ECE595 / STAT598: Machine Learning I
Lecture 34 Min-Distance Attacks

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Today's Agenda

- Last lecture we have learned the basic terminologies of adversarial attack.
- In today's and the next lectures, we will go into the details of how to attack.
- We will discuss three forms of attacks
  - Min-distance attack
  - Max-loss attack
  - Regularized attack
- We will discuss everything for the **linear model**.
- And then we will talk about **deep models**.
- You are only required to know how to attack the linear model.
- For deep models, you probably need to have some prior experience with deep neural networks in order to understand what we are going to discuss.

Outline

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

Today's Lecture

- Linear models
  - Definition
  - Geometry
  - Optimization solutions
- Deep models
  - Deep fool
  - ℓ_∞ case

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.

Geometry: Attack as a Projection

What is the Geometry of the Attack?

- Claim: Attacking a data point = projecting it onto the decision boundary
- Let us look at ℓ_2 minimum distance attack

Theorem (Minimum-Distance Attack as a Projection)

The minimum-distance attack via ℓ_2 is equivalent to the projection

$$\begin{aligned} \boldsymbol{x}^* &= \underset{\boldsymbol{x}\in\Omega}{\text{argmin}} \;\; \|\boldsymbol{x}-\boldsymbol{x}_0\|^2, \quad \text{where} \quad \Omega = \{\boldsymbol{x} \mid \max_{j\neq t}\{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x}) \leq 0\}, \\ &= \mathcal{P}_\Omega(\boldsymbol{x}_0), \end{aligned}$$

where $\mathcal{P}_\Omega(\cdot)$ denotes the projection onto the set Ω.

Geometry: Attack as a Projection

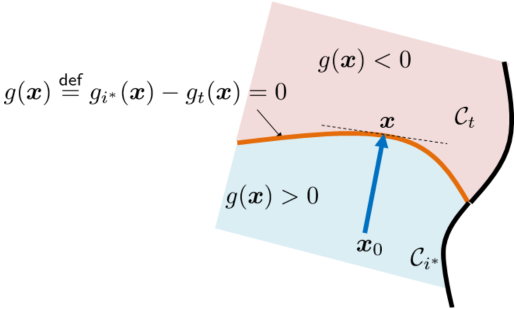
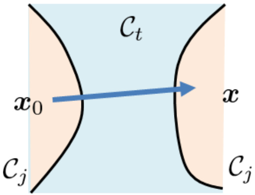
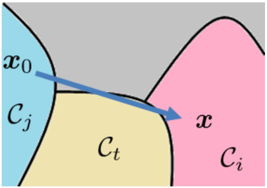


Figure: Geometry: Given an input data point $\boldsymbol{x}_0$, our goal is to send $\boldsymbol{x}_0$ to a targeted class $\mathcal{C}_t$ by minimizing the distance between $\boldsymbol{x}$ and $\boldsymbol{x}_0$. The decision boundary is characterized by $g(\boldsymbol{x}) = g_{i^*}(\boldsymbol{x}) - g_t(\boldsymbol{x})$. The optimal solution is the projection of $\boldsymbol{x}_0$ onto the decision boundary.

Geometry: Overshoot

- What if you move along the attack direction but overshoot?
- Define
$$\boldsymbol{x} = \boldsymbol{x}_0 + \alpha(\mathcal{P}_\Omega(\boldsymbol{x}_0) - \boldsymbol{x}_0).$$
- Three cases:
  - You overshoot but you still stay in the target class.
  - You overshoot and you go back to the original class.
  - You overshoot and you go to another class.





Targeted VS Untargeted Attack

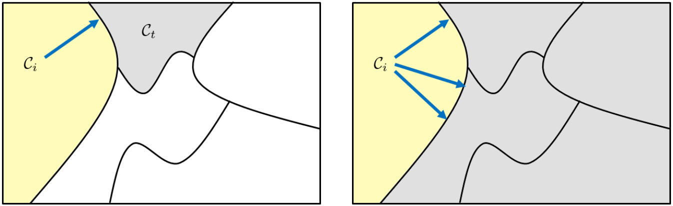


Figure: [Left] Targeted attack: The attack has to be specific from $\mathcal{C}_i$ to $\mathcal{C}_t$. [Right] Untargeted attack: The attack vector can point to anywhere outside $\mathcal{C}_i$.

