Efficient local linearity regularization to overcome catastrophic overfitting

Elias Abad Rocamora[1], Fanghui Liu[2], Grigorios G. Chrysos[3], Pablo M. Olmos[4], Volkan Cevher[1]

Vienna 05/2024

Background: Adversarial Training (AT)

Considering a classifier $\boldsymbol{f_\theta}$ parametrized by $\boldsymbol{\theta}$ and a distribution $\mathcal{D}$ on inputs $\boldsymbol{x}$ and labels y.

- We are interested in fast Adversarial Training (AT) (Madry et al., 2018) :

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{||\boldsymbol{\delta}||_\infty \leq \epsilon} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}+\boldsymbol{\delta}), y) \right]$$

Madry et al., Towards Deep Learning Models Resistant to Adversarial Attacks *ICLR, 2018*.

Background: Adversarial Training (AT)

Considering a classifier $\boldsymbol{f_\theta}$ parametrized by $\boldsymbol{\theta}$ and a distribution $\mathcal{D}$ on inputs $\boldsymbol{x}$ and labels y.

- We are interested in fast Adversarial Training (AT) (Madry et al., 2018) :

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{||\boldsymbol{\delta}||_\infty \leq \epsilon} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}+\boldsymbol{\delta}), y) \right]$$

- Solving the inner $\max$ slows down training:

$$\boldsymbol{\delta}_{t+1} = \mathop{\text{proj}}_{||\boldsymbol{\delta}||_\infty \leq \epsilon} \left[\boldsymbol{\delta}_{t+1} + \gamma \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}+\boldsymbol{\delta}_t), y)\right)\right]$$

10 PGD steps are tipically employed $\Rightarrow \times 10$ overhead! 😵‍💫

Madry et al., Towards Deep Learning Models Resistant to Adversarial Attacks *ICLR, 2018*.

Background: Catastrophic Overfitting (CO)

- As a cheap alternative, 1 PGD step (FGSM) can be used:

$$\boldsymbol{\delta}_{\mathrm{FGSM}} = \epsilon \cdot \operatorname{sign}\left(\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}), y)\right)$$

Wong et al., Fast is better than free: Revisiting adversarial training *ICLR, 2020.*

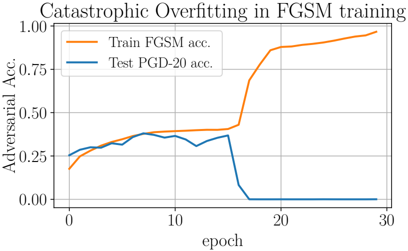
Andriuschenko and Flammarion, Understanding and Improving Fast Adversarial Training *NeurIPS, 2020.*

Background: Catastrophic Overfitting (CO)

- As a cheap alternative, 1 PGD step (FGSM) can be used:

$$\boldsymbol{\delta}_{\mathsf{FGSM}} = \epsilon \cdot \mathsf{sign}\left(\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y)\right)$$

- Sadly, Catastrophic Overfitting (CO) appears:
  - Example: PreActResNet18 on CIFAR10 at $\epsilon = 8/255$.
  - Training results 100% robust to FGSM attacks and 0% robust to PGD-20 attacks 😱



Wong et al., Fast is better than free: Revisiting adversarial training *ICLR, 2020*.
Andriuschenko and Flammarion, Understanding and Improving Fast Adversarial Training *NeurIPS, 2020*.

Background: Local linearity regularization

- Local linearity regularization can overcome CO. Let $\boldsymbol{\eta} \sim \mathrm{Unif}(-\epsilon, \epsilon)$:

Background: Local linearity regularization

- Local linearity regularization can overcome CO. Let $\boldsymbol{\eta} \sim \text{Unif}(-\epsilon, \epsilon)$:
- GradAlign:

$$1 - \text{cos-sim}\left[\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y),\ \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x} + \boldsymbol{\eta}), y)\right]$$

Andriuschenko and Flammarion, Understanding and Improving Fast Adversarial Training *NeurIPS, 2020*.

