ECE595 / STAT598: Machine Learning I
Lecture 33 Adversarial Attack: An Overview

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

PURDUE UNIVERSITY

Today's Agenda

- We have studied
  - Part 1: Basic learning pipeline
  - Part 2: Algorithms
  - Part 3: Learning theory
- Now, we want to study the robustness of learning algorithms
- Robustness = easiness to fail when input is perturbed. Perturbation can be in any kind.
- Robust machine learning is a very rich topic.
- In the past, we have robust SVM, robust kernel regression, robust PCA, etc.
- More recently, we have **transfer learning** etc.
- In this course, we will look at something very narrow, called **adversarial robustness**.
- That is, robustness against **attacks**.
- Adversarial attack is a very **hot** topic, as of today.
- We should not over-emphasize its importance. There are many other important problems.

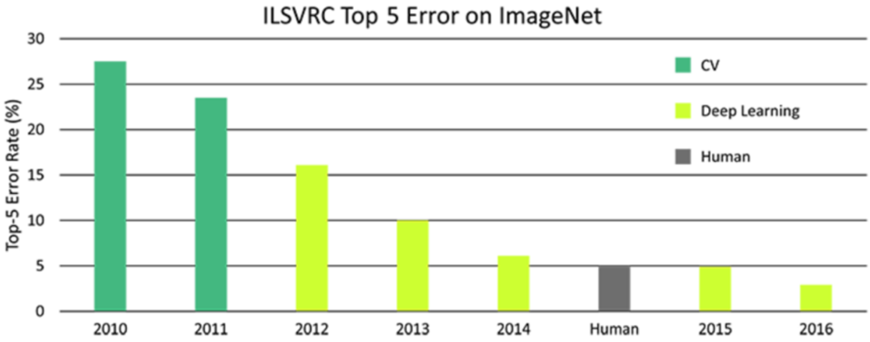
Outline

- Lecture 33 Overview
- Lecture 34 Min-distance attack
- Lecture 35 Max-loss attack and regularized attack

Today's Lecture

- What are adversarial attacks?
  - The surprising findings by Szegedy (2013) and Goodfellow (2014)
  - Examples of attacks
  - Physical attacks
- Basic terminologies
  - Defining attack
  - Multi-class problem
  - Three forms of attack
  - Objective function and constraint sets

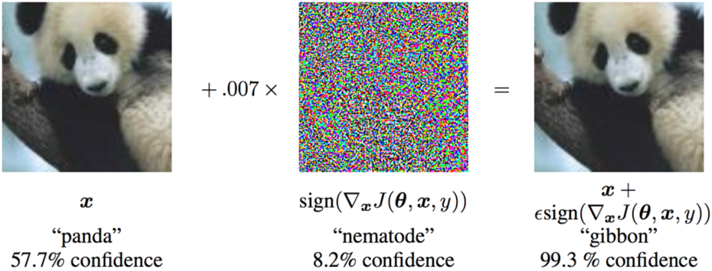
A Report in 2017



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

Adversarial Attack Example: FGSM

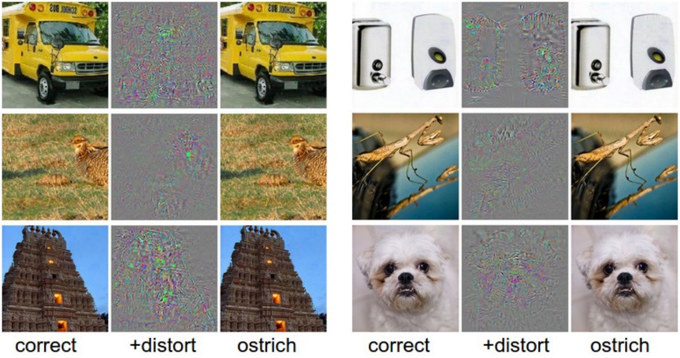
- It is not difficult to fool a classifier
- The perturbation could be perceptually not noticeable



Goodfellow et al. "Explaining and Harnessing Adversarial Examples", https://arxiv.org/pdf/1412.6572.pdf

Adversarial Attack Example: Szegedy's 2013 Paper

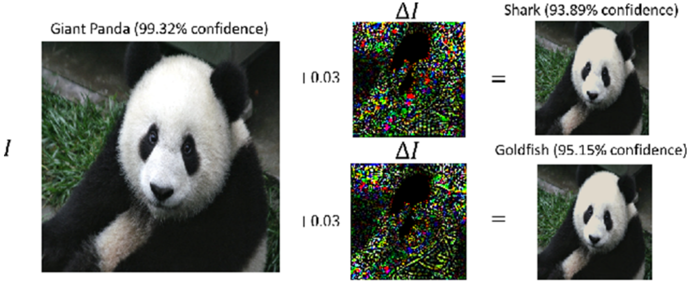
- This paper actually appears one year before Goodfellow's 2014 paper.