- Targeted attack: The constraint set Ω is

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

- Untargeted attack: The constraint set Ω is

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

White-box VS Black-box Attack

- **White-box**: You know everything about the classifier.
- So you know all g_i's, completely.
- The constraint set is

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

- **Black-box**: You can only probe the classifier finite times.
- So you only know $\left\{g_i(\boldsymbol{x}^{(1)}), g_i(\boldsymbol{x}^{(2)}), \ldots, g_i(\boldsymbol{x}^{(M)})\right\}$.
- The constraint set is

$$\Omega = \{\boldsymbol{x} \mid \max_{j \neq t}\{\widehat{g}_j(\boldsymbol{x})\} - \widehat{g}_t(\boldsymbol{x}) \leq 0\},$$

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

Launching the Attack: Basic Principles

- **Principle 1**: You need to solve the optimization

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

  or its variations.
- **Principle 2**: You do not need to solve inequality. Equality is enough.
  - You just need to hit the decision boundary.
  - Then you add a small ϵ to your step.
- **Principle 3**: You do not need to be optimal.
  - Optimal = The nastiest attack.
  - You can still fool the classifier with a less nasty attack.
- **Our Plan**: Look at **linear** classifiers, and **binary** classifiers only.

So, if we restrict ourselves to binary linear classifiers ...

The min-distance attack (ℓ_2-norm)

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \\ \text{subject to} & \max_{j \neq t} \{g_j(\boldsymbol{x})\} - g_t(\boldsymbol{x}) \leq 0, \end{array}$$

will become ...

- **Linear** classifiers, we have

$$g_i(\boldsymbol{x}) - g_t(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0.$$

- **Two** class: the constraint is simplified to

$$g_i(\boldsymbol{x}) - g_t(\boldsymbol{x}) \leq 0$$

- And we just need to hit the boundary. So the attack becomes

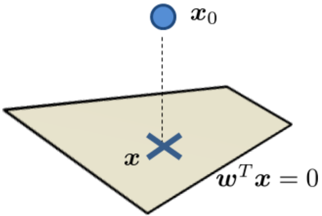
$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \\ \text{subject to} & \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0. \end{array}$$

Recall: Distance Between Point and Plane

What is the closest distance between a point and a plane?

- $\boldsymbol{w}^T\boldsymbol{x} = 0$ is a line.
- Find a point $\boldsymbol{x}$ on the line that is closest to $\boldsymbol{x}_0$.
- Solution is

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{x}_0 + \boldsymbol{w}(\boldsymbol{w}^T\boldsymbol{w})^{-1}(0 - \boldsymbol{w}^T\boldsymbol{x}_0) \\ &= \boldsymbol{x}_0 - \left(\frac{\boldsymbol{w}^T\boldsymbol{x}_0}{\|\boldsymbol{w}\|^2}\right)^T \boldsymbol{w}.\end{aligned}$$



Minimum-Distance Attack: Solving the Optimization

Theorem (Minimum ℓ_2 Norm Attack for Two-Class Linear Classifier)

The adversarial attack to a two-class linear classifier is the solution of

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \quad \text{subject to} \quad \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0,$$

which is given by

$$\boldsymbol{x}^* = \boldsymbol{x}_0 - \left(\frac{\boldsymbol{w}^T\boldsymbol{x}_0 + w_0}{\|\boldsymbol{w}\|_2}\right)\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}.$$

- This is just finding the closest point to a hyperplane!
- $\boldsymbol{w}/\|\boldsymbol{w}\|_2$ is the normal direction = best attack angle.
- $\frac{\boldsymbol{w}^T\boldsymbol{x}_0 + w_0}{\|\boldsymbol{w}\|_2}$ is the step size.

Minimum-Distance Attack: Two-Class Linear Classifier

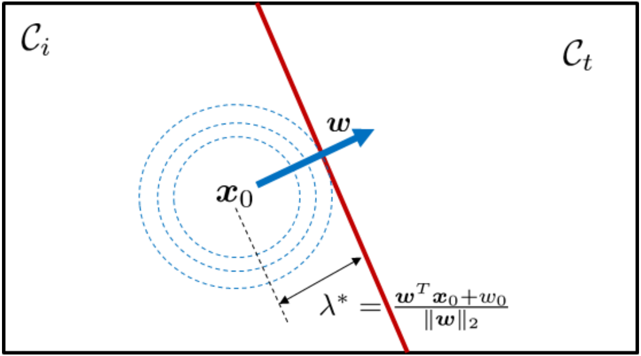


Figure: Geometry of minimum-distance attack for a two-class linear classifier with objective function $\|\boldsymbol{x} - \boldsymbol{x}_0\|^2$. The solution is a projection of the input $\boldsymbol{x}_0$ onto the separating hyperplane of the classifier.

Outline

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

Today's Lecture

- Linear models
  - Definition
  - Geometry
  - Optimization solutions
- Deep models
  - Deep fool
  - ℓ_∞ case

Deep-Fool (CVPR, 2016)

Let's Connect to the Real Problem.

