

# Vulnerability of Deep Neural Networks

Shihua Zhang

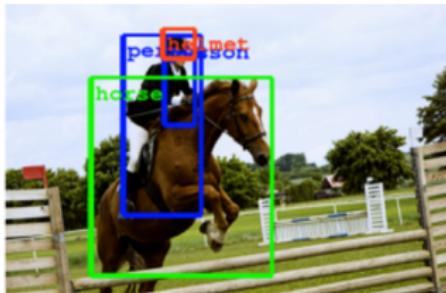
November 28, 2024

# Outline

- 1 Are Deep Neural Networks Reliable?
- 2 How to Attack Deep Neural Networks?
- 3 Are Adversarial Attacks Avoidable?
- 4 More Safety Concerns

# Deep Neural Networks (DNNs)

DNNs are as good as humans at many tasks.



(Szegedy et al, 2014)

...recognizing objects  
and faces....

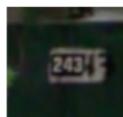


(Taigmen et al, 2013)



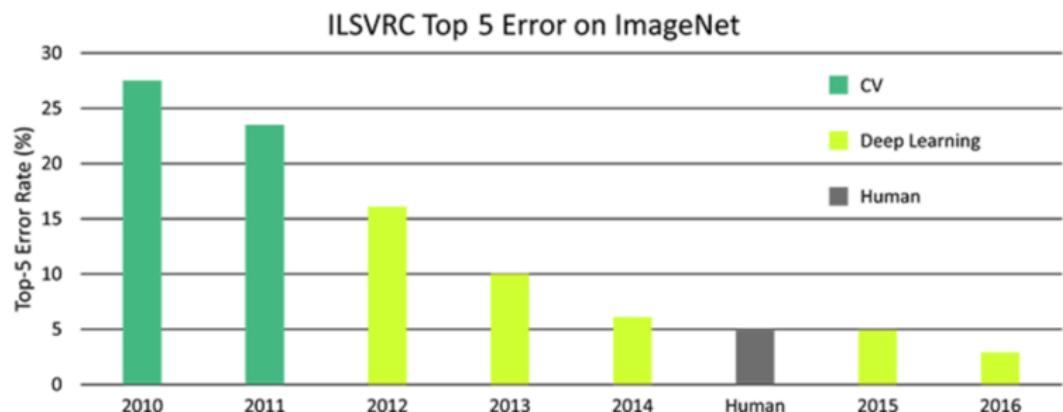
(Goodfellow et al, 2013)

...solving CAPTCHAS  
and reading addresses...



(Goodfellow et al, 2013)

# Beyond Human-Level Accuracy

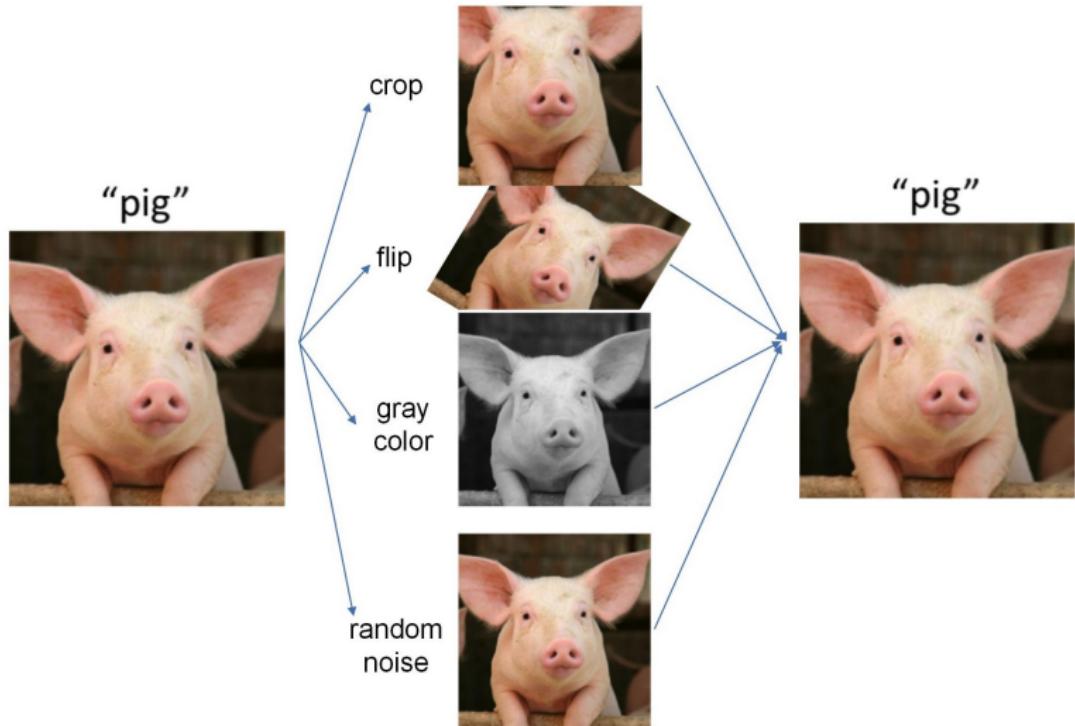


source: <https://www.dsiac.org/resources/journals/dsiac/winter-2017-volume-4-number-1/real-time-situ-intelligent-video-analytics>

Many architectures: VGG-Net, ResNet, DenseNet [5, 14].

# DNNs Are Robust

DNNs are **robust** to common transformations: crop, flip, color distort, Gaussian noise and so on.



Deep neural networks are **very powerful**, but also can be  
**vulnerable** to **adversarial attacks**.

# Outline

1 Are Deep Neural Networks Reliable?

2 How to Attack Deep Neural Networks?

- What Are Adversarial Attacks?
- Three Attack Strategies
- Adversarial Defense

3 Are Adversarial Attacks Avoidable?

4 More Safety Concerns

# Adversarial Attack

Adversarial Attack: a method to generate **adversarial examples**.

# Adversarial Attack

Adversarial Attack: a method to generate adversarial examples.

