Augmented Lagrangian Adversarial Attacks

Chao Chen

LEHIGH UNIVERSITY

Notations

Sample $x \in \mathcal{X} \subset \mathbb{R}^d, y \in \mathcal{Y}$. In this paper, $\mathcal{X} = [0,1]^d$

A model $f: \mathbb{R}^d \to \mathbb{R}^K$ maps x to logits (before softmax) $z \in \mathbb{R}^K$.

The classification probability $p_y(y|x) = \mathrm{softmax}_y(z)$

$f_k(x)$ is the k-th element of the vector $f(x)$.

Adversarial example generation

To find adversarial examples, two alternatives:

1. Satisfying a distance constraint:

$$\text{find } \delta \qquad \text{s.t. } \operatorname{argmax}_k f_k(x+\delta) \neq y$$
$$\boxed{D(x+\delta, x) \leq \epsilon;} \quad x+\delta \in \mathcal{X}$$

2. Minimize distortions w.r.t. a distance function:

$$\min_{\delta} D(x+\delta, x) \qquad \text{s.t. } \operatorname{argmax}_k f_k(x+\delta) \neq y$$
$$x+\delta \in \mathcal{X}$$

D is a distance function.

Solving Eq. (2) is equivalent to solving Eq. (1) for every ϵ.

The constraint $x+\delta \in \mathcal{X}$ is handled by a simple projection operation $\mathcal{P}_{[0,1]}$ (omit in the rest).

Adversarial example generation

$$\min_{\delta} D(x+\delta, x) \qquad \text{s.t.} \;\; \text{argmax}_k \, f_k(x+\delta) \neq y$$

Since argmax is not differentiable, we replace it with an inequality constraint on logits:

$$\min_{\delta} D(x+\delta, x) \qquad \text{s.t.} \;\; f_y(x+\delta) - \max_{k\neq y} f_k(x+\delta) < 0$$

However, the constraint is not scale invariant, which results in "gradient masking":

For some x with $\nabla_x \ell(x,y) \approx 0$, it becomes in practice $\nabla_x \ell(x,y) = 0$, since typically single precision is used. Thus the sign of the gradient becomes zero and one does not get meaningful ascent directions.

Replace the constraint with Difference of Logits Ratio (DLR)

$$\min_{\delta} D(x+\delta, x) \qquad \text{s.t.} \;\; \text{DLR}^{+}(f(x+\delta), y) < 0$$

$$\text{DLR}^{+}(z,y) = \frac{z_y - \max\limits_{i\neq y} z_i}{z_{\pi_1} - z_{\pi_3}}$$

$z = f(x)$ and π is the decreasing ordering of elements of z.

General Lagrangian algorithm

$$\min g(x) \qquad \text{s.t.} \;\; h(x) < 0$$

Construct a Lagrangian function and set $P(h(x), \rho) = -\rho h(x)$:

$$G(x, \rho) = g(x) + P(h(x), \rho) = g(x) - \rho h(x), \rho \in \mathbb{R}_+$$

To find the partial derivative of G w.r.t. x and ρ, and set them to zero:

$$\frac{\partial G}{\partial x} = 0, \qquad \frac{\partial G}{\partial \rho} = 0$$

Usually, it is hard to find zero-gradient points directly, we use gradient descent:

$$x^{(i+1)} = x^{(i)} - \eta_x \frac{\partial G}{\partial x}, \qquad \rho^{(i+1)} = \rho^{(i)} + \eta_\rho \frac{\partial G}{\partial \rho}$$

General Augmented Lagrangian algorithm

$$\min g(x) \qquad \text{s.t.} \;\; h(x) < 0$$

Construct a Lagrangian function with $P(h(x), \rho, \mu)$:

$$G(x, \rho, \mu) = g(x) + P(h(x), \rho, \mu), \rho \in \mathbb{R}_+, \mu \in \mathbb{R}_+$$

