarial examples.

Adversarial defenses: methods to defend against adversarial examples.

Adversarial robustness: property to resist misclassification of adversarial examples.

Adversarial detection: methods to detect adversarial examples.

Transferability: adversarial examples generated to fool a specific model can also be used to fool other models.

Common Terms

Adversarial perturbation: difference between original example and its adversarial counterpart

Whitebox attack: when attacker has full access to the victim model

Blackbox attack: when attacker only has access to victim's output

Targeted attack: when attacker wants an adversary to be mispredicted in a specific way

Non-targeted attack: when attacker does not care if an example is mispredicted in a specific way

Agenda

Intro

Adversarial Attacks

Adversarial Defenses

Takeaways

Goal:

Minimally modify inputs to confuse the victim model

We can re-formulate as:

Minimally modify inputs to maximize some loss function

Any ideas how?

Attack & Defense Methods

Attack strategies:

Whitebox:
- Direct gradient step(s): FGSM, BIM, R+FGSM, DAG
- Gradient-based greedy algorithm: JSMA
- Iterative optimization: L-BFGS, C&W, UAP, Adversarial Eyeglasses, RP_2, EOT
- Parameterized optimization: ATN

Blackbox:
- Decision boundary approximation: SBA
- Evolutionary algorithm: One Pixel Attack
- Finite difference method: ZOO

Defense strategies:

Data augmentation:
- Adversarial training
- Ensemble adversarial training
- PGD adversarial training

Gradient regularization:
- DCN
- Defensive distillation

Input manipulation:
- PixelDefend

Detection methods:
- Adversary detector networks
- Feature squeezing

Fast Gradient Sign Method (FGSM)

- Take **gradient w.r.t. input**
- Then do **single-step gradient ascent**

$$x' = x + \epsilon \cdot \mathrm{sign}(\nabla_x \mathcal{L}(x, y))$$

- Is above equation for targeted/non-targeted attack?
- If targeted, how to make it untargeted (and vice-versa)?
- Why use "sign" operator?

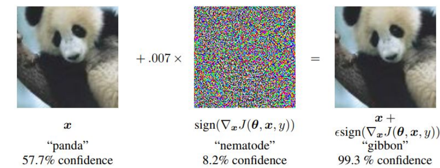


Illustration of FGSM. Adversarial example (right) misclassified as "Gibbon" with high confidence (Goodfellow et al., 2015).

Note: ***ℒ(x,y)*** and ***J(x,y)*** denote the training loss function, ***x*** is an input, ***y*** is the true label of ***x***, **ϵ** is a small constant where **ϵ** > 0

Explaining and Harnessing Adversarial Examples (Goodfellow et al., 2015)

Implementing FGSM (see notebook)

Link: https://github.com/rrwiyatn/deep-learning/blob/master/fast_gradient_sign_attack/fgsm.ipynb

Basic Iterative Method (BIM)

- Iterative variant of FGSM

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\Big\{\boldsymbol{X}_N^{adv} \boxed{+} \alpha\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}_N^{adv}, \boxed{y_{true}})\big)\Big\}$$

- Targeted attack variant of BIM called Iterative Least-Likely Class Method (ILLCM)

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\left\{\boldsymbol{X}_N^{adv} \boxed{-} \alpha\,\mathrm{sign}\left(\nabla_X J(\boldsymbol{X}_N^{adv}, \boxed{y_{LL}})\right)\right\}$$

$$y_{LL} = \arg\min_y \{p(y|\boldsymbol{X})\}$$

$$Clip_{X,\epsilon}\{\boldsymbol{X}'\}(x,y,z) = \min\Big\{255, \boldsymbol{X}(x,y,z)+\epsilon, \max\big\{0, \boldsymbol{X}(x,y,z)-\epsilon, \boldsymbol{X}'(x,y,z)\big\}\Big\}$$

- When to use ILLCM?