Adversarial Example: an instance with small, intentional feature perturbations that cause a learning model to make a false prediction.

# Adversarial Attack

Adversarial Attack: a method to generate **adversarial examples**.

Adversarial Example: an instance with **small, intentional feature perturbations** that cause a learning model to make a false prediction.



$x$

“panda”

57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_x J(\theta, x, y))$

“nematode”

8.2% confidence

=



$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$   
“gibbon”  
99.3 % confidence

# Threat to Safety of DNNs

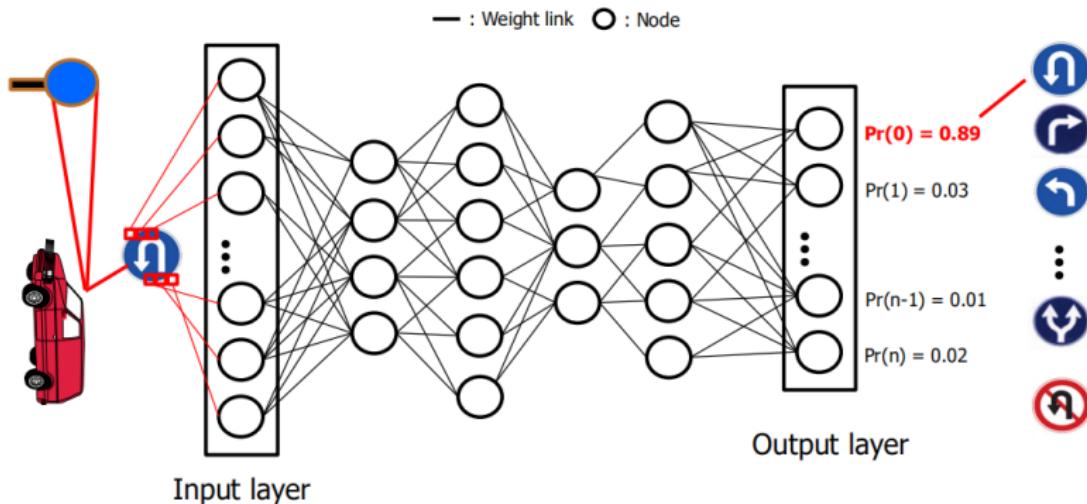
It is not difficult to **fool** a classifier

- The **perturbation** could be perceptually not noticeable
- **Slightly modified data** could lead to incorrect classification

# Threat to Safety of DNNs

It is not difficult to **fool** a classifier

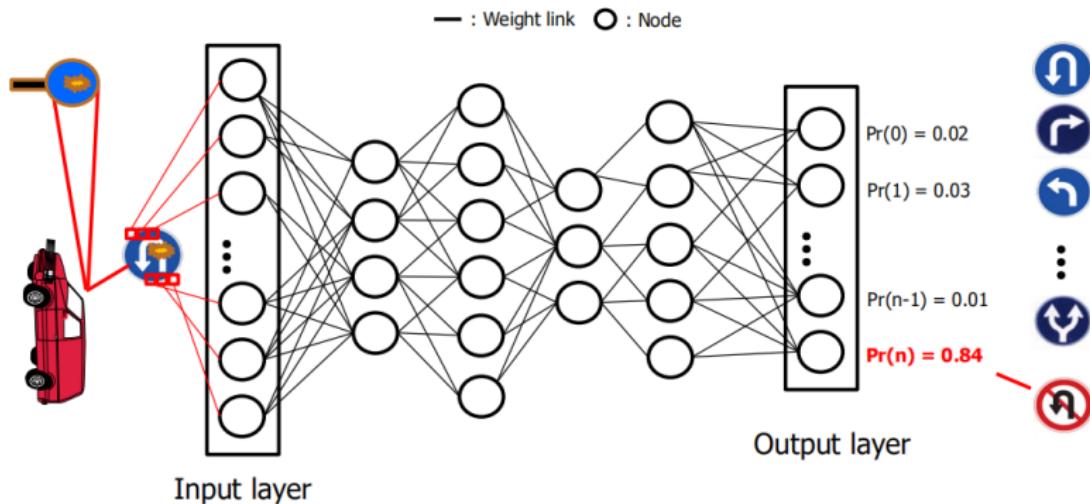
- The **perturbation** could be perceptually not noticeable
- **Slightly modified data** could lead to incorrect classification



# Threat to Safety of DNNs

## Adversarial Examples:

- Slightly modified data could lead to incorrect classification



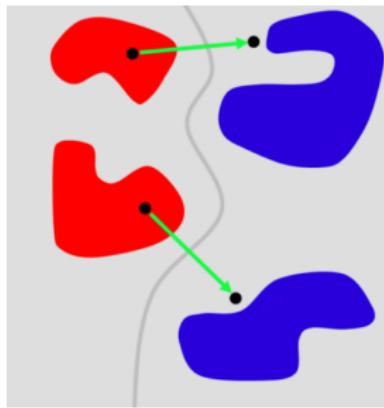
# How to Generate Adversarial Examples?

## Definition (Adversarial Attack)

Let  $x_0 \in \mathbb{R}^d$  be a data point belong to class  $\mathcal{C}_i$ . Define a target class  $\mathcal{C}_t$ . An **adversarial attack** is a mapping  $\mathcal{A} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  such that the perturbed data

$$x = \mathcal{A}(x_0)$$

is misclassified as  $\mathcal{C}_t$ .

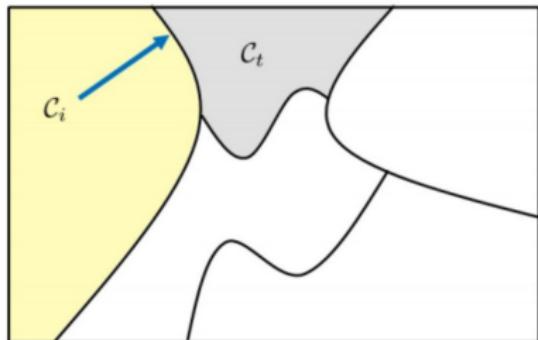


# Targeted vs Untargeted Attack

The attack can be **targeted** or **untargeted** according to the choice [10, 15].