$P(h(x), \rho, \mu)$ is a penalty-Lagrangian function such that $P'(y, \rho, \mu) = \frac{\partial}{\partial y} P(y, \rho, \mu)$ exists and is continuous for all $y \in \mathbb{R}$ and $(\rho, \mu) \in \mathbb{R}_+^2$, and satisfies four axioms:

Axiom 1. $\forall y \in \mathbb{R}, \forall(\rho, \mu) \in (\mathbb{R}_+^*)^2, \; \frac{\partial}{\partial y} P(y, \rho, \mu) \geq 0$

Axiom 2. $\forall(\rho, \mu) \in (\mathbb{R}_+^*)^2, \; \frac{\partial}{\partial y} P(0, \rho, \mu) = \mu$

Axiom 3. If, for all $j \in \mathbb{N}$, $0 < \mu_{\min} \leq \mu^{(j)} \leq \mu_{\max} < \infty$, then: $\lim_{j\to\infty} \rho^{(j)} = \infty$ and $\lim_{j\to\infty} y^{(j)} = y > 0$ imply that $\lim_{j\to\infty} \frac{\partial}{\partial y} P(y^{(j)}, \rho^{(j)}, \mu^{(j)}) = \infty$

Axiom 4. If, for all $j \in \mathbb{N}$, $0 < \mu_{\min} \leq \mu^{(j)} \leq \mu_{\max} < \infty$, then: $\lim_{j\to\infty} \rho^{(j)} = \infty$ and $\lim_{j\to\infty} y^{(j)} = y < 0$ imply that $\lim_{j\to\infty} \frac{\partial}{\partial y} P(y^{(j)}, \rho^{(j)}, \mu^{(j)}) = 0$

Example for $P(y, \rho, \mu)$:

$$\mathrm{PHR}(y, \rho, \mu) = \frac{1}{2\rho}(\max\{0, \mu + \rho y\}^2 - \mu^2) \tag{19}$$

$$P_1(y, \rho, \mu) = \begin{cases} \mu y + \frac{1}{2}\rho y^2 + \rho^2 y^3 & \text{if} \quad y \geq 0 \\ \mu y + \frac{1}{2}\rho y^2 & \text{if} \quad -\frac{\mu}{\rho} \leq y \leq 0 \\ -\frac{1}{2\rho}\mu^2 & \text{if} \quad y \leq -\frac{\mu}{\rho} \end{cases} \tag{20}$$

$$P_2(y, \rho, \mu) = \begin{cases} \mu y + \mu\rho y^2 + \frac{1}{6}\rho^2 y^3 & \text{if} \quad y \geq 0 \\ \frac{\mu y}{1 - \rho y} & \text{if} \quad y \leq 0 \end{cases} \tag{21}$$

$$P_3(y, \rho, \mu) = \begin{cases} \mu y + \mu\rho y^2 & \text{if} \quad y \geq 0 \\ \frac{\mu y}{1 - \rho y} & \text{if} \quad y \leq 0 \end{cases} \tag{22}$$

General Augmented Lagrangian algorithm

Line 2 - 3: inner iteration to find minimal x on current situation.

Line 4: update μ

When $h(x)$ is not satisfied:

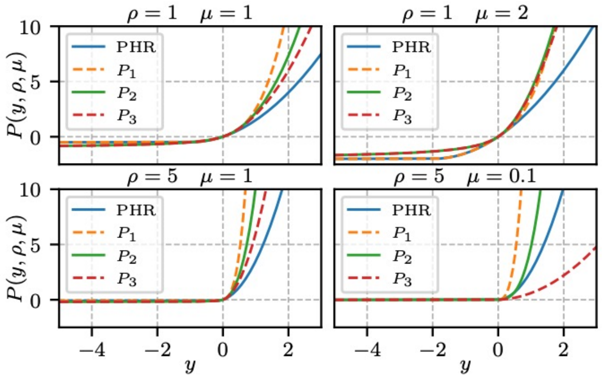
$\rightarrow P'(h(x), \rho, \mu)$ should be large (to infinity)