Adversarial Examples in the Physical World (Kurakin et al., 2017)
Adversarial Machine Learning at Scale (Kurakin et al., 2017)

Random FGSM (R+FGSM)

- FGSM variant with a random starting point

$$x^{\text{adv}} = x' + (\varepsilon - \alpha) \cdot \mathtt{sign}\left(\nabla_{x'} J(x', y_{\text{true}})\right), \quad \text{where} \quad x' = x + \alpha \cdot \mathtt{sign}(\mathcal{N}(\mathbf{0}^d, \mathbf{I}^d))$$

- Designed to circumvent a defense method called adversarial training (Goodfellow et al., 2015) in its naive implementation
- We will come back to the motivation later in defense section

Ensemble Adversarial Training: Attacks and Defenses (Tramèr et al., 2018)
Explaining and Harnessing Adversarial Examples (Goodfellow et al., 2015)

This categorization is only based on methods covered in this presentation and does not encompasses every single attack or defense method that exist as of today.

Attack & Defense Methods

Attack strategies:

Whitebox:

- Direct gradient step(s): FGSM, BIM, R+FGSM, DAG
- Gradient-based greedy algorithm: JSMA
- Iterative optimization: L-BFGS, UAP, C&W, Adversarial Eyeglasses, RP_2, EOT
- Parameterized optimization: ATN

Blackbox:

- Decision boundary approximation: SBA
- Evolutionary algorithm: One Pixel Attack
- Finite difference method: ZOO

Defense strategies:

Data augmentation:

- Adversarial training
- Ensemble adversarial training
- PGD adversarial training

Gradient regularization:

- DCN
- Defensive distillation

Input manipulation:

- PixelDefend

Detection methods:

- Adversary detector networks
- Feature squeezing

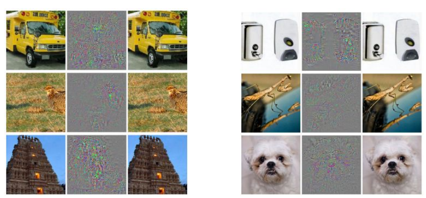
Details will be skipped since background knowledge in optimization (e.g., L-BFGS) is needed.

L-BFGS Attack

- Model adversarial example generation as optimization problem
- Use L-BFGS as optimizer
- Given a victim model ***f***, find ***r*** that minimizes:

$$c|r| + \text{loss}_f(x + r, l)$$
$$\text{subject to } x + r \in [0, 1]^m$$

- Note that the loss function does not have to be the same with the training loss of the victim



Adversarial examples generated using this method (Szegedy et al., 2014).

Intriguing Properties of Neural Networks (Szegedy et al., 2014)

Details will be skipped, this is just to provide an example on how to get creative with existing methods.

Universal Adversarial Perturbation (UAP)

Universal perturbations:

Single perturbation that can be added to multiple inputs to make them adversarial

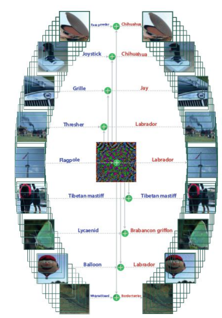


Illustration of UAP (Moosavi-Dezfooli et al., 2016).

Algorithm 1 Computation of universal perturbations.

input: Data points X, classifier $\hat{k}$, desired ℓ_p norm of the perturbation ξ, desired accuracy on perturbed samples δ.
output: Universal perturbation vector v.
Initialize $v \leftarrow 0$.
while $\text{Err}(X_v) \leq 1 - \delta$ **do**
  for each datapoint $x_i \in X$ **do**
    if $\hat{k}(x_i + v) = \hat{k}(x_i)$ **then**
      Compute the *minimal* perturbation that sends $x_i + v$ to the decision boundary:
$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \text{ s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i).$$
      Update the perturbation:
$$v \leftarrow \mathcal{P}_{p,\xi}(v + \Delta v_i).$$
    end if
  end for
end while

$$\mathcal{P}_{p,\xi}(v) = \arg\min_{v'} \|v - v'\|_2 \text{ subject to } \|v'\|_p \leq \xi$$

Universal Adversarial Perturbations (Moosavi-Dezfooli et al., 2016)

This categorization is only based on methods covered in this presentation and does not encompasses every single attack or defense method that exist as of today.