Szegedy et al. Intriguing properties of neural networks
https://arxiv.org/abs/1312.6199

Adversarial Attack: Targeted Attack

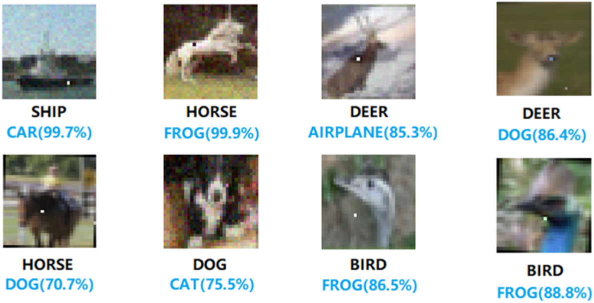
- Targeted Attack



Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics, https://arxiv.org/abs/1612.07767

Adversarial Attack Example: One Pixel

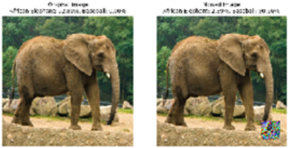
- One-pixel Attack



One pixel attack for fooling deep neural networks https://arxiv.org/abs/1710.08864

Adversarial Attack Example: Patch

- Adding a patch



African-Elephant (92.8%) → Baseball (90.7%)

Sports Car (92.8%) → Shih-Tzu (90.7%)

Brown Bear (87.9%) → **Tree Frog** (82.7%)

Minivan (90.7%) → **Tree Frog** (86.4%)

LaVAN: Localized and Visible Adversarial Noise, https://arxiv.org/abs/1801.02608

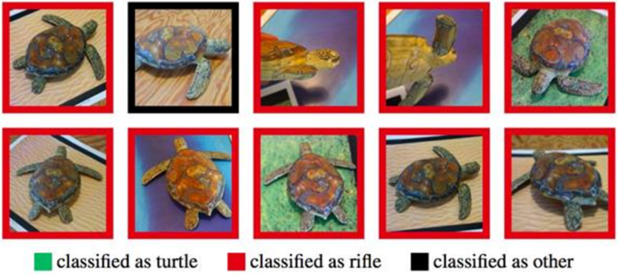
Adversarial Attack Example: Stop Sign

- The Michigan / Berkeley Stop Sign

Robust Physical-World Attacks on Deep Learning Models
https://arxiv.org/abs/1707.08945

Adversarial Attack Example: Turtle

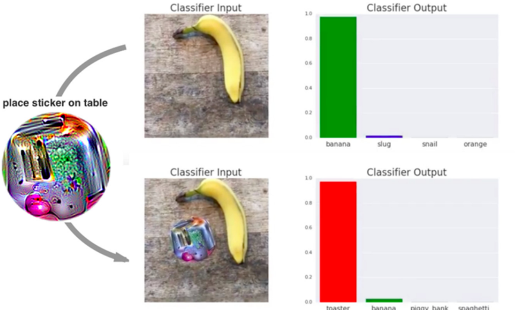
- The MIT 3D Turtle



Synthesizing Robust Adversarial Examples
https://arxiv.org/pdf/1707.07397.pdf
https://www.youtube.com/watch?v=YXy6oX1iNoA

Adversarial Attack Example: Toaster

- Google Toaster



Adversarial Patch
https://arxiv.org/abs/1712.09665
https://www.youtube.com/watch?v=i1sp4X57TL4

Adversarial Attack Example: Glass

- CMU Glass

Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2016, October).
Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition.
In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1528-1540). ACM.

Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition
`https://www.cs.cmu.edu/~sbhagava/papers/face-rec-ccs16.pdf`
`https://www.archive.ece.cmu.edu/~lbauer/proj/advml.php`

Adversarial Attack: A Survey in 2017

Table III: Summary of Applications for Adversarial Examples

Applications	Representative Study	Method	Adversarial Falsification	Adversary's Knowledge	Adversarial Specificity	Perturbation Scope	Perturbation Limitation	Attack Frequency	Perturbation Measurement	Dataset	Architecture
Reinforcement Learning	[93]	FGSM	N/A	White-box & Black-box	Non-Targeted	Individual	N/A	One-time	$\ell_1, \ell_2, \ell_\infty$	Atari	DQN, TRPO, A3C
	[94]	FGSM	N/A	White-box	Non-Targeted	Individual	N/A	One-time	N/A	Atari Pong	A3C
Generative Modeling	[95]	Feature Adversary, C&W	N/A	White-box	Targeted	Individual	Optimized	Iterative	ℓ_2	MNIST, SVHN, CelebA	VAE, VAE-GAN
	[96]	Feature Adversary	N/A	White-box	Targeted	Individual	Optimized	Iterative	ℓ_2	MNIST, SVHN	VAE, AE
Face Recognition	[67]	Impersonation & Dodging Attack	False negative	white-box & black-box	Targeted & Non-Targeted	Universal	Optimized	Iterative	Total Variation	LFW,	VGGFace
Object Detection	[22]	DAG	False negative & False positive	White-box & Black-box	Non-Targeted	Individual	N/A	Iterative	N/A	VOC2007, VOC2012	Faster-RCNN
Semantic Segmentation	[22]	DAG	False negative & False positive	White-box & Black-box	Non-Targeted	Individual	N/A	Iterative	N/A	DeepLab	FCN
	[97]	ILLC	False negative	White-box	Targeted	Individual	N/A	Iterative	ℓ_∞	Cityscapes	FCN
	[98]	ILLC	False negative	White-box	Targeted	Universal	N/A	Iterative	N/A	Cityscapes	FCN
Reading Comprehension	[99]	AddSent, AddAny	N/A	Black-box	Non-Targeted	Individual	N/A	One-time & Iterative	N/A	SQuAD	BiDAF, Match-LSTM, and twelve other published models
	[100]	Reinforcement Learning	False negative	White-box	Non-Targeted	Individual	Optimized	Iterative	ℓ_0	TripAdvisor Dataset	Bi-LSTM, memory network
Malware Detection	[101]	JSMA	False negative	White-box	Targeted	Individual	Optimized	Iterative	ℓ_2	DREBIN	2-layer FC
	[102]	Reinforcement Learning	False negative	Black-box	Targeted	Individual	N/A	Iterative	N/A	N/A	Gradient Boosted Decision Tree
	[103]	GAN	False negative	Black-box	Targeted	Individual	N/A	Iterative	N/A	malwr	Multi-layer Perceptron
	[104]	GAN	False negative	Black-box	Targeted	Individual	N/A	Iterative	N/A	Alexa Top 1M	Random Forest
	[105]	Generic Programming	False negative	Black-box	Targeted	Individual	N/A	Iterative	N/A	Contagio	Random Forest, SVM

Adversarial Examples: Attacks and Defenses for Deep Learning
https://arxiv.org/abs/1712.07107

Outline

- Lecture 33 Overview
- Lecture 34 Min-distance attack
- Lecture 35 Max-loss attack and regularized attack

Today's Lecture

- What are adversarial attacks?
  - The surprising findings by Szegedy (2013) and Goodfellow (2014)
  - Examples of attacks
  - Physical attacks
- Basic terminologies
  - Defining attack
  - Multi-class problem
  - Three forms of attack
  - Objective function and constraint sets

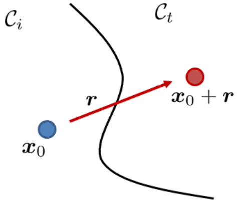
Definition: Additive Adversarial Attack

Definition (Additive Adversarial Attack)

Let $\boldsymbol{x}_0 \in \mathbb{R}^d$ be a data point belong to class $\mathcal{C}_i$. Define a target class $\mathcal{C}_t$. An **additive** adversarial attack is an addition of a perturbation $\boldsymbol{r} \in \mathbb{R}^d$ such that the perturbed data