$\rightarrow$ increase μ (weights of penalty)

Line 5-9: update ρ

$\rightarrow$ if constraints $h(x)$ do not reduce significantly

$\rightarrow$ multiply ρ with a fixed factor (2~100)



Algorithm 1 Generic Augmented Lagrangian method

Require: Function to minimize f, constraint function g
Require: Initial value $\boldsymbol{x}^{(0)}$
Require: Penalty function P, initial multiplier $\mu^{(0)}$, $\rho^{(0)}$

1: **for** $i \leftarrow 0$ to $N-1$ **do**
2: Using $\boldsymbol{x}^{(i)}$ as initialization, minimize (approximately): $G(\boldsymbol{x}) = g(\boldsymbol{x}) + P(h(\boldsymbol{x}), \rho^{(i)}, \mu^{(i)})$
3: $\boldsymbol{x}^{(i+1)} \leftarrow$ approximate minimizer of G
4: $\mu^{(i+1)} \leftarrow P'(h(\boldsymbol{x}^{(i+1)}), \rho^{(i)}, \mu^{(i)})$
5: **if** the constraint does not improve **then**
6: Set $\rho^{(i+1)} > \rho^{(i)}$
7: **else**
8: $\rho^{(i+1)} \leftarrow \rho^{(i)}$
9: **end if**
10: **end for**

Augmented Lagrangian Attack

LEHIGH UNIVERSITY

Compare Generic Augmented Lagrangian and ALMA algorithms.

Recall that $\min_{\delta} D(x+\delta, x) \quad \text{s.t.} \quad \mathrm{DLR}^{+}(f(x+\delta), y) < 0$

Use EMA and projection gradient to update μ;

Update x and μ update jointly and discard inner iterations;

New rules to update ρ.

Algorithm 1 Generic Augmented Lagrangian method

Require: Function to minimize f, constraint function g
Require: Initial value $\boldsymbol{x}^{(0)}$
Require: Penalty function P, initial multiplier $\mu^{(0)}$, $\rho^{(0)}$

```
for i ← 0 to N − 1 do
    Using x^(i) as initialization, minimize (approximately):
    G(x) = g(x) + P(h(x), ρ^(i), μ^(i))
    x^(i+1) ← approximate minimizer of G
    μ^(i+1) ← P'(h(x^(i+1)), ρ^(i), μ^(i))
    if the constraint does not improve then
        Set ρ^(i+1) > ρ^(i)
    else
        ρ^(i+1) ← ρ^(i)
    end if
end for
```

Algorithm 2 ALMA attack

Require: Classifier f, original image $\boldsymbol{x}$, true or target label y
Require: Number of iterations N, initial step size $\eta^{(0)}$, penalty parameter increase rate $\gamma > 1$, constraint improvement rate $\tau \in [0,1]$, M number of steps between ρ increase.
Require: D distance function
Require: Penalty function P, initial multiplier $\mu^{(0)}$, initial penalty parameter $\rho^{(0)}$

```
Initialize x̃^(0) ← x,
for i ← 0 to N − 1 do
    z ← f(x̃^(i))
    d^(i) ← DLR⁺(z, y)                              ▷ tDLR⁺ for targeted attack
    μ̂ ← ∇_d P(d^(i), ρ^(i), μ^(i))                 ▷ New penalty multiplier
    μ^(i+1) ← 𝒫_[μmin, μmax][α μ^(i) + (1 − α) μ̂]   ▷ EMA
    L ← D(x̃^(i), x) + P(d^(i), ρ^(i), μ^(i+1))      ▷ Loss
    g ← ∇_x̃ L                                       ▷ Gradient of loss w.r.t. x̃
    x̃^(i+1) ← 𝒫_[0,1][x̃^(i) − η^(i) g]             ▷ Step and box-constraint
    if (i + 1) mod M = 0 and d^(j) > 0, ∀j ∈ {0, ..., i}
    and d^(i) > τ d^(i−M) then
        ρ^(i+1) ← γ ρ^(i)                           ▷ If no adversarial has been
    else                                            found and d does not de-
        ρ^(i+1) ← ρ^(i)                             crease significantly, in-
    end if                                          crease ρ by a factor of γ
end for
Return x̃^(i) that is adversarial and has smallest D(x̃^(i), x)
```