Attack & Defense Methods

Attack strategies:

Whitebox:

- Direct gradient step(s): FGSM, BIM, R+FGSM, DAG
- Gradient-based greedy algorithm: JSMA
- Iterative optimization: L-BFGS, C&W, UAP, Adversarial Eyeglasses, RP_2, EOT
- Parameterized optimization: ATN

Blackbox:

- Decision boundary approximation: SBA
- Evolutionary algorithm: One Pixel Attack
- Finite difference method: ZOO

Defense strategies:

Data augmentation:

- Adversarial training
- Ensemble adversarial training
- PGD adversarial training

Gradient regularization:

- DCN
- Defensive distillation

Input manipulation:

- PixelDefend

Detection methods:

- Adversary detector networks
- Feature squeezing

Adversarial Transformation Network (ATN)

- Train a neural network to generate adversaries
- 2 variants: Adversarial Autoencoder (AAE) and Perturbation ATN (P-ATN)
- AAE: Given a victim model ***f***, train a generator network G_t on dataset ***X*** to output adversarial examples ***X'*** such that ***f(X') = t***, where ***t*** is a target misclassification class
- Every G_t can only be used generate adversarial examples that are misclassified as class ***t***

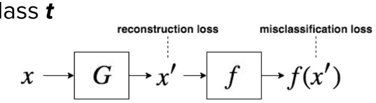
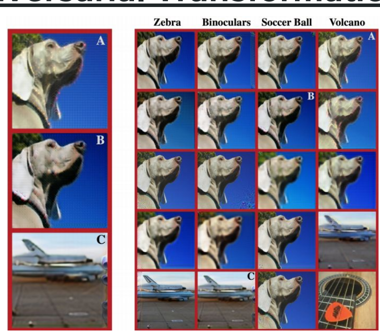


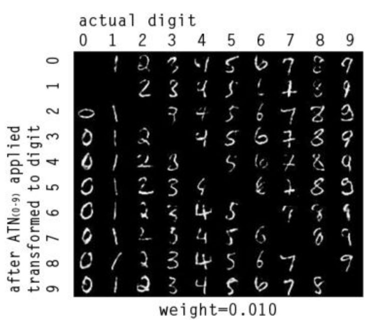
Illustration of AAE.

Adversarial Transformation Networks: Learning to Generate Adversarial Examples (Baluja & Fischer, 2017)

Adversarial Transformation Network (ATN)



Generated adversaries that fool Inception Resnet V2 (Baluja & Fischer, 2017)



Generated adversarial MNIST (Baluja & Fischer, 2017)

Adversarial Transformation Networks: Learning to Generate Adversarial Examples (Baluja & Fischer, 2017)

This categorization is only based on methods covered in this presentation and does not encompasses every single attack or defense method that exist as of today.

Attack & Defense Methods

Attack strategies:

Whitebox:

- Direct gradient step(s): FGSM, BIM, R+FGSM, DAG
- Gradient-based greedy algorithm: JSMA
- Iterative optimization: L-BFGS, C&W, UAP, Adversarial Eyeglasses, RP_2, EOT
- Parameterized optimization: ATN

Blackbox:

- Decision boundary approximation: SBA
- Evolutionary algorithm: One Pixel Attack
- Finite difference method: ZOO

Defense strategies:

Data augmentation:

- Adversarial training
- Ensemble adversarial training
- PGD adversarial training

Gradient regularization:

- DCN
- Defensive distillation

Input manipulation:

- PixelDefend

Detection methods:

- Adversary detector networks
- Feature squeezing

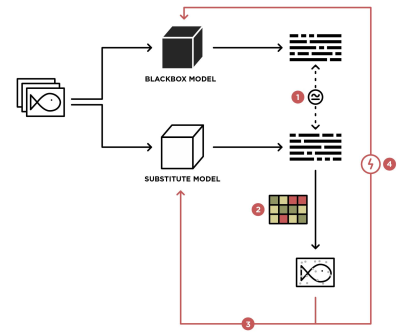
Substitute Blackbox Attack (SBA)