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{r}$$

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



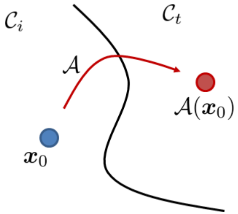
Definition: General Adversarial Attack

Definition (Adversarial Attack)

Let $\boldsymbol{x}_0 \in \mathbb{R}^d$ 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

$$\boldsymbol{x} = \mathcal{A}(\boldsymbol{x}_0)$$

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



Example: Geometric Attack

Fast Geometrically-Perturbed Adversarial Faces (WACV 2019)

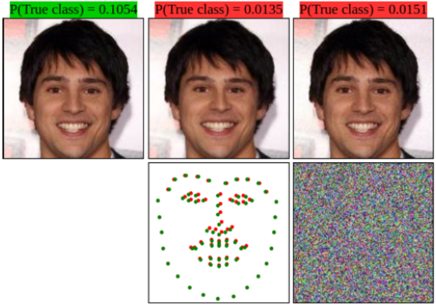
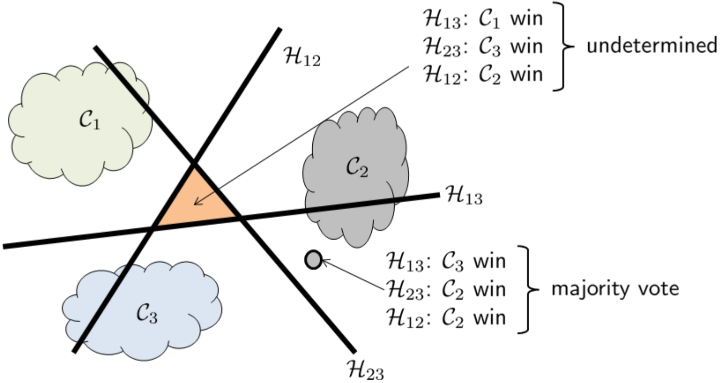


Figure 1. Comparison of the proposed attack to an intensity-based attack. First column: the ground truth image, which is correctly classified. Second column: the spatially transformed adversarial image wrongly classified and the corresponding adversarial landmark locations computed by our method. Third column: the adversarial image wrongly classified and the corresponding perturbation generated by the fast gradient sign method [7]. The proposed method leads to natural adversarial faces which are clean from additive noise.

https://arxiv.org/pdf/1809.08999.pdf

The Multi-Class Problem

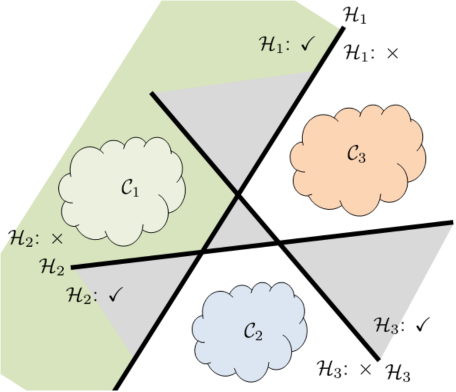
Approach 1: One-on-One



- Class i VS Class j
- Give me a point, check which class has more votes
- There is an undetermined region

The Multi-Class Problem

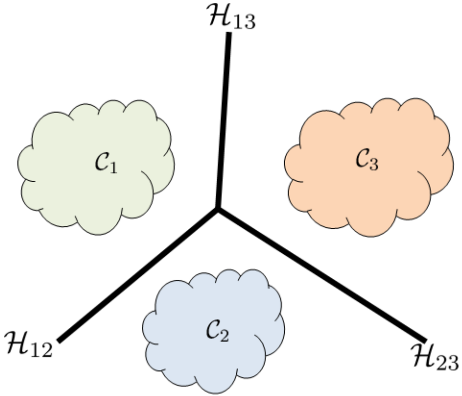
Approach 2: One-on-All



- Class i VS not Class i
- Give me a point, check which class has no conflict
- There are undetermined regions

The Multi-Class Problem

Approach 3: Linear Machine



- Every point in the space gets assigned a class.
- You give me $\boldsymbol{x}$, I compute $g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_K(\boldsymbol{x})$.
- If $g_i(\boldsymbol{x}) \geq g_j(\boldsymbol{x})$ for all $j \neq i$, then $\boldsymbol{x}$ belongs to class i.