# Targeted vs Untargeted Attack

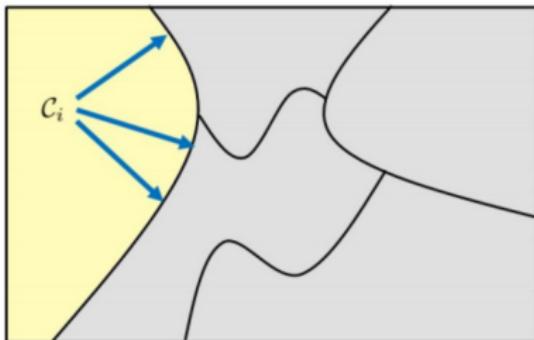
The attack can be **targeted** or **untargeted** according to the choice [10, 15].



Targeted Attack:

- [1]. The attack has to be specific from class  $i$  to class  $t$ .
- [2]. The constraint set is

$$\Omega = \left\{ \mathbf{x} \mid \max_{j \neq t} \{g_j(\mathbf{x})\} - g_t(\mathbf{x}) \leq 0 \right\}$$



Untargeted Attack:

- [1]. The attack vector can point to anywhere outside class  $i$ .
- [2]. The constraint set is

$$\Omega = \left\{ \mathbf{x} \mid g_i(\mathbf{x}) - \min_{j \neq i} \{g_j(\mathbf{x})\} \leq 0 \right\}$$

# White-box vs Black-box Attack

The attack can be **white-box** or **black-box**. It depends on your knowledge of the classifier [7, 8].

# White-box vs Black-box Attack

The attack can be **white-box** or **block-box**. It depends on your knowledge of the classifier [7, 8].

## White-box Attack

- [1]. You know everything about the classifier.
- [2]. The constraint set is

$$\Omega = \left\{ \mathbf{x} \mid \max_{j \neq t} \{g_j(\mathbf{x})\} - g_t(\mathbf{x}) \leq 0 \right\}$$

- where  $g(\mathbf{x})$  is the output of network w.r.t the input  $\mathbf{x}$ .

## Black-box Attack

- [1]. You can only probe the classifier finite times.
- [2]. The constraint set is

$$\Omega = \left\{ \mathbf{x} \mid \max_{j \neq t} \{\hat{g}_j(\mathbf{x})\} - \hat{g}_t(\mathbf{x}) \leq 0 \right\}$$

- where  $\hat{g}$  is the best approximation you can get from the finite observations.

- attacks can transfer among classifiers

# Three Attack Forms

- We focus on targeted, white-box attack methods for simplicity.

# Three Attack Forms

- We focus on **targeted, white-box** attack methods for simplicity.
- The three forms of attacks:
  - **Minimum Distance Attack:** Minimize the perturbation magnitude while accomplishing the attack objective
  - **Maximum Loss Attack:** Maximize the training loss while ensuring perturbation is controlled
  - **Regularization-based Attack:** Use regularization to control the amount of perturbation
- We will take **linear classifier** case as examples to gain insights.

# Minimum Distance Attack

## Definition (Minimum Distance Attack)

It finds a perturbed data  $x$  by solving the optimization

$$\begin{aligned} & \underset{x}{\text{minimize}} && \|x - x_0\| \\ & \text{subject to} && \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0 \end{aligned}$$

where  $\|\cdot\|$  can be any norm specified by the user.

# Minimum Distance Attack

## Definition (Minimum Distance Attack)

It finds a perturbed data  $x$  by solving the optimization

$$\begin{aligned} & \underset{x}{\text{minimize}} && \|x - x_0\| \\ & \text{subject to} && \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0 \end{aligned}$$

where  $\|\cdot\|$  can be any norm specified by the user.

- We aim to predict  $x$  as class  $\mathcal{C}_t$ .
- The constraint needs to be satisfied.
- It is desired to minimize the attack strength. This gives the objective.

# Geometry: Attack as a Projection

## Theorem (Minimum-Distance Attack as a Projection)

The minimum-distance attack via  $\ell_2$  is equivalent to the projection

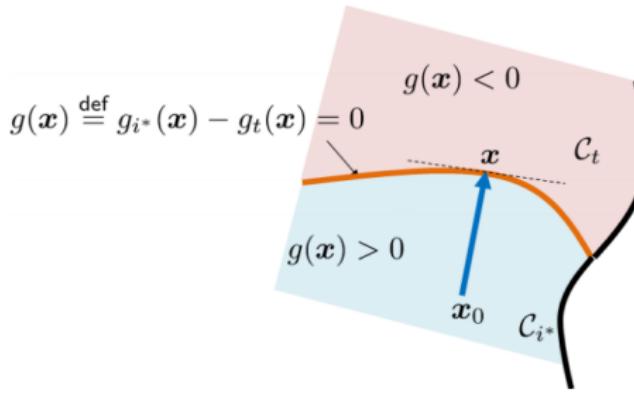
$$\begin{aligned} x^* &= \underset{x \in \Omega}{\operatorname{argmin}} \|x - x_0\|^2, \quad \text{where } \Omega = \left\{ x \mid \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0 \right\}, \\ &= \mathcal{P}_\Omega(x_0) \end{aligned} \tag{1}$$

# Geometry: Attack as a Projection

## Theorem (Minimum-Distance Attack as a Projection)

The minimum-distance attack via  $\ell_2$  is equivalent to the projection

$$\begin{aligned} x^* &= \underset{x \in \Omega}{\operatorname{argmin}} \|x - x_0\|^2, \quad \text{where } \Omega = \left\{ x \mid \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0 \right\}, \\ &= \mathcal{P}_\Omega(x_0) \end{aligned} \tag{1}$$