- Blackbox attack by approximating decision boundaries of victim
- Assumption: attacker has access to the softmax probability
- Procedures (see next slide):
  - Train a substitute model on a dataset **labelled by the victim**
  - Attack the substitute model using any whitebox methods
  - The generated adversaries should be transferable to the victim model (thanks to transferability property)
- Successfully attacked Google, Amazon, and MetaMind image recognition models

Practical Black-Box Attacks against Machine Learning (Papernot et al., 2016)

Substitute Blackbox Attack (SBA)

1. Train a substitute model on a dataset **labelled by the victim (i.e., blackbox model)**
2. Attack the substitute model using any whitebox methods
3. Validate that the adversarial examples fool the substitute model
4. Use the adversarial examples to fool the blackbox model



Source: https://medium.com/element-ai-research-lab/tricking-a-machine-into-thinking-youre-milla-jovovich-b19bf322d55c

This categorization is only based on methods covered in this presentation and does not encompasses every single attack or defense method that exist as of today.

Attack & Defense Methods

Attack strategies:

Whitebox:
- Direct gradient step(s): FGSM, BIM, R+FGSM, DAG
- Gradient-based greedy algorithm: JSMA
- Iterative optimization: L-BFGS, C&W, UAP, Adversarial Eyeglasses, RP_2, EOT
- Parameterized optimization: ATN

Blackbox:
- Decision boundary approximation: SBA
- Evolutionary algorithm: One Pixel Attack
- Finite difference method: ZOO

Defense strategies:

Data augmentation:
- Adversarial training
- Ensemble adversarial training
- PGD adversarial training

Gradient regularization:
- DCN
- Defensive distillation

Input manipulation:
- PixelDefend

Detection methods:
- Adversary detector networks
- Feature squeezing

Details will be skipped since knowledge in optimization is needed (e.g., ADAM and coordinate descent). This is just to provide an example on how to get creative with existing methods.

Zeroth Order Optimization (ZOO)

- Blackbox attack via **finite difference approximation**

$$\hat{g}_i := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x} + h\mathbf{e}_i) - f(\mathbf{x} - h\mathbf{e}_i)}{2h}$$

- ***f*** is victim model, $\boldsymbol{e}_i$ is a vector where only the ***i***-th element is 1, $\boldsymbol{x}_i$ is the ***i***-th element of input ***x***
- What is the main disadvantage of using finite difference?

Algorithm 2 ZOO-ADAM: Zeroth Order Stochastic Coordinate Descent with Coordinate-wise ADAM

Require: Step size η, ADAM states $M \in \mathbb{R}^p, v \in \mathbb{R}^p, T \in \mathbb{Z}^p$, ADAM hyper-parameters $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$

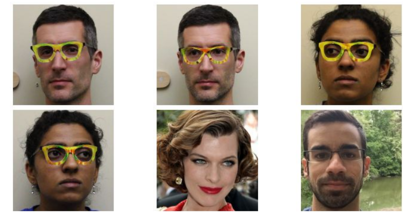
```
M ← 0, v ← 0, T ← 0
while not converged do
    Randomly pick a coordinate i ∈ {1, ··· , p}
    Estimate ĝ_i using (6)
    T_i ← T_i + 1
    M_i ← β_1 M_i + (1 − β_1) ĝ_i,    v_i ← β_2 v_i + (1 − β_2) ĝ_i^2
    M̂_i = M_i / (1 − β_1^{T_i}),    v̂_i = v_i / (1 − β_2^{T_i})
    δ* = −η M̂_i / (√v̂_i + ε)
    Update x_i ← x_i + δ*
end while
```

ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models (Chen et al., 2017)

Some real world examples

Details will be skipped, this is just to provide an example on how to get creative with existing methods.

Adversarial Eyeglasses



Example of adversarial eyeglasses (Sharif et al., 2016). Top row depicts an input given to a model, while bottom row depicts the person from the target misclassification class.

Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition (Sharif et al., 2016)

Details will be skipped, this is just to provide an example on how to get creative with existing methods.