Correct Classification

- We are mostly interested the linear machine problem.
- Let us try to simplify the notation. The statement:

  If $g_i(\boldsymbol{x}) \geq g_j(\boldsymbol{x})$ for all $j \neq i$, then $\boldsymbol{x}$ belongs to class i.

  is equivalent to (asking everyone to be less than 0)

$$
\begin{aligned}
g_1(\boldsymbol{x}) - g_i(\boldsymbol{x}) &\leq 0 \\
&\vdots \\
g_k(\boldsymbol{x}) - g_i(\boldsymbol{x}) &\leq 0,
\end{aligned}
$$

- and is also equivalent to (asking the worst guy to be less than 0)

$$
\max_{j \neq i} \{g_j(\boldsymbol{x})\} - g_i(\boldsymbol{x}) \leq 0
$$

- Therefore, if I want to launch an **adversarial attack**, I want to move you to class t:

$$
\max_{j \neq t} \{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x}) \leq 0.
$$

Our Approach

Here is what we are going to do

- First, we will preview the three **equivalent** forms of attack:
  - 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
- Then, we will try to understand the **geometry** of the attacks.
- We will look at the **linear classifier** case to gain insights.

Minimum Distance Attack

Definition (Minimum Distance Attack)

The **minimum distance attack** finds a perturbed data $\boldsymbol{x}$ by solving the optimization

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \|\boldsymbol{x} - \boldsymbol{x}_0\| \\ \text{subject to} \quad & \max_{j \neq t} \{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x}) \leq 0, \end{aligned} \tag{1}$$

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

- I want to make you to class $\mathcal{C}_t$.
- So the constraint needs to be satisfied.
- But I also want to minimize the attack strength. This gives the objective.

Maximum Loss Attack

Definition (Maximum Loss Attack)

The **maximum loss attack** finds a perturbed data $\boldsymbol{x}$ by solving the optimization

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{maximize}} \quad & g_t(\boldsymbol{x}) - \max_{j \neq t} \{g_j(\boldsymbol{x})\} \\ \text{subject to} \quad & \|\boldsymbol{x} - \boldsymbol{x}_0\| \leq \eta, \end{aligned} \tag{2}$$

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

- I want to bound my attack $\|\boldsymbol{x} - \boldsymbol{x}_0\| \leq \eta$
- I want to make $g_t(\boldsymbol{x})$ as big as possible
- So I want to maximize $g_t(\boldsymbol{x}) - \max_{j \neq t} \{g_j(\boldsymbol{x})\}$
- This is equivalent to

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

Regularization-based Attack

Definition (Regularization-based Attack)

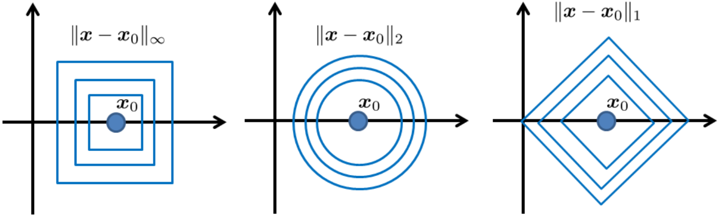
The **regularization-based attack** finds a perturbed data $\boldsymbol{x}$ by solving the optimization

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\| + \lambda \left(\max_{j \neq t} \{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x}) \right) \tag{3}$$

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.

Understanding the Geometry: Objective Function



- ℓ_0-norm: $\varphi(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_0\|_0$, which gives the most sparse solution. Useful when we want to limit the number of attack pixels.
- ℓ_1-norm: $\varphi(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_0\|_1$, which is a convex surrogate of the ℓ_0-norm.
- ℓ_∞-norm: $\varphi(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty$, which minimizes the maximum element of the perturbation.