- CURE: $$||\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x} + \boldsymbol{\eta}), y)||_2$$

Moosavi-Dezfooli et al., Robustness via Curvature Regularization, and Vice Versa *CVPR, 2019*.

- LLR:

$$\left|\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x} + \boldsymbol{\eta}), y) - \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y) + \boldsymbol{\eta}^\top \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y)\right|$$

Qin et al., Adversarial Robustness through Local Linearization *NeurIPS, 2019*.

Background: Local linearity regularization

- Local linearity regularization can overcome CO. Let $\boldsymbol{\eta} \sim \text{Unif}(-\epsilon, \epsilon)$:
- GradAlign:

$$1 - \text{cos-sim}\left[\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y),\ \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x} + \boldsymbol{\eta}), y)\right]$$

Andriuschenko and Flammarion, Understanding and Improving Fast Adversarial Training *NeurIPS, 2020*.

- CURE:

$$||\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x} + \boldsymbol{\eta}), y)||_2$$

Moosavi-Dezfooli et al., Robustness via Curvature Regularization, and Vice Versa *CVPR, 2019*.

- LLR:

$$\left|\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x} + \boldsymbol{\eta}), y) - \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y) + \boldsymbol{\eta}^\top \nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y)\right|$$

Qin et al., Adversarial Robustness through Local Linearization *NeurIPS, 2019*.

- The problem: Differentiating gradients $(\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{x}} \cdot)$ is costly.

Background: Local linearity regularization

- Local linearity regularization can overcome CO. Let $\boldsymbol{\eta} \sim \text{Unif}(-\epsilon, \epsilon)$:
- GradAlign:

$$1 - \text{cos-sim}\left[\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y),\ \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}+\boldsymbol{\eta}), y)\right]$$

Andriuschenko and Flammarion, Understanding and Improving Fast Adversarial Training *NeurIPS, 2020*.

- CURE:

$$||\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y) - \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}+\boldsymbol{\eta}), y)||_2$$

Moosavi-Dezfooli et al., Robustness via Curvature Regularization, and Vice Versa *CVPR, 2019*.

- LLR:

$$\left|\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}+\boldsymbol{\eta}), y) - \mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y) + \boldsymbol{\eta}^\top \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{f_\theta}(\boldsymbol{x}), y)\right|$$

Qin et al., Adversarial Robustness through Local Linearization *NeurIPS, 2019*.

- The problem: Differentiating gradients ($\nabla_{\boldsymbol{\theta}}\nabla_{\boldsymbol{x}}\cdot$) is costly.
- The challenge: *Can we efficiently regularize local linearity?*

Our Method: ELLE

- If a function h is locally linear in a convex set $\mathcal{X}$, it must safisfy:

$$h((1-\alpha)\cdot\boldsymbol{x}_a+\alpha\cdot\boldsymbol{x}_b) = (1-\alpha)\cdot h(\boldsymbol{x}_a)+\alpha\cdot h(\boldsymbol{x}_b), \quad \forall\alpha \in [0,1], \forall\boldsymbol{x}_a, \boldsymbol{x}_b \in \mathcal{X}$$

Our Method: ELLE

- If a function h is locally linear in a convex set $\mathcal{X}$, it must safisfy:

$$h((1-\alpha)\cdot\boldsymbol{x}_a+\alpha\cdot\boldsymbol{x}_b) = (1-\alpha)\cdot h(\boldsymbol{x}_a)+\alpha\cdot h(\boldsymbol{x}_b), \quad \forall\alpha \in [0,1], \forall\boldsymbol{x}_a, \boldsymbol{x}_b \in \mathcal{X}$$

- ELLE: Let $\boldsymbol{x}_a, \boldsymbol{x}_b \sim \text{Unif}\left(\boldsymbol{x}+\boldsymbol{\delta} : ||\boldsymbol{\delta}||_\infty \leq \epsilon\right)$, $\alpha \sim \text{Unif}([0,1])$ and $\boldsymbol{x}_c = (1-\alpha)\cdot\boldsymbol{x}_a + \alpha\cdot\boldsymbol{x}_b$:

$$\left[(1-\alpha)\cdot\mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_a), y) + \alpha\cdot\mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_b), y) - \mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_c), y)\right]^2$$