Augmented Lagrangian Attack

Use EMA and projection gradient to update μ to alleviate the spiking values.

Increase ρ when no adversarial example is found, and the constraint doesn't improve.

Other tricks:

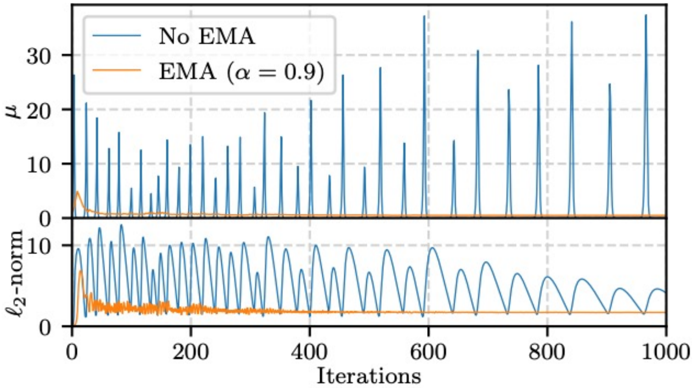
Penalty function $P(h(x), \rho, \mu)$ is of great importance.

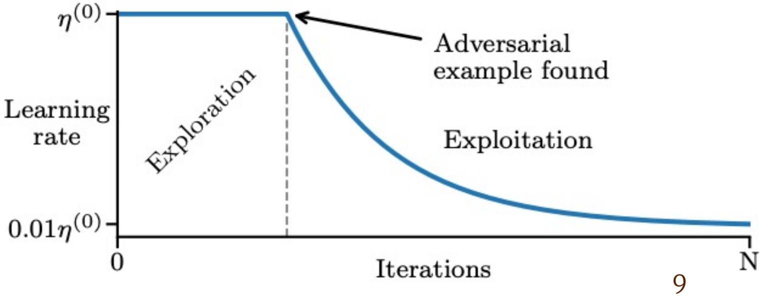
$$P_2(y, \rho, \mu) = \begin{cases} \mu y + \mu \rho y^2 + \frac{1}{6}\rho^2 y^3 & \text{if} \quad y \geq 0 \\ \frac{\mu y}{1 - \rho y} & \text{if} \quad y < 0 \end{cases}$$

Initialize learning rate adaptively.

Apply learning rate decay to balance exploration and exploitation.

Replace classic gradient descent with RMSProp and momentum method.





Experiments

Datasets: MNIST, CIFAR10, ImageNet

Budgets: 100 and 1000 iterations

Metrics:

Attack Success Rate (ASR): the proportion of examples for which an adversarial perturbation is found; and the median perturbation size.

Complexity: the average number of forward and backward propagations per sample needed.

Target models:

For MNIST: 1. *SmallCNN* with regularly training; 2. *SmallCNN-DDN* with l_2-adversarially training;

3. SmallCNN-TRADES with l_∞-adversarially training; 4. CROWN-IBP with l_∞-adversarially training.

For CIFAR10: 1. Wide ResNet 28-10 with regularly training; 2. Wide ResNet 28-10 with l_∞-adversarially training;

3. ResNet-50 with l_2-adversarially training.

For ImageNet: 1. ResNet with regularly training; 2. ResNet with l_∞-adversarially training;

3. ResNet with l_2-adversarially training.

Baselines:

