On the Loss Landscape of Adversarial Training: Identifying Challenges and How to Overcome Them

Chen Liu†, Mathieu Salzmann†, Tao Lin†, Ryota Tomioka‡, Sabine Süsstrunk†
† EPFL, ‡ Microsoft Research

NEURAL INFORMATION PROCESSING SYSTEMS

Background

Given samples $\{\mathbf{x}_i\}_{i=0}^N$, an adversarial budget $\mathcal{S}_\epsilon = \{\delta | \|\delta\|_\infty \leq \epsilon\}$ and the loss objective g of a classifier parameterized by θ, adversarial training is solving the min-max problem below:

$$\min_\theta \mathcal{L}_\epsilon(\theta) := \frac{1}{N}\sum_{i=1}^{N} g_\epsilon(\mathbf{x}_i, \theta)$$
$$g_\epsilon(\mathbf{x}_i, \theta) := \max_{\delta \in \mathcal{S}_\epsilon} g(\mathbf{x}_i + \delta, \theta) \ . \tag{1}$$

Compared with non-adversarial training, adversarial training has: 1) **slower convergence** 2) **larger generalization gap**

Linear model

For linear model, the trainable parameter is a matrix $\mathbf{W}$. Under different values of ϵ, we prove that:

- The set of "robust parameters", which makes the model robust against attacks, shrinks with the increase of ϵ.
- When ϵ is large enough, the optimal parameter of problem (1) is $\mathbf{W} = \mathbf{0}$. The model is a constant classifier.

For deep nonlinear networks, similar phenomena are observed: we obtain a constant classifier from adversarial training when ϵ is large and the model is not over-parameterized.

Code on GitHub:

github.com/liuchen11/AdversaryLossLandscape

Nonlinear Model

Assume the Lipschitzian continuous of g:

$$\|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\| \ ,$$
$$\|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\| \ ,$$
$$\|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| \leq L_{\theta\mathbf{x}} \|\mathbf{x}_1 - \mathbf{x}_2\|_\infty \ . \tag{2}$$

Then we obtain:

$$\|\mathcal{L}_\epsilon(\theta_1) - \mathcal{L}_\epsilon(\theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\| \ ,$$
$$\|\nabla_\theta \mathcal{L}_\epsilon(\theta_1) - \nabla_\theta \mathcal{L}_\epsilon(\theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\| + 2\epsilon L_{\theta\mathbf{x}} \