Understanding the Geometry: Constraint

- The constraint set is

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

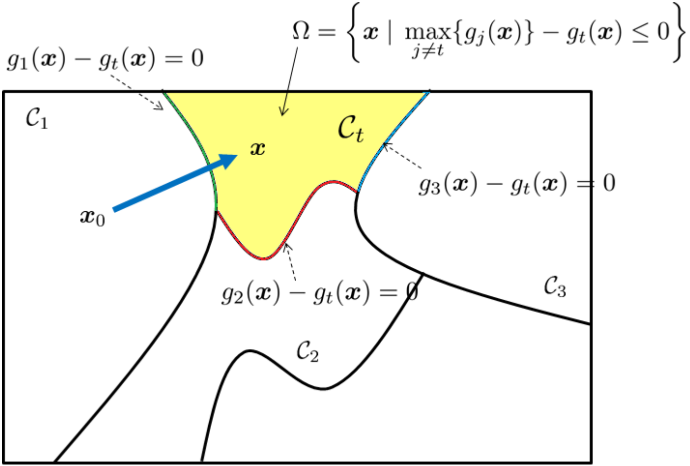
- We can write Ω as

$$\Omega = \left\{ \boldsymbol{x} \;\middle|\; \begin{array}{ll} g_1(\boldsymbol{x}) - g_t(\boldsymbol{x}) & \leq 0 \\ g_2(\boldsymbol{x}) - g_t(\boldsymbol{x}) & \leq 0 \\ \vdots & \\ g_k(\boldsymbol{x}) - g_t(\boldsymbol{x}) & \leq 0 \end{array} \right\}$$

- Remark: If you want to replace max by i^*, then i^* is a function of $\boldsymbol{x}$:

$$\Omega = \left\{\boldsymbol{x} \mid g_{i^*(\boldsymbol{x})}(\boldsymbol{x}) - g_t(\boldsymbol{x}) \leq 0\right\}.$$

Understanding the Geometry: Constraint



Linear Classifier

- Let us take a closer look at the linear case.
- Each discriminant function takes the form

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i,0}.$$

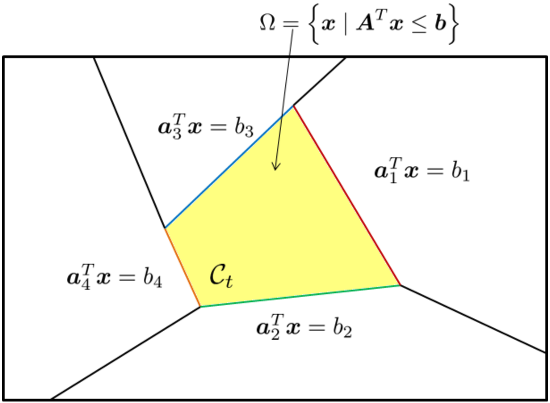
- The decision boundary between the i-th class and the t-th class is therefore

$$g(\boldsymbol{x}) = (\boldsymbol{w}_i - \boldsymbol{w}_t)^T \boldsymbol{x} + w_{i,0} - w_{t,0} = 0.$$

- The constraint set Ω is

$$\begin{bmatrix} \boldsymbol{w}_1^T - \boldsymbol{w}_t^T \\ \vdots \\ \boldsymbol{w}_{t-1}^T - \boldsymbol{w}_t^T \\ \boldsymbol{w}_{t+1}^T - \boldsymbol{w}_t^T \\ \vdots \\ \boldsymbol{w}_k^T - \boldsymbol{w}_t^T \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} w_{1,0} - w_{t,0} \\ \vdots \\ w_{t-1,0} - w_{t,0} \\ w_{t+1,0} - w_{t,0} \\ \vdots \\ w_{k,0} - w_{t,0} \end{bmatrix} \le \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{A}^T \boldsymbol{x} \le \boldsymbol{b}$$

Linear Classifier



- You can show $\Omega = \{\boldsymbol{A}^T \boldsymbol{x} \leq \boldsymbol{b}\}$ is convex.
- But the complement $\Omega^c = \{\boldsymbol{A}^T \boldsymbol{x} > \boldsymbol{b}\}$ is not convex.
- So targeted attack is easier to analyze than untargeted attack.

Attack: The Simplest Example

The optimization is:

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

- Suppose we use ℓ_2-norm, and consider **linear** classifiers, then
- the attack is given by

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \quad \text{subject to} \quad \boldsymbol{A}^T \boldsymbol{x} \leq \boldsymbol{b},$$

- This is a **quadratic programming** problem.
- We will discuss how to solve this problem analytically.

Summary

- Adversarial attack is a universal phenomenon for **any** classifier.
- Attacking deep networks are popular because people think that they are unbeatable.
- There is really nothing too magical behind adversarial attack.
- All attacks are based on one of the three forms of attacks.
- Deep networks are trickier, as we will see, because the internal model information is not easy to extract.
- We will learn the basic principles of attacks, and try to gain insights from linear models.