Our Method: ELLE

- If a function h is locally linear in a convex set $\mathcal{X}$, it must safisfy:

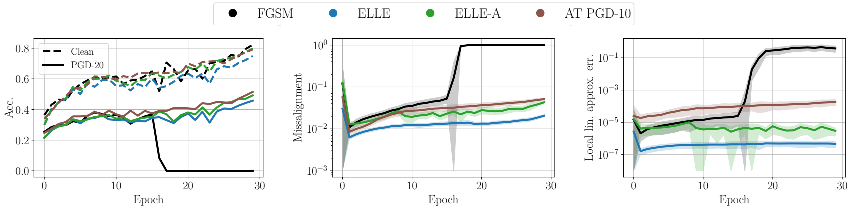
$$h((1-\alpha)\cdot\boldsymbol{x}_a+\alpha\cdot\boldsymbol{x}_b) = (1-\alpha)\cdot h(\boldsymbol{x}_a)+\alpha\cdot h(\boldsymbol{x}_b), \quad \forall\alpha \in [0,1], \forall\boldsymbol{x}_a, \boldsymbol{x}_b \in \mathcal{X}$$

- ELLE: Let $\boldsymbol{x}_a, \boldsymbol{x}_b \sim$ Unif $(\boldsymbol{x}+\boldsymbol{\delta} : ||\boldsymbol{\delta}||_\infty \leq \epsilon)$, $\alpha \sim$ Unif$([0,1])$ and $\boldsymbol{x}_c = (1-\alpha)\cdot\boldsymbol{x}_a + \alpha\cdot\boldsymbol{x}_b$:

$$\left[(1-\alpha)\cdot\mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_a), y) + \alpha\cdot\mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_b), y) - \mathcal{L}(\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_c), y)\right]^2$$

- Our regularization term does not involve differentiating gradients 🤩.

Our Method: ELLE



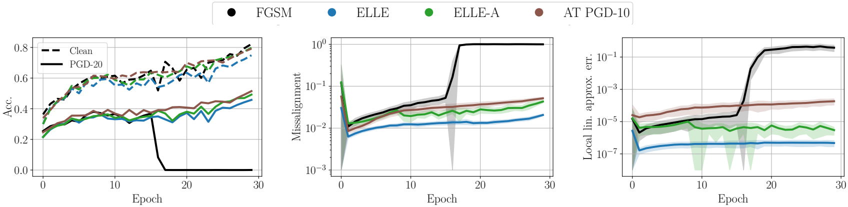
Test accuracy

GradAlign (log-scale)

ELLE (log-scale)

Figure: PreActResNet in CIFAR10 at $\epsilon = 8/255$

Our Method: ELLE

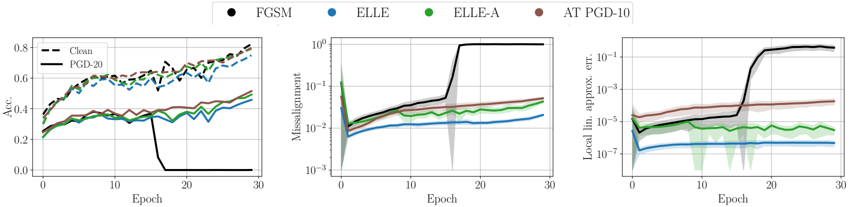


Test accuracy GradAlign (log-scale) ELLE (log-scale)

Figure: PreActResNet in CIFAR10 at $\epsilon = 8/255$

- Our local linearity metric follows the one of GradAlign at a reduced cost.