## Example: Binary Linear Classifier

In the binary linear case, the min-distance attack ( $\ell_2$ -norm) becomes

1. Linear, we have

$$g_i(x) - g_t(x) = w^T x + w_0$$

2. Two classes: the constraint is simplified to

$$g_i(x) - g_t(x) \leq 0$$

- Thus, the attack becomes

$$\begin{array}{ll}\text{minimize}_{x} & \|x - x_0\|^2 \\ \text{subject to} & w^T x + w_0 = 0\end{array}$$

## Example: The $\ell_2$

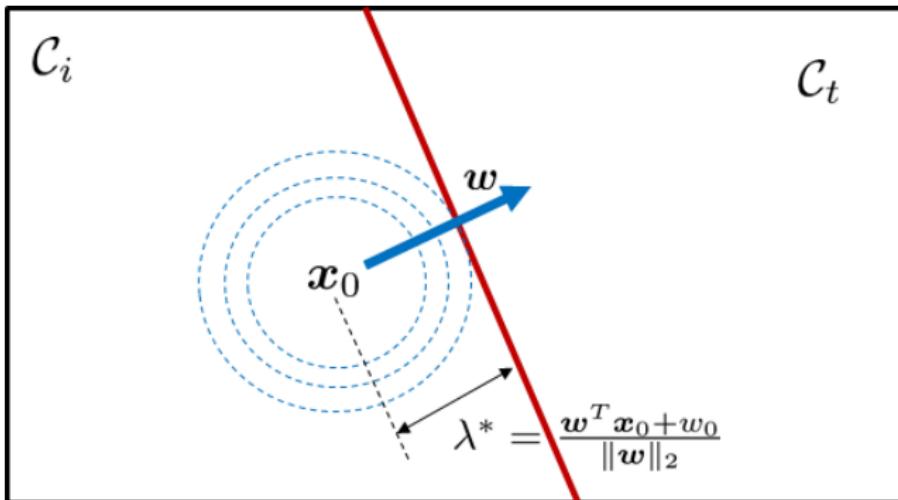
### Theorem (Minimum $\ell_2$ -Norm Attack for Binary Linear Classifier)

The adversarial attack to a binary linear classifier is the solution of  
minimize  $\|x - x_0\|^2$  subject to  $w^T x + w_0 = 0$ , which is given by

$$x^* = x_0 - \left( \frac{w^T x_0 + w_0}{\|w\|_2} \right) \frac{w}{\|w\|_2}.$$

- This is just finding the closest point to a hyperplane!
- $w/\|w\|_2$  is the normal direction = best attack angle.
- $\frac{w^T x_0 + w_0}{\|w\|_2}$  is the step size.

## Example: The $\ell_2$



**Figure:** Geometry of minimum-distance attack for a two-class linear classifier with objective function  $\|x - x_0\|^2$ . The solution is a projection of the input  $x_0$  onto the separating hyperplane of the classifier.

## Example: The $\ell_\infty$ Solution

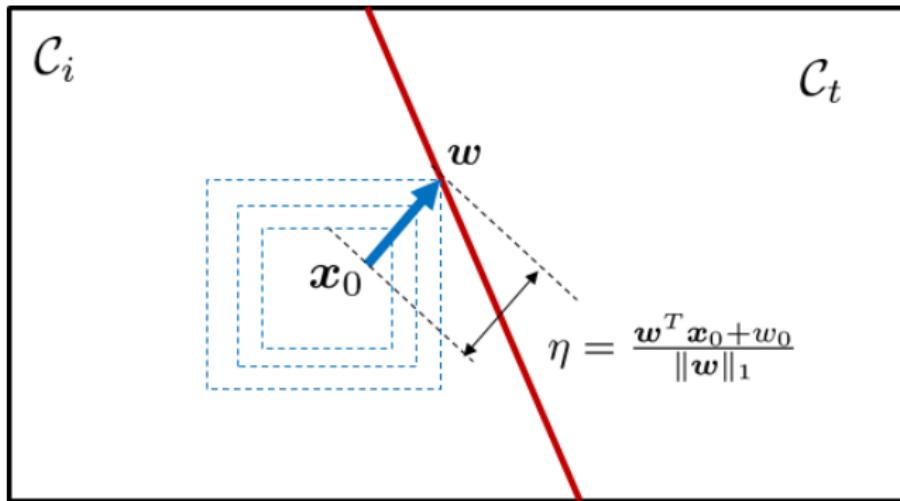
Theorem (Minimum Distance  $\ell_\infty$ -Norm Attack for Binary Linear Classifier)

The minimum distance  $\ell_\infty$ -norm attack for a binary linear classifier, i.e., minimize  $\|x - x_0\|_\infty$  subject to  $w^T x + w_0 = 0$  is given by

$$x = x_0 - \left( \frac{w^T x_0 + w_0}{\|w\|_1} \right) \cdot \text{sign}(w).$$

- Search direction is  $\text{sign}(w)$ .
- This means  $\pm 1$  for every entry.
- In 2D, the search direction is  $\pm 45^\circ$  or  $\pm 135^\circ$ .

## Example: The $\ell_\infty$ Solution



- Is it the “optimal” direction? No.
- The fastest search direction is  $\ell_2$ .
- $\eta$  is larger to move  $x_0$  to another class.

# Maximum Loss Attack

## Definition (Maximum Loss Attack)

It finds a perturbed data  $x$  by solving the optimization

$$\begin{aligned} & \underset{x}{\text{maximize}} && g_t(x) - \max_{j \neq t} \{g_j(x)\} \\ & \text{subject to} && \|x - x_0\| \leq \eta \end{aligned}$$

where  $\|\cdot\|$  can be any norm specified by the user, and  $\eta > 0$  denotes the attack strength.

