

CSC 144V: Autonomous Vehicles

Lecture 4.1

Adversarial Attacks Exercises Solution

Department of Computer Science,
Hofstra University

Exercise: FGSM

A toy “image” has 3 pixels, $x = (0.5, 0.2, 0.8)$, with valid pixel values in $[0, 1]$. The attacker uses an ℓ_∞ bound $\epsilon = 0.1$. At $\delta = 0$ the gradient of the loss with respect to the input is $\nabla \delta \text{ Loss} = (2.0, -0.5, 0.0)$.

FGSM.

- (i) Compute δ and the adversarial input $x + \delta$.
- (ii) Compute the ℓ_1 , ℓ_2 and ℓ_∞ norms of δ .

	pixel 1	pixel 2	pixel 3
x	0.5	0.2	0.8
$\nabla \delta \text{ Loss}$	2.0	-0.5	0.0
$\text{sign}(\nabla)$			
δ			
$x + \delta$			

ℓ_1	ℓ_2	ℓ_∞

Exercise Solution: FGSM

- $\text{sign}(\nabla) = (+1, -1, 0)$; $\text{sign}(0) = 0$.
- $\delta = \varepsilon \cdot \text{sign}(\nabla) = (0.1, -0.1, 0)$.
- $x + \delta = (0.6, 0.1, 0.8)$, all within $[0, 1]$.
- $\ell_1 = 0.1 + 0.1 + 0 = 0.2$
- $\ell_2 = \sqrt{0.01 + 0.01} = 0.141$
- $\ell_\infty = \max |\delta_i| = 0.1$

	pixel 1	pixel 2	pixel 3
x	0.5	0.2	0.8
$\nabla \delta$ Loss	2.0	-0.5	0.0
$\text{sign}(\nabla)$	+1	-1	0
δ	0.1	-0.1	0
$x + \delta$	0.6	0.1	0.8

ℓ_1	ℓ_2	ℓ_∞
0.2	0.141	0.1

Exercise: PGD

Use **PGD** with step size $\alpha = 0.04$ and start from $\delta = 0$ (same x , $\varepsilon = 0.1$ as before).

- In each iteration take $\delta \leftarrow \delta + \alpha \cdot \text{sign}(\text{gradient})$.
- Then **project** δ back into the ε -ball: clip every coordinate to $[-\varepsilon, \varepsilon]$.
- The gradient signs in the 3 iterations are $(+, -, +)$, $(+, -, +)$ and $(+, -, -)$.

Fill in the table.

Iteration	sign(gradient)	δ before projection	δ after projection
1	$(+, -, +)$		
2	$(+, -, +)$		
3	$(+, -, -)$		

Exercise Solution: PGD

- Iteration 3 needs the projection: $0.12 \rightarrow 0.10$ and $-0.12 \rightarrow -0.10$.
- Unlike FGSM, PGD's final δ need not reach a corner of the ε -ball (third pixel stays at 0.04); final $x + \delta = (0.6, 0.1, 0.84)$ is still in $[0, 1]$.

Iteration	sign(gradient)	δ before projection	δ after projection
1	(+, -, +)	(0.04, -0.04, 0.04)	(0.04, -0.04, 0.04)
2	(+, -, +)	(0.08, -0.08, 0.08)	(0.08, -0.08, 0.08)
3	(+, -, -)	(0.12, -0.12, 0.04)	(0.10, -0.10, 0.04)

Exercise: Adversarial training

Fill in the blanks of the adversarial-robust objective:

min over _____ of $E[$ max over _____ of $\text{Loss}(x + \delta, y; \theta)$]

and state which gradient is used to update θ (Danskin's theorem).

Exercise Solution: Adversarial training

min over θ of $E[\text{max over } \delta \in \Delta \text{ of } \text{Loss}(x + \delta, y; \theta)]$

- Inner maximization: crafts the worst-case perturbation (the attack). Outer minimization: trains the classifier.
- By **Danskin's theorem** the gradient is taken at the maximizer: $\nabla_{\theta} \text{Loss}(x + \delta^*, y; \theta)$, with $\delta^* = \text{argmax over } \Delta \text{ of } \text{Loss}(x + \delta, y; \theta)$.
- Valid if the max is solved exactly, so a strong inner attack such as PGD matters.

Exercise: Physical attack

A **sticker** is optimized to make a camera misread a stop sign.

Name **two real-world factors** that a pixel-level ε -ball attack ignores, and state how the optimization is modified to handle them.

Exercise Solution: Physical attack

- **Factors:** viewing angle, distance and rotation; lighting; camera noise; printing colors (also: only part of the sign may be modified, e.g. a sticker).
- **Modification:** average the objective over k transformed copies of the sign (expectation over transformation):

argmin over δ of $\lambda \|M \cdot \delta\| + (1/k) \sum_i J(f(x_i + M \cdot \delta), y_{\text{target}})$

- The mask M localizes the perturbation (ℓ_1 promotes sparsity), so a single δ works across conditions.