Our Method: ELLE

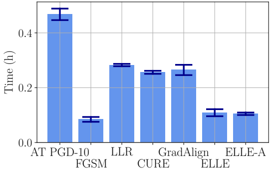


Test accuracy | GradAlign (log-scale) | ELLE (log-scale)

Figure: PreActResNet in CIFAR10 at $\epsilon = 8/255$

- Our local linearity metric follows the one of GradAlign at a reduced cost.
- When regularized (● and ●), CO is overcome.

Comparison with other regularization terms

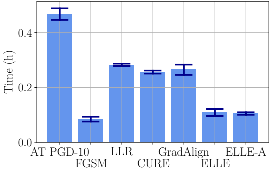


Runtime comparison

ϵ	8		16	
Method	AA	Clean	AA	Clean
LLR	$42.18 \pm (0.20)$	$75.02 \pm (0.09)$	$16.92 \pm (0.20)$	$42.81 \pm (9.62)$
CURE	$43.60 \pm (0.17)$	$77.74 \pm (0.11)$	$\underline{18.25} \pm (0.45)$	$52.49 \pm (0.04)$
GradAlign	$\mathbf{44.66} \pm (0.21)$	$\mathbf{80.50} \pm (0.07)$	$17.46 \pm (1.71)$	$44.35 \pm (15.32)$
ELLE	$42.78 \pm (0.95)$	$\underline{80.13} \pm (0.32)$	$\mathbf{18.28} \pm (0.17)$	$\mathbf{59.73} \pm (0.16)$
ELLE-A	$\underline{44.32} \pm (0.04)$	$79.81 \pm (0.10)$	$18.03 \pm (0.15)$	$\underline{59.21} \pm (1.23)$
AT PGD-10	$46.95 \pm (0.11)$	$79.11 \pm (0.08)$	$24.77 \pm (0.26)$	$59.64 \pm (0.46)$

PreActResNet18 in CIFAR10

Comparison with other regularization terms



ϵ	8		16	
Method	AA	Clean	AA	Clean
LLR	$42.18 \pm (0.20)$	$75.02 \pm (0.09)$	$16.92 \pm (0.20)$	$42.81 \pm (9.62)$
CURE	$43.60 \pm (0.17)$	$77.74 \pm (0.11)$	$\underline{18.25} \pm (0.45)$	$52.49 \pm (0.04)$
GradAlign	$\mathbf{44.66} \pm (0.21)$	$\mathbf{80.50} \pm (0.07)$	$17.46 \pm (1.71)$	$44.35 \pm (15.32)$
ELLE	$42.78 \pm (0.95)$	$\underline{80.13} \pm (0.32)$	$\mathbf{18.28} \pm (0.17)$	$\mathbf{59.73} \pm (0.16)$
ELLE-A	$\underline{44.32} \pm (0.04)$	$79.81 \pm (0.10)$	$18.03 \pm (0.15)$	$\underline{59.21} \pm (1.23)$
AT PGD-10	$46.95 \pm (0.11)$	$79.11 \pm (0.08)$	$24.77 \pm (0.26)$	$59.64 \pm (0.46)$

Runtime comparison

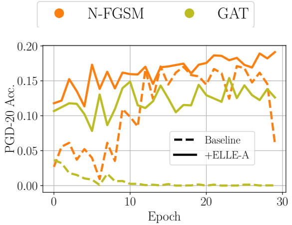
PreActResNet18 in CIFAR10

- ELLE attains comparable performance with negligible overhead.