- Proposed by Moosavi-Dezfooli, Fawzi and Frossard
- Generalize linear classifier to neural network

Definition (DeepFool Attack by Moosavi-Dezfooli et al. 2016)

The DeepFool attack for a two-class classification generates the attack by solving the optimization

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\|^2 \quad \text{subject to} \quad g(\boldsymbol{x}) = 0,$$

where $g(\boldsymbol{x}) = 0$ is the nonlinear decision boundary separating the two classes.

How to deal with non-linearity?

- First order approximation

$$g(\boldsymbol{x}) \approx g(\boldsymbol{x}^{(k)}) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T (\boldsymbol{x} - \boldsymbol{x}^{(k)}),$$

- Modify the problem (assume $\boldsymbol{x}^{(0)} = \boldsymbol{x}_0$)

$$\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\text{argmin}} \quad \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2 \qquad \text{subject to} \quad g(\boldsymbol{x}) = 0.$$

$$\vdots$$

$$\begin{aligned} \boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\text{argmin}} \quad & \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2 \\ \text{subject to} \quad & g(\boldsymbol{x}^{(k)}) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T (\boldsymbol{x} - \boldsymbol{x}^{(k)}) = 0. \end{aligned}$$

- Now, rewrite

$$\begin{aligned} & g(\boldsymbol{x}^{(k)}) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T (\boldsymbol{x} - \boldsymbol{x}^{(k)}) \\ & = \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T \boldsymbol{x} + g(\boldsymbol{x}^{(k)}) - \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T \boldsymbol{x}^{(k)}. \end{aligned}$$

How to deal with non-linearity?

- So here is our problem

$$\begin{aligned} \boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\text{argmin}} \quad & \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2 \\ \text{subject to} \quad & g(\boldsymbol{x}^{(k)}) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T (\boldsymbol{x} - \boldsymbol{x}^{(k)}) = 0. \end{aligned}$$

- Let $\boldsymbol{w}^{(k)} = \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})$ and $w_0^{(k)} = g(\boldsymbol{x}^{(k)}) - \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T \boldsymbol{x}^{(k)}$
- Then equivalent to

$$\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\text{argmin}} \quad \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2 \quad \text{subject to} \quad (\boldsymbol{w}^{(k)})^T \boldsymbol{x} + w_0^{(k)} = 0$$

- This is just a linear problem!

How to deal with non-linearity?

- Here is the optimization

$$\boldsymbol{x}^{(k+1)} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \quad \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2 \quad \text{subject to} \quad (\boldsymbol{w}^{(k)})^T \boldsymbol{x} + w_0^{(k)} = 0$$

- So the solution is

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} - \left(\frac{(\boldsymbol{w}^{(k)})^T \boldsymbol{x}^{(k)} + w_0^{(k)}}{\|\boldsymbol{w}^{(k)}\|^2} \right) \boldsymbol{w}^{(k)} \\ &= \boldsymbol{x}^{(k)} - \left(\frac{g(\boldsymbol{x}^{(k)})}{\|\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})\|^2} \right) \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}). \end{aligned}$$

- How to evaluate the gradient?
- $\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})$ can be computed via back propagation.

How to deal with non-linearity?

- Now, for this attack

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(\frac{g(\boldsymbol{x}^{(k)})}{\|\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})\|^2}\right) \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}).$$

- You can control the perturbation magnitude:

$$\boldsymbol{x}^{(k+1)} = \mathcal{P}_{[0,1]} \left\{ \boldsymbol{x}^{(k)} - \left(\frac{g(\boldsymbol{x}^{(k)})}{\|\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})\|^2}\right) \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}) \right\}.$$

- $\mathcal{P}_{[0,1]}$: Projection onto a ball, e.g., $\mathcal{P}_{[0,1]}(\boldsymbol{x})$ clips $\boldsymbol{x}$ to $[0, 1]$.

Corollary (DeepFool Algorithm for Two-Class Problem)

An iterative procedure to obtain the DeepFool attack solution is

$$\begin{aligned}\boldsymbol{x}^{(k+1)} &= \underset{\boldsymbol{x}}{argmin} \quad \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2 \\ &\qquad \text{subject to} \quad g(\boldsymbol{x}^{(k)}) + \nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})^T(\boldsymbol{x} - \boldsymbol{x}^{(k)}) = 0 \\ &= \boldsymbol{x}^{(k)} - \left(\frac{g(\boldsymbol{x}^{(k)})}{\|\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)})\|^2}\right)\nabla_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}),\end{aligned}$$

with $\boldsymbol{x}^{(0)} = \boldsymbol{x}_0$.

- This is not the complete Deep-fool.
- We assume two classes only.
- If you have multiple classes, you need to take care of "$\max_{j \neq t} g_j(\boldsymbol{x})$"

The ℓ_∞ Case

- How about we try to solve this?