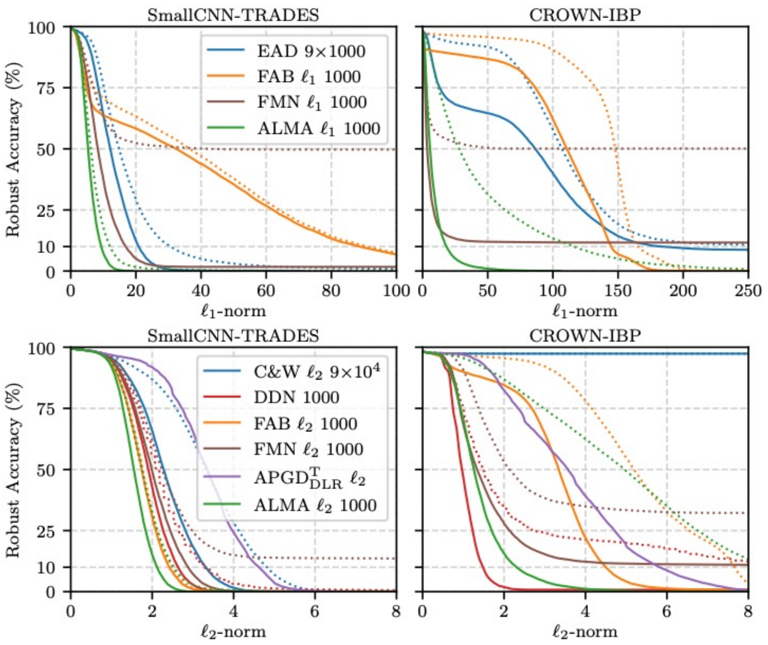
l_1 attacks: EAD, FAB and FMN; l_2 attacks: C&W, DDN, FAB and FMN;

CIEDE2000 attacks: PerC-AL; LPIPS attacks: LPA; Other attacks: APGD, C&W type attack for CIEDE2000 and LPIPS.

Experimental Results

ALMA performs best w.r.t. ASR, perturbation geometric median size, complexity, robust accuracy. (MNIST dataset)

Distance	Attack	ASR (%)	Median Distance	Forwards / Backwards
ℓ_1-norm	EAD 9×100 [9]	97.18	21.94	807 / 407
	EAD 9×1000 [9]	97.76	19.19	5 025 / 2 516
	FAB ℓ_1 100 [12]	99.80	27.41	201 / 1 000
	FAB ℓ_1 1000 [12]	99.83	24.21	2 001 / 10 000
	FMN ℓ_1 100 [25]	69.87	–	100 / 100
	FMN ℓ_1 1000 [25]	95.35	7.34	1 000 / 1 000
	ALMA ℓ_1 100	99.90	11.45	100 / 100
	ALMA ℓ_1 1000	100	7.16	1 000 / 1 000
ℓ_2-norm	C&W ℓ_2 9×1000 [7]	40.19	–	8 643 / 8 643
	C&W ℓ_2 9×10 000 [7]	40.20	–	85 907 / 85 907
	DDN 100 [26]	98.48	1.86	100 / 100
	DDN 1000 [26]	99.83	1.61	1 000 / 1 000
	FAB ℓ_2 100 [12]	83.25	2.10	201 / 1 000
	FAB ℓ_2 1000 [12]	96.28	1.77	2 001 / 10 000
	FMN ℓ_2 100 [25]	70.99	3.18	100 / 100
	FMN ℓ_2 1000 [25]	96.02	1.77	1 000 / 1 000
	APGD$^{\mathrm{T}}_{\mathrm{DLR}}$ ℓ_2‡ [13]	99.98	2.52	12 271 / 12 253
	ALMA ℓ_2 100	99.72	2.38	100 / 100
	ALMA ℓ_2 1000	100	1.61	1 000 / 1 000



Experimental Results

ALMA performs best, similar conclusion on CIFAR10 and ImageNet.