Adversarial Road Signs

- Similar attack method to the adversarial eyeglasses with different masks
- Perceptually different, but **inconspicuous**

“STOP” signs misclassified as a “speed limit” sign (Eykholt et al., 2017).

Robust Physical-World Attacks on Deep Learning Models (Eykholt et al., 2017)

Details will be skipped, this is just to provide an example on how to get creative with existing methods.

Adversarial Turtle

3D printed adversarial turtle misclassified as rifle (Athalye et al., 2018).

Synthesizing Robust Adversarial Examples (Athalye et al., 2018)

Agenda

- Intro
- Adversarial Attacks
- Adversarial Defenses
- Takeaways

This categorization is only based on methods covered in this presentation and does not encompasses every single attack or defense method that exist as of today.

Attack & Defense Methods

Attack strategies:

Whitebox:
- Direct gradient step(s): FGSM, BIM, R+FGSM, DAG
- Gradient-based greedy algorithm: JSMA
- Iterative optimization: L-BFGS, C&W, UAP, Adversarial Eyeglasses, RP_2, EOT
- Parameterized optimization: ATN

Blackbox:
- Decision boundary approximation: SBA
- Evolutionary algorithm: One Pixel Attack
- Finite difference method: ZOO

Defense strategies:

Data augmentation:
- Adversarial training
- Ensemble adversarial training
- PGD adversarial training

Gradient regularization:
- DCN
- Defensive distillation

Input manipulation:
- PixelDefend

Detection methods:
- Adversary detector networks
- Feature squeezing

Adversarial Training

- Include adversarial examples as part of the training set
- If using FGSM adversaries, training loss becomes:

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1-\alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)))$$

- In other words, new adversarial examples are generated **per training iteration** based on the state of the model at that iteration
- <u>**NOT**</u> the same as the Generative Adversarial Nets (GAN)
- Robust to adversaries included during adversarial training

Note: ***J(·)*** denotes the classification loss (e.g. cross-entropy), $\boldsymbol{\theta}$ denotes the model's parameter, $\boldsymbol{\alpha}$ denotes a constant that weigh the importance on classifying normal versus adversarial examples where $\alpha \in [0, 1]$.

Explaining and Harnessing Adversarial Examples (Goodfellow et al., 2015)

Adversarial Training

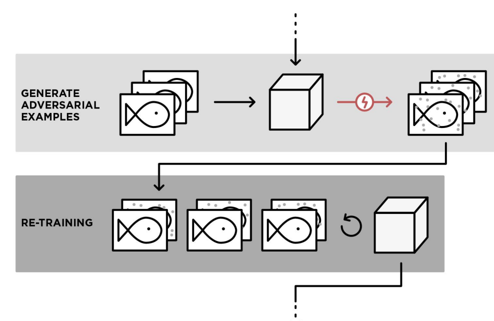


Illustration of adversarial training.

Source: https://medium.com/element-ai-research-lab/securing-machine-learning-models-against-adversarial-attacks-b6cd5d2be8e2

BEWARE: Gradient Masking

- When a defense method prevents a model to reveal meaningful gradients
- At the origin, local gradient towards ϵ_1 is larger compared to ϵ_2 direction
- But loss actually higher in ϵ_2 direction for higher ϵ values
- Many directions orthogonal to ϵ_1 with higher loss at larger ϵ
- Often **unintentional**

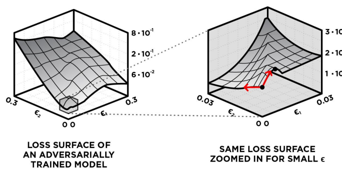


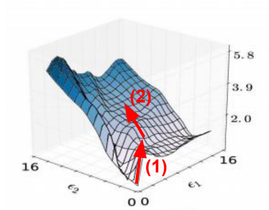
Illustration of loss surface of a model trained with FGSM adversarial training (adapted from Tramèr et al., 2018). Here, ϵ_1 is the direction given by calculating ***dL/dx***, and ϵ_2 is direction orthogonal to ϵ_1.