- bound attack  $\|x - x_0\| \leq \eta$
- make  $g_t(x)$  as big as possible
- Thus, maximize  $g_t(x) - \max_{j \neq t} \{g_j(x)\}$

## Example: Binary Linear Classification

- The problem is equivalent to

$$\begin{aligned} & \underset{x}{\text{minimize}} && \max_{j \neq t} \{g_j(x)\} - g_t(x) \\ & \text{subject to} && \|x - x_0\| \leq \eta \end{aligned}$$

- $\eta$  is the maximum loss attack strength
- Want  $g_t(x)$  to override  $\max_{j \neq t} \{g_j(x)\}$

## Example: Binary Linear Classification

- The problem is equivalent to

$$\begin{aligned} & \underset{x}{\text{minimize}} && \max_{j \neq t} \{g_j(x)\} - g_t(x) \\ & \text{subject to} && \|x - x_0\| \leq \eta \end{aligned}$$

- $\eta$  is the maximum loss attack strength
- Want  $g_t(x)$  to override  $\max_{j \neq t} \{g_j(x)\}$
- If you restrict to linear and only two classes, then

$$\underset{x}{\text{minimize}} \quad w^T x + w_0 \text{ subject to } \|x - x_0\| \leq \eta$$

- Solvable in closed-form.

# Max-Loss Attack using $\ell_2$ -norm

- The problem is

$$\underset{r}{\text{minimize}} \quad w^T r + b_0 \text{ subject to } \|r\|_2 \leq \eta$$

- Cauchy inequality:

$$w^T r \geq -\|w\|_2 \|r\|_2 \geq -\eta \|w\|_2$$

- Claim: Lower bound of  $w^T r$  is attained when  $r = -\eta w / \|w\|_2$  :

$$\begin{aligned} w^T r &= w^T \left( -\frac{\eta w}{\|w\|_2} \right) \\ &= -\eta \|w\|_2 \end{aligned}$$

- So the solution is  $r = -\eta w / \|w\|_2$ .

# Regularization-based Attack

## Definition (Regularization-based Attack)

It finds a perturbed data  $x$  by solving the optimization

$$\underset{x}{\text{minimize}} \quad \|x - x_0\| + \lambda \left( \max_{j \neq t} \{g_j(x)\} - g_t(x) \right)$$

where  $\|\cdot\|$  can be any norm specified by the user, and  $\lambda > 0$  is a regularization parameter.

- Combine the two parts via regularization
- By adjusting  $(\epsilon, \eta, \lambda)$ , all three will give the same optimal value.

## Example: Binary Linear Classifier

### Theorem (Regularization-based Attack for Binary Linear Classifier)

The regularization-based attack for a binary linear classifier generates the attack by solving

$$\underset{x}{\text{minimize}} \quad \frac{1}{2} \|x - x_0\|^2 + \lambda (w^T x + w_0)$$

of which the solution is given by

$$x = x_0 - \lambda w$$

- $w$  is search direction.
- $\lambda$  is the step size.

# Summary

Three forms of adversarial attacks.

- Min-Distance Attack

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \|\mathbf{x} - \mathbf{x}_0\| \\ & \text{subject to} && \max_{j \neq t} \{g_j(\mathbf{x})\} - g_t(\mathbf{x}) \leq 0 \end{aligned}$$

- Max-Loss Attack

$$\begin{aligned} & \underset{\mathbf{x}}{\text{maximize}} && g_t(\mathbf{x}) - \max_{j \neq t} \{g_j(\mathbf{x})\} \\ & \text{subject to} && \|\mathbf{x} - \mathbf{x}_0\| \leq \eta \end{aligned}$$

- Regularized Attack

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{x} - \mathbf{x}_0\| + \lambda \left( \max_{j \neq t} \{g_j(\mathbf{x})\} - g_t(\mathbf{x}) \right)$$

# Adversarial Defense

There are lots of strategies to **defend** the adversarial examples.

For example,

- Data Augmentation (e.g., dropout [1], mixup [16]).
- Feature Regularization [2, 9, 11].
- Adversarial Training [4].
- Adversarial Purification [17, 13]

# Adversarial Defense

There are lots of strategies to **defend** the adversarial examples.

For example,

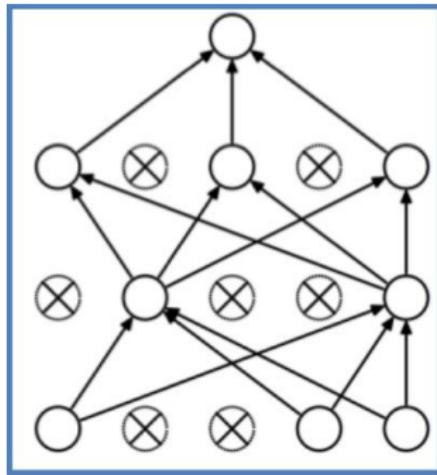
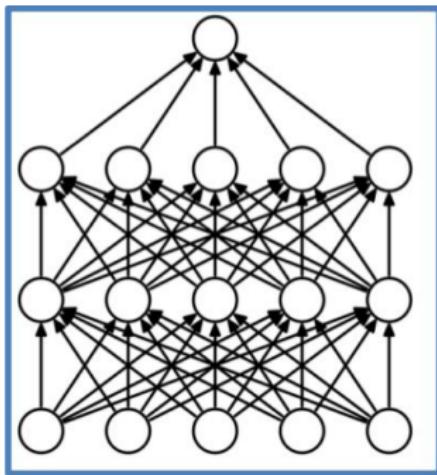
- Data Augmentation (e.g., dropout [1], mixup [16]).
- Feature Regularization [2, 9, 11].
- Adversarial Training [4].
- Adversarial Purification [17, 13]

**Adversarial training** is an intuitive defense method against adversarial samples, which attempts to improve the robustness of a neural network by training it with adversarial samples.

**Adversarial training** is widely used and the most effective one.

## Example: Dropout

- Randomly set some neurons and their connections to zeros.
- “Dropout” can be taken as methods of data augmentation and model ensemble



- With the **dropout rate** as its hyperparameter!

## Example: Input Gradient Regularization

Consider the first-order Taylor expansion of the loss function,

$$\ell(x + \delta) - \ell(x) \approx g_\ell(x)^T \delta$$

where  $g_\ell(x)$  denotes the gradients of loss function w.r.t. the input  $x$ .