Distance	Attack	CIFAR10 ASR (%)	CIFAR10 Median Distance	CIFAR10 Forwards / Backwards	ImageNet ASR (%)	ImageNet Median Distance	ImageNet Forwards / Backwards
ℓ_1-norm	EAD 9×100 [9] (AAAI'17)	100	6.11	572 / 290	100	13.87	488 / 248
	EAD 9×1000 [9] (AAAI'17)	100	5.44	4 284 / 2 146	100	12.83	3 758 / 1 883
	FAB† ℓ_1 100 [12] (ICML'20)	96.58	4.26	201 / 1 000	88.82	10.72	1 810 / 900
	FAB† ℓ_1 1000 [12] (ICML'20)	99.00	3.78	2 001 / 10 000	89.07	8.88	18 010 / 9 000
	FMN ℓ_1 100 [25]	99.90	3.64	100 / 100	94.33	8.43	100 / 100
	FMN ℓ_1 1000 [25]	99.83	3.54	1 000 / 1 000	93.93	7.58	1 000 / 1 000
	ALMA ℓ_1 100	100	4.31	100 / 100	100	19.79	100 / 100
	ALMA ℓ_1 1000	100	3.69	1 000 / 1 000	100	12.10	1 000 / 1 000
ℓ_2-norm	C&W ℓ_2 9×1000 [7] (SP'17)	100	0.40	7 976 / 7 974	99.83	0.57	7 248 / 7 246
	C&W ℓ_2 9×10 000 [7] (SP'17)	100	0.40	78 081 / 78 079	99.83	0.57	67 479 / 67 476
	DDN 100 [26] (CVPR'19)	100	0.43	100 / 100	99.70	0.51	100 / 100
	DDN 1000 [26] (CVPR'19)	100	0.42	1 000 / 1 000	99.87	0.50	1 000 / 1 000
	FAB† ℓ_2 100 [12] (ICML'20)	100	0.41	201 / 1 000	99.70	0.35	1 810 / 900
	FAB† ℓ_2 1000 [12] (ICML'20)	100	0.41	2 001 / 10 000	98.90	0.35	18 010 / 9 000
	FMN ℓ_2 100 [25]	99.90	0.43	100 / 100	99.43	0.38	100 / 100
	FMN ℓ_2 1000 [25]	99.83	0.40	1 000 / 1 000	99.63	0.36	1 000 / 1 000
	$\text{APGD}^{\text{T}}_{\text{DLR}}$ ℓ_2‡ [13] (ICML'20)	100	0.38	5 345 / 5 321	100	0.34	6 096 / 6 068
	ALMA ℓ_2 100	100	0.40	100 / 100	100	0.38	100 / 100
	ALMA ℓ_2 1000	100	0.38	1 000 / 1 000	100	0.35	1 000 / 1 000
CIEDE 2000	C&W CIEDE2000 9×1000	100	0.93	6 729 / 6 726	100	1.39	5 635 / 5 632
	PerC-AL 100 [37] (CVPR'20)	100	2.87	201 / 100	99.90	3.55	201 / 100
	PerC-AL 1000 [37] (CVPR'20)	100	2.72	2 001 / 1 000	99.93	3.42	2 001 / 1 000
	ALMA CIEDE2000 100	100	1.09	100 / 100	100	0.75	100 / 100
	ALMA CIEDE2000 1000	100	0.78	1 000 / 1 000	100	0.63	1 000 / 1 000
LPIPS ×10−2	C&W LPIPS 9×1000	100	0.47	6 658 / 6 655	100	2.07	4 950 / 4 944
	LPA‡ [21] (ICLR'21)	100	5.39	1 118 / 1 108	100	5.79	1 211 / 1 201
	ALMA LPIPS 100	99.97	2.47	100 / 100	100	1.59	100 / 100
	ALMA LPIPS 1000	100	0.60	1 000 / 1 000	100	1.13	1000 / 1000

Thank you

LEHIGH UNIVERSITY