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty \quad \text{subject to} \quad \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0.$$

- Not the ℓ_2-norm, but the ℓ_∞-norm.
- Let $\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{x}_0$, $b_0 = -(\boldsymbol{w}^T\boldsymbol{x}_0 + w_0)$.
- Rewrite the problem as

$$\underset{\boldsymbol{r}}{\text{minimize}} \quad \|\boldsymbol{r}\|_\infty \quad \text{subject to} \quad \boldsymbol{w}^T\boldsymbol{r} = b_0.$$

- Setup Lagrangian function and take derivative?

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\lambda}) = \|\boldsymbol{r}\|_\infty + \lambda(b_0 - \boldsymbol{w}^T\boldsymbol{r}).$$

- Doesn't work because ℓ_∞ is not differentiable.

Solving the ℓ_∞-norm Problem

Theorem (Holder's Inequality)

Let $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{y} \in \mathbb{R}^d$. Then,

$$-\|\boldsymbol{x}\|_p\|\boldsymbol{y}\|_q \leq |\boldsymbol{x}^T\boldsymbol{y}| \leq \|\boldsymbol{x}\|_p\|\boldsymbol{y}\|_q$$

for any p and q such that $\frac{1}{p} + \frac{1}{q} = 1$, where $p \in [1, \infty]$.

- Let $p = 1$ and $q = \infty$
- Can show that $|\boldsymbol{x}^T\boldsymbol{y}| \leq \|\boldsymbol{x}\|_1\|\boldsymbol{y}\|_\infty$
- Then

$$|b_0| = |\boldsymbol{w}^T\boldsymbol{r}| \leq \|\boldsymbol{w}\|_1\|\boldsymbol{r}\|_\infty, \quad \Longrightarrow \quad \|\boldsymbol{r}\|_\infty \geq \frac{|b_0|}{\|\boldsymbol{w}\|_1}.$$

- So $\|\boldsymbol{r}\|_\infty$ is lower bounded by a constant.
- If $\boldsymbol{r}^*$ can reach this lower bound, then $\boldsymbol{r}^*$ is the minimizer.

Solving the ℓ_∞-norm Problem

- How about this candidate?

$$\boldsymbol{r} = \eta \cdot \text{sign}(\boldsymbol{w})$$

  for some constant η to be determined.

- We can show that

$$\|\boldsymbol{r}\|_\infty = \max_i |\eta \cdot \text{sign}(w_i)| = |\eta|.$$

- So if we let $\eta = b_0/\|\boldsymbol{w}\|_1$, then we will have

$$\|\boldsymbol{r}\|_\infty = |\eta| = \frac{|b_0|}{\|\boldsymbol{w}\|_1}.$$

- Lower bound achieved! So the solution is

$$\boldsymbol{r} = \frac{|b_0|}{\|\boldsymbol{w}\|_1} \cdot \text{sign}(\boldsymbol{w})$$

The ℓ_∞ Solution

Theorem (Minimum Distance ℓ_∞ Norm Attack for Two-Class Linear Classifier)

The minimum distance ℓ_∞ norm attack for a two-class linear classifier, i.e.,

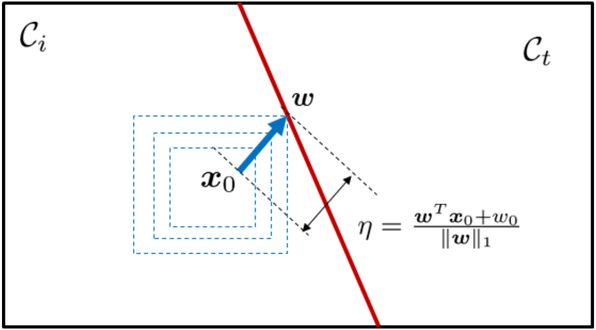
$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty \quad \text{subject to} \quad \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$$

is given by

$$\boldsymbol{x} = \boldsymbol{x}_0 - \left(\frac{\boldsymbol{w}^T\boldsymbol{x}_0 + w_0}{\|\boldsymbol{w}\|_1}\right) \cdot \text{sign}(\boldsymbol{w}).$$

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

The ℓ_∞ Solution



- Is it the "optimal" direction? No.
- The fastest search direction is ℓ_2.
- Can it move $\boldsymbol{x}_0$ to another class? Yes, if η is large enough.

Summary

Min-Distance Attack

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

- We have talked about the geometry.
- You can see that the geometry applies beyond linear models.
- For linear models, we can derive closed-form solutions.
- Deep models apply successive approximations.

Next Lecture

- **Max-Loss Attack**

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

- **Regularized Attack**

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