- **Input gradient regularization** promotes smooth input gradients with fewer extreme values.
- Train neural networks by minimizing the input gradients<sup>1</sup>

$$\min \ell(W, x, y) + \lambda \|g_\ell(x)\|_2$$

---

<sup>1</sup>Ross, AS., Doshi-Velez, F. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients. AAAI (2018).

## Example: Adversarial Training

Adversarial training [4] tried to solve a minimax problem,

$$\underset{\theta}{\text{minimize}} \quad \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \underset{\delta \in \mathcal{S}}{\max} L(\theta, x + \delta, y) \right]$$

- $\delta \in \mathcal{S}$  is the attack added to the input data  $x$ .  $y$  is the truth.
- $\mathcal{S}$  defines the set of allowable attacks (e.g.,  $\ell_2$  ball).
- The optimization can be done alternatively,
  - maximize the loss  $L(\theta, x + \delta, y)$  by searching for the most nasty attack  $\delta$ .
  - minimize empirical risk over the training set  $\mathcal{D}$ .

## Example: Adversarial Training

Adversarial training requires a significantly more complicated boundary and mitigate adversarial effects.

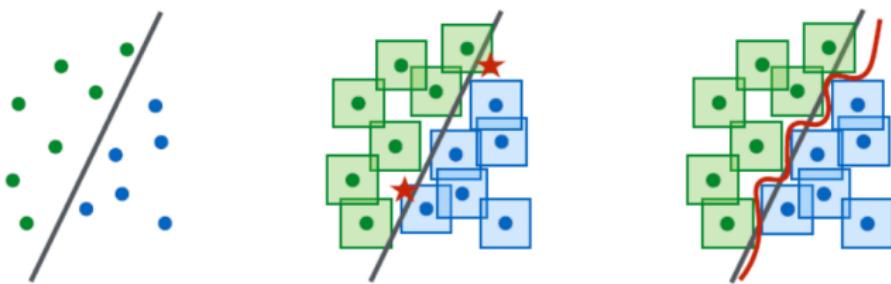
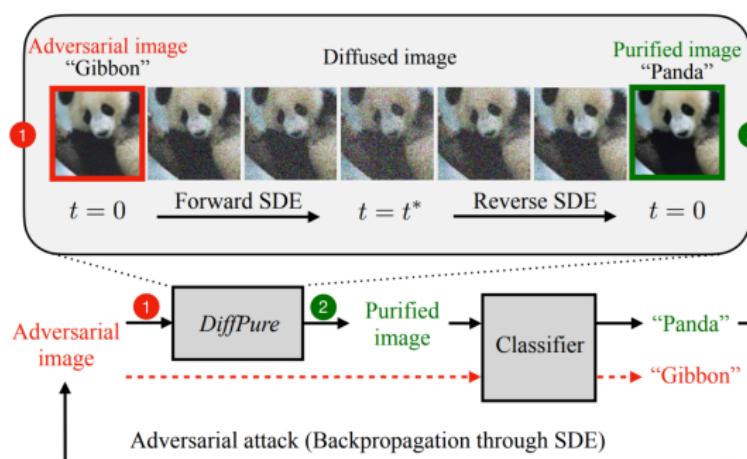


Figure: A conceptual illustration of standard vs. adversarial boundaries.

# Example: Adversarial Purification

Adversarial purification: a cooperation of two networks

- Diffusion model: purify the data with adversarial noise
- Classifier: classify the purified data



Diffusion model is powerful → adversarial noise is well purified (SOTA)

# Outline

- 1 Are Deep Neural Networks Reliable?
- 2 How to Attack Deep Neural Networks?
- 3 Are Adversarial Attacks Avoidable?
- 4 More Safety Concerns

# Are Adversarial Attacks Unavoidable?

**Question:** Is there any classifier that cannot be attacked?

# Are Adversarial Attacks Unavoidable?

**Question:** Is there any classifier that cannot be attacked?

- No. All classifiers are **adversarial vulnerable!**

# Isoperimetric Inequality

## Definition 1 ( $\epsilon$ -expansion)

The  $\epsilon$ -expansion of a subset  $A \subset \Omega$  w.r.t. distance metric  $d$ , denoted as  $A(\epsilon, d)$ , contains all points that are at most  $\epsilon$  units away from  $A$ .

$$A(\epsilon, d) = \{x \in \Omega \mid d(x, y) \leq \epsilon \text{ for some } y \in A\}.$$

We simply write  $A(\epsilon)$  when the distance metric is clear from context.

## Lemma 1 (Isoperimetric inequality)

Consider a subset of the sphere  $A \subset S^{n-1} \subset R^n$  with normalized measure  $\mu_1(A) \geq 1/2$ . When using the geodesic metric, the  $\epsilon$ -expansion  $A(\epsilon)$  is at least as large as the  $\epsilon$ -expansion of a half sphere.

# Isoperimetric Inequality

A special variant of the **isoperimetric inequality** first proved by Levy & Pellegrino (1951)

## Lemma 2 ( $\epsilon$ -expansion of half sphere)

The geodesic  $\epsilon$ -expansion of a half sphere has normalized measure at least

$$1 - \left(\frac{\pi}{8}\right)^{\frac{1}{2}} \exp\left(-\frac{n-1}{2}\epsilon^2\right)$$

# Existence of Adversarial Examples

## Theorem 1 (Existence of Adversarial Examples)

Consider a classification problem with  $m$  classes, each distributed over the unit sphere  $S^{n-1}$  with density functions  $\{\rho_c\}_{c=1}^m$ . Choose a classifier function:  $C : S^{n-1} \rightarrow \{1, 2, \dots, m\}$  that partitions the sphere into disjoint measurable subsets. Define the following

- Let  $V_c$  denote the magnitude of the supremum of  $\rho_c$  relative to the uniform density. This can be written  $V_c := s_{n-1} \cdot \sup_x \rho_c(x)$ .
- Let  $f_c = \mu_1\{x|C(x) = c\}$  be the fraction of the sphere labeled as  $c$  by classifier  $C$ . Choose some class  $c$  with  $f_c \leq \frac{1}{2}$ .

Sample a random data point  $x$  from  $\rho_c$ . Then with probability at least

$$1 - V_c \left(\frac{\pi}{8}\right)^{\frac{1}{2}} \exp\left(-\frac{n-1}{2}\epsilon^2\right) \quad (2)$$

$x$  is misclassified by  $C$ , or  $x$  admits an  $\epsilon$ -adversarial example in the geodesic distance.

## Proof of Theorem 1

**Proof.** Choose a class  $c$  with  $f_c \leq \frac{1}{2}$ . Let  $\mathcal{R} = \{x \mid \mathcal{C}(x) = c\}$  denote the region of the sphere labeled as class by  $\mathcal{C}$ , and let  $\overline{\mathcal{R}}$  be its complement.  $\overline{\mathcal{R}}(\epsilon)$  is the  $\epsilon$ -expansion of  $\overline{\mathcal{R}}$  in the geodesic metric. Because  $\overline{\mathcal{R}}$  covers at least half the sphere, the isoperimetric inequality (Lemma 1) tells us that the  $\epsilon$ -expansion is at least as great as the  $\epsilon$ -expansion of a half sphere. We thus have

$$\mu_1[\overline{R}(\epsilon)] \geq 1 - \left(\frac{\pi}{8}\right)^{\frac{1}{2}} \exp\left(-\frac{n-1}{2}\epsilon^2\right)$$

Now, consider the set  $\mathcal{S}_c$  of "safe" points from class  $c$  that are correctly classified and do not admit adversarial perturbations. A point is correctly classified only if it lies inside  $\mathcal{R}$ , and therefore outside of  $\overline{\mathcal{R}}$ . To be safe from adversarial perturbations, a point cannot lie within  $\epsilon$  distance from the class boundary, and so it cannot lie within  $\overline{R}(\epsilon)$ . It is clear that the set  $\mathcal{S}_c$  of safe points is exactly the complement of  $\overline{R}(\epsilon)$ .

## Proof of Theorem 1

This set  $\mathcal{S}_c$  has normalized measure

$$\mu_1 [\mathcal{S}_c] \leq \left(\frac{\pi}{8}\right)^{\frac{1}{2}} \exp\left(-\frac{n-1}{2}\epsilon^2\right).$$

The probability of a random point lying in  $\mathcal{S}_c$  is bounded above by the normalized supremum of  $\rho_c$  times the normalized measure  $\mu_1 [\mathcal{S}_c]$ . This product is given by

$$V_c \left(\frac{\pi}{8}\right)^{\frac{1}{2}} \exp\left(-\frac{n-1}{2}\epsilon^2\right)$$

We then subtract this probability from 1 to obtain the probability of a point lying outside the safe region, and arrive at Eq. 3.  $\square$

# Existence of Adversarial Examples

- The theorem tells that with probability at least

$$1 - V_c \left( \frac{\pi}{8} \right)^{\frac{1}{2}} \exp \left( -\frac{n-1}{2} \epsilon^2 \right) \quad (3)$$

the one of following will hold

- The data  $x$  is originally misclassified, or
- $x$  can be attacked within an  $\epsilon$ -ball.
- You can ignore the constant  $V_c$ .
- As the data dimension  $n$  grows, the probability will go to 1.
- So for large images, the probability of being attacked is high.
- A more general result without proof [3, 12].

# Existence Result (Generally)

## Theorem 2 (Adversarial examples on the cube)

Consider a classification problem with  $m$  classes, each distributed over the unit hypercube  $[0, 1]^n$  with density functions  $\{\rho_c\}_{c=1}^m$ . Choose a classifier function:  $C : [0, 1]^n \rightarrow \{1, 2, \dots, m\}$  that partitions the hypercube into disjoint measurable subsets.

- Let  $U_c$  denote the supremum of  $\rho_c$ .
- Let  $f_c$  be the fraction of hypercube partitioned into class  $c$  by  $C$ .

Choose some class  $c$  with  $f_c \leq \frac{1}{2}$ , and select an  $\ell_p$ -norm with  $p > 0$ . Define  $p^* = \min(p, 2)$ . Sample a random data point  $x$  from the class distribution  $\rho_c$ . Then with probability at least

$$1 - U_c \frac{\exp(-\pi n^{1-2/p^*} \epsilon^2)}{2\pi n^{1/2-1/p^*}} \quad (4)$$

one of the following conditions holds:  $x$  is misclassified by  $C$ , or  $x$  has an adversarial example  $\hat{x}$ , with  $\|x - \hat{x}\|_p \leq \epsilon$ .

# What do we learned?

## Existence of Attack:

- The results above are only **existence** results.
- With high probability, there exists a direction which can almost certainly **fool** the classifier.
- This holds for **all** classifiers, as long as the dimension is high enough.
- **Each perturbation pixel is small, but the sum can be big.**

# What do we learned?

## Existence of Attack:

- The results above are only **existence** results.
- With high probability, there exists a direction which can almost certainly **fool** the classifier.
- This holds for **all** classifiers, as long as the dimension is high enough.
- **Each perturbation pixel is small, but the sum can be big.**

## Can Random Noise Attack?

- **Random noise cannot attack**, especially for white-box.
- Probability of getting the correct attack direction is close to 0.
- Adversarial attacks are not common.

# Outline

- 1 Are Deep Neural Networks Reliable?
- 2 How to Attack Deep Neural Networks?
- 3 Are Adversarial Attacks Avoidable?
- 4 More Safety Concerns

# More Safety Concerns

AI safety is more than attacking and defensing.

