Adaptive Adversarial Norm Space for Efficient Adversarial Training

UNIVERSITY OF OULU

Hui Kuurila-Zhang, Haoyu Chen and Guoying Zhao
Center for Machine Vision and Signal Analysis (CMVS)
University of Oulu, Oulu, Finland

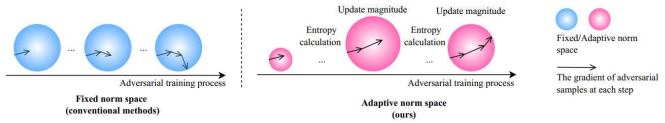

Figure 1: Adaptive entropy-guided adversarial norm space during adversarial training.

Motivation

- Adaptive adversarial training strategies have better performance.
- The existing adaptive adversarial training methods are computationally expensive.
- Is it possible to design an adaptive adversarial training algorithm that retains the good performance but not computationally expensive?

Problem definition

Adversarial training:

$$\min_{w} \frac{1}{n} \sum_{i=1}^{n} \max_{||x_i' - x_i||_p \leq \varepsilon} l(f_w(x_i'), y_i), \tag{1}$$

Entropy-Guided Cyclical Adversarial Strategy (ECAS)

- **Cyclically changeable adversarial norm space**

$$\varepsilon_n = \frac{\varepsilon_{\max} - \varepsilon_{\min}}{\alpha} \times \mathrm{mod}(\frac{n}{\alpha}) + \varepsilon_{\min}, \tag{2}$$

- **Entropy-guided constraints on norm space**

Algorithm 1 ECAS

Input: Current epoch *epoch*, epsilon space *epsilon_space*, entropy of a batch of inputs *H*, lower entropy *h_low*, higher entropy *h_high*
Output: Adversarial strength for this batch at the current epoch *epsilon_batch*

function ECAS(*epoch*, *epsilon_space*, *H*, *h_low*, *h_high*):
  big_epsilons ← The three biggest epsilons from *epsilon_space*
  epsilon_batch ← Initialize the epsilons with ones
  ε ← CYCLICAL_EPSILON(*epoch*, *epsilon_space*)
  epsilon_batch ← *epsilon_batch* × ε
  epsilon_batch[*H* > *h_high*] ← The smallest ε from *epsilon_space*
  epsilon_batch[*H* < *h_low*] ← Take a random item from (*big_epsilons*)
  return *epsilon_batch*

Optimization

ECAS generates a tailored magnitude (ε_i) of the norm space for each sample i. The adversarial example is then generated by Eq. (4).

$$x_i' = \max_{||x_i' - x_i||_p \leq \varepsilon_i} l(f_w(x_i'), y_i), \tag{4}$$

The parameter of the network $f_w(\cdot)$ is updated by Eq.(5).

$$w \leftarrow w - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla_w l(f_w(x_i'), y_i), \tag{5}$$

Experiments

- **Verification of different schedulers and contribution of each component**

Scheduler	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
Baseline (Fixed)	0.8517	0.5607	0.5508	0.5488	0.5391	0.5169
+ Linear	0.8351	0.5704	0.5587	0.5548	0.5497	0.5273
+ Cyclic (batch)	0.8682	0.5648	0.5482	0.5451	0.5495	0.5243
+ Cyclic	**0.8634**	**0.5766**	**0.5670**	**0.5637**	**0.5567**	**0.5325**
+ Cyclic + entropy	**0.8632**	**0.5780**	**0.5680**	**0.5648**	**0.5589**	**0.5343**

Table 1: Performance comparison of models (WRN34-10) trained by PGD-AT [14] with different schedulers on CIFAR-10. Results in bold are from our methods.

- **Running time analysis**

	CIFAR-10			CIFAR-100		
Framework	PGD-AT	TRADES	AWP	PGD-AT	TRADES	AWP
+L: runtime	681.0s	6048.4s	5587.6s	428.3s	6051.4s	1081.2s
+Ours: runtime	**89.4s**	**52.0s**	**59.7s**	**150.5s**	**166.0s**	**149.3s**
+L: clean	0.8623	0.8524	0.8774	0.6180	0.6062	0.6489
+Ours: clean	0.8632	0.8399	0.8817	0.6116	0.5869	0.6477
+L: robust	0.5358	0.5415	0.5552	0.2903	0.2812	0.3077
+Ours: robust	0.5343	0.5202	0.5454	0.2883	0.2824	0.2978

Table 2: Comparison of the extra running time (1st and 2nd rows) (tested on CIFAR-10 and CIFAR-100 with WRN34-10) when integrating the LAS-AT ("+L:" in the table) or ECAS ("+Ours:") to the baseline methods, and their clean accuracy (3rd and 4th rows) and robust accuracy (5th and 6th rows).

- **Comparisons with SOTA methods**

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
PGD-AT [14]	0.8517	0.5607	0.5508	0.5488	0.5391	0.5169
TRADES [23]	0.8572	0.5675	0.5610	0.5590	0.5387	0.5340
FAT [24]	0.8797	0.5031	0.4986	0.4879	0.4865	0.4748
MART [18]	0.8417	0.5898	0.5856	0.5806	0.5458	0.5110
AWP [20]	0.8557	0.5892	0.5813	0.5792	0.5603	0.5390
LAS-AWP [9]	0.8774	**0.6109**	**0.6016**	**0.5979**	**0.5822**	**0.5552**
ECAS-AWP (ours)	**0.8817**	0.6038	0.5910	0.5875	0.5750	0.5454

Table 4: Test result on CIFAR-10 with WRN34-10, the best performance is shown in bold, and the second best is marked underlined

- **Visualization of the norm space learned by ECAS**

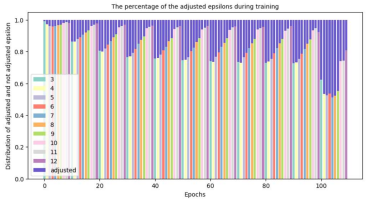

Figure 2: The percentage distribution of the adjusted and not adjusted magnitude of the norm space in ECAS-PGD-AT during training on the CIFAR-10 dataset

Conclusion

- We empirically identify that cyclically changing the adversarial norm space can improve the robustness of the network
- We propose a simple yet effective entropy-guided cyclical adversarial strategy (ECAS) to periodically adjust the norm space of adversarial examples.
- The proposed method ECAS is easy to integrate with other adversarial training methods and improves their performance to a comparable level as the SOTA method without adding too much extra cost to the computations as the SOTA method does.

Oulu Capital of Northern Scandinavia

Infotech Oulu

BMVC 2023

Science with Arctic Attitude