Ensemble Adversarial Training: Attacks and Defenses (Tramèr et al., 2018)
Source: https://medium.com/element-ai-research-lab/securing-machine-learning-models-against-adversarial-attacks-b6cd5d2be8e2

REVISIT: R+FGSM

$$x^{adv} = x' + (\varepsilon - \alpha) \cdot \overset{(2)}{\mathtt{sign}\left(\nabla_{x'} J(x', y_{true})\right)}, \quad \text{where} \quad x' = x + \alpha \cdot \overset{(1)}{\mathtt{sign}(\mathcal{N}(\mathbf{0}^d, \mathbf{I}^d))}$$

- R+FGSM: add small random perturbation **(1) before** calculating the gradient **(2)** (i.e. random start)
- Can circumvent naive implementation of FGSM adversarial training



(a) Loss of model $v3_{adv}$.

Illustration of how R+FGSM circumvents FGSM adversarial training.

Ensemble Adversarial Training: Attacks and Defenses (Tramèr et al., 2018)

PGD Adversarial Training

- Adversarial training from robust optimization perspective

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y)\right]$$

- In other words: find a set of parameter $\boldsymbol{\theta}$ that **minimizes** the loss in the **worst-case** scenario (Minimax formulation)
- Empirically showed that adversaries generated using R+BIM, which they called Projected Gradient Descent (PGD) method, are the worst-case adversaries
- In practice: perform adversarial training **only** on PGD adversaries

Towards Deep Learning Models Resistant to Adversarial Attacks (Madry et al., 2018)

Agenda

Intro
Adversarial Attacks
Adversarial Defenses
Takeaways

This categorization is only based on methods covered in this presentation and does not encompasses every single attack or defense method that exist as of today.

Attack & Defense Methods

Attack strategies:

Whitebox:
- Direct gradient step(s): FGSM, BIM, R+FGSM, DAG
- Gradient-based greedy algorithm: JSMA
- Iterative optimization: L-BFGS, UAP, C&W, Adversarial Eyeglasses, RP_2, EOT
- Parameterized optimization: ATN

Blackbox:
- Decision boundary approximation: SBA
- Evolutionary algorithm: One Pixel Attack
- Finite difference method: ZOO

Defense strategies:

Data augmentation:
- Adversarial training
- Ensemble adversarial training
- PGD adversarial training

Gradient regularization:
- DCN
- Defensive distillation

Input manipulation:
- PixelDefend

Detection methods:
- Adversary detector networks
- Feature squeezing

Recall our goal:

Minimally modify inputs to maximize some loss function

Gradient-based Attack Methods

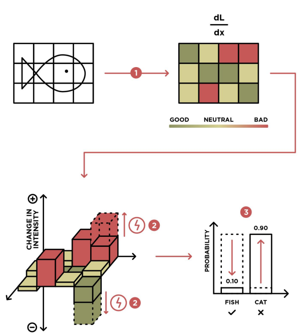


Illustration of gradient-based attacks.

Source: https://medium.com/element-ai-research-lab/tricking-a-machine-into-thinking-youre-milla-jovovich-b19bf322d55c

Takeaway Messages

- Arms race between adversarial attacks and defenses: attackers are winning
- Beware of **gradient masking**, often it is unintentional and may give false robustness
  - 7 out of 9 defenses accepted to ICLR 2018 were successfully attacked just few days after acceptance decision date (Athalye et al., 2018)
- Although most attacks focus on virtual world adversaries, there are works that aim to generate adversarial examples in the physical world (e.g. the adversarial eyeglasses, adversarial turtle, etc.)
- The field is very empirical, need more works that can provide guarantee on adversarial robustness (e.g., by providing upper bound of a proposed defense method)

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples (Athalye et al., 2018)

Thank You. Questions?

My blog on adversarial examples:

1. Tricking a Machine Learning into Thinking You're Milla Jovovich (https://medium.com/element-ai-research-lab/tricking-a-machine-into-thinking-youre-milla-jovovich-b19bf322d55c)

2. Securing Machine Learning Models Against Adversarial Attacks (https://medium.com/element-ai-research-lab/securing-machine-learning-models-against-adversarial-attacks-b6cd5d2be8e2)