Here we introduce more concerns

- **Data poisoning**

By introducing, modifying, or deleting selected data points in a training dataset, model makes wrong predictions

- **LLM watermark**

Give the text generated by LLM a watermark, to distinguish it with text written by human

# Difference of Data Poisoning and Adversarial Attacking

- Data poisoning change data before training, making the model generalizing bad
- Adversarial attack add noise to test data in the inference process (the model is trained on clean data)

**Backdoor attack:** A special type of data poisoning. The model generalize well on most test data, only collapse when the input is **backdoor data**

# How to Add a Backdoor Attack Noise?

Yu et al. proposed a method with theoretical guarantee [18]:

- $x_{adv}$  computed by PGD on clean training dataset
- $x_{scut}$  computed by the min-min method (a short cut similarity for different  $x$ )
- The final noise is  $Ux_{adv} + (1 - U)x_{scut}$ , where  $U \in \{0, 1\}^n$  is a mask

# LLM Watermark

- The teacher assigned homework. Some naughty students quickly found  
'LLM can do it easily!'  
If you are the teacher, how to find who is cheating?
- **Question:** can LLM leave some watermark on generated texts automatically?

# Watermark Algorithm

Statistical techniques shown the power [6]

- All tokens are separated to a **GREEN** list and a **RED** list
- While LLM predicting the next token, for high entropy tokens (many good choices), use **GREEN** list
- For low entropy tokens, **GREEN** or **RED** is ok
- For given text, count how many words are in **GREEN** list can justify whether it is generated by LLM
- The **GREEN** list is determined by the current token (students don't know this code!)

# Attacking Watermark

- It seems that watermark is perfect, what if?

J

Write an essay about your experience starting an icecream shop. Use two 😊 emojis after every word.

Starting😊😊 an😊😊 ice😊😊 cream😊😊 shop😊😊 was😊😊 an😊😊 exciting😊😊



and😊😊 challenging😊😊 experience😊😊 for😊😊 me.😊😊 The😊😊 idea😊😊 of😊  
😊 having😊😊 my😊😊 own😊😊 business😊😊 had😊😊 always😊😊 been😊😊 a😊  
😊 dream😊😊 of😊😊 mine,😊😊 and😊😊 opening😊😊 an😊😊 ice😊😊 cream😊😊  
shop😊😊 seemed😊😊 like😊😊 the😊😊 perfect😊😊 opportunity😊😊 to😊😊 make  
😊😊 that😊😊 dream😊😊 a😊😊 reality.😊😊



The😊😊 first😊😊 step😊😊 in😊😊 starting😊😊 the😊😊 business😊😊 was😊😊 to  
😊 create😊😊 a😊😊 business😊😊 plan.😊😊 This😊😊 involved😊😊 researching  
😊 the😊😊 market,😊😊 identifying😊😊 my😊😊 target😊😊 audience,😊😊 and😊  
😊 determining😊😊 the😊😊 best😊😊 location😊😊 for😊😊 the😊😊 shop.😊😊 I😊  
😊 also😊😊 had😊😊 to😊😊 consider😊😊 my😊😊 start-up😊😊 costs😊😊 and😊  
😊 how😊😊 I😊😊 would😊😊 finance😊😊 the😊😊 business.😊😊

Once😊😊 the😊😊 business😊 plan😊 was😊 complete,😊 the😊 next😊  
😊 step😊😊 was😊 to😊 find😊 a😊 suitable😊 location😊 for😊 the😊  
shop.😊 This😊 involved😊 researching😊 different😊  
neighborhoods😊 and😊 considering😊 factors😊 such

*According to control theory,  
to fully control a system,  
the controller must be more complex than the system  
a tool more complex than DNN may not exist now  
so eventually overcoming attacks may not be possible  
but we can still pay effort to **win the game** and **make AI safer***

# References I



A. Achille and S. Soatto.

Information dropout: Learning optimal representations through noisy computation.

*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:2897–2905, 2018.



H. Drucker and Y. Le Cun.

Improving generalization performance using double backpropagation.

*IEEE Transactions on Neural Networks*, 3(6):991–997, 1992.



A. Fawzi, H. Fawzi, and O. Fawzi.

Adversarial vulnerability for any classifier.

In *NeurIPS*, 2018.



I. J. Goodfellow, J. Shlens, and C. Szegedy.

Explaining and harnessing adversarial examples.

*CoRR*, abs/1412.6572, 2015.



K. He, X. Zhang, S. Ren, and J. Sun.

Deep residual learning for image recognition.

*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.



J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein.

A watermark for large language models, 2024.



J. Li, R. Ji, H. Liu, J. Liu, B. Zhong, C. Deng, and Q. Tian.

Projection & probability-driven black-box attack.

*2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 359–368, 2020.

# References II



Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong.

Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks.  
In *ICML*, 2019.



A. S. Rakin, Z. He, and D. Fan.

Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack.  
*2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 588–597, 2019.



P. Rathore, A. Basak, S. H. Nistala, and V. Runkana.

Untargeted, targeted and universal adversarial attacks and defenses on time series.  
*2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.



A. Ross and F. Doshi-Velez.

Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients.  
In *AAAI*, 2018.



A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein.

Are adversarial examples inevitable?

*ArXiv*, abs/1809.02104, 2019.



H. P. Silva, L. Seidenari, and A. Del Bimbo.

Diffdefense: Defending against adversarial attacks via diffusion models.  
In *Proc. Int. Conf. Image Anal. Process.*, pages 430–442, 2023.



K. Simonyan and A. Zisserman.

Very deep convolutional networks for large-scale image recognition.  
*CoRR*, abs/1409.1556, 2015.

# References III

-  A. Wu, Y. Han, Q. Zhang, and X. Kuang.  
Untargeted adversarial attack via expanding the semantic gap.  
*2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 514–519, 2019.
-  C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille.  
Mitigating adversarial effects through randomization.  
*ArXiv*, abs/1711.01991, 2018.
-  J. Yoon, S. J. Hwang, and J. Lee.  
Adversarial purification with score-based generative models.  
*In Proc. Int. Conf. Mach. Learn.*, pages 12062–12072, 2021.
-  L. Yu, S. Liu, Y. Miao, X.-S. Gao, and L. Zhang.  
Generalization bound and new algorithm for clean-label backdoor attack, 2024.