Combination with other methods

ϵ		16		26	
Method	Model	AA	Clean	AA	Clean
GAT	PRN	$0.54 \pm (0.53)$	$78.52 \pm (0.25)$	$0.01 \pm (0.00)$	$84.35 \pm (0.34)$
GAT+ELLE-A	PRN	$13.83 \pm (4.63)$	$65.71 \pm (2.48)$	$6.56 \pm (2.79)$	$58.37 \pm (2.39)$
N-FGSM	PRN	$\mathbf{20.59} \pm (0.21)$	$61.24 \pm (0.26)$	$\underline{10.96} \pm (0.26)$	$37.73 \pm (0.32)$
N-FGSM+ELLE-A	PRN	$\underline{20.48} \pm (0.57)$	$61.21 \pm (0.14)$	$\mathbf{12.03} \pm (1.02)$	$26.77 \pm (3.25)$
AT PGD-10		$24.77 \pm (0.26)$	$59.64 \pm (0.46)$	$14.42 \pm (0.00)$	$34.90 \pm (0.61)$
GAT	WRN	$0.95 \pm (0.09)$	$84.09 \pm (0.10)$	$0.00 \pm (0.00)$	$89.04 \pm (0.28)$
GAT+ELLE-A	WRN	$17.30 \pm (1.20)$	$64.27 \pm (3.56)$	$6.74 \pm (2.08)$	$42.04 \pm (10.52)$
N-FGSM	WRN	$\underline{20.54} \pm (0.18)$	$63.83 \pm (1.24)$	$3.31 \pm (2.58)$	$27.96 \pm (9.36)$
N-FGSM+ELLE-A	WRN	$\mathbf{21.28} \pm (0.07)$	$64.25 \pm (0.20)$	$\mathbf{12.22} \pm (0.25)$	$33.50 \pm (0.14)$
AT PGD-10		$26.77 \pm (0.28)$	$64.97 \pm (0.09)$	$14.61 \pm (0.10)$	$36.30 \pm (0.62)$

CIFAR10 *Short* schedule



WRN in CIFAR10 at $\epsilon = \frac{26}{255}$

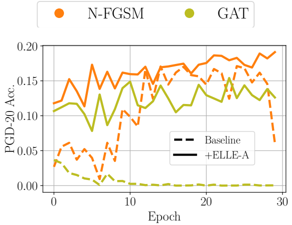
Sriramanan et al., Guided adversarial attack for evaluating and enhancing adversarial defenses *NeurIPS, 2020*.

de Jorge et al., Make some noise: Reliable and efficient single-step adversarial training *NeurIPS, 2022*.

Combination with other methods

ϵ		16		26	
Method	Model	AA	Clean	AA	Clean
GAT	PRN	0.54 ± (0.53)	78.52 ± (0.25)	0.01 ± (0.00)	84.35 ± (0.34)
GAT+ELLE-A	PRN	13.83 ± (4.63)	65.71 ± (2.48)	6.56 ± (2.79)	58.37 ± (2.39)
N-FGSM	PRN	**20.59** ± (0.21)	61.24 ± (0.26)	10.96 ± (0.26)	37.73 ± (0.32)
N-FGSM+ELLE-A	PRN	20.48 ± (0.57)	61.21 ± (0.14)	**12.03** ± (1.02)	26.77 ± (3.25)
AT PGD-10		24.77 ± (0.26)	59.64 ± (0.46)	14.42 ± (0.00)	34.90 ± (0.61)
GAT	WRN	0.95 ± (0.09)	84.09 ± (0.10)	0.00 ± (0.00)	89.04 ± (0.28)
GAT+ELLE-A	WRN	17.30 ± (1.20)	64.27 ± (3.56)	6.74 ± (2.08)	42.04 ± (10.52)
N-FGSM	WRN	20.54 ± (0.18)	63.83 ± (1.24)	3.31 ± (2.58)	27.96 ± (9.36)
N-FGSM+ELLE-A	WRN	**21.28** ± (0.07)	64.25 ± (0.20)	**12.22** ± (0.25)	33.50 ± (0.14)
AT PGD-10		26.77 ± (0.28)	64.97 ± (0.09)	14.61 ± (0.10)	36.30 ± (0.62)

CIFAR10 *Short* schedule



WRN in CIFAR10 at $\epsilon = \frac{26}{255}$

- Plugging our regularization term stabilizes and improves the performance of popular single-step AT methods.

Sriramanan et al., Guided adversarial attack for evaluating and enhancing adversarial defenses *NeurIPS, 2020*.

de Jorge et al., Make some noise: Reliable and efficient single-step adversarial training *NeurIPS, 2022*.

Thanks

Thanks for your attention!

contact: `elias.abadrocamora@epfl.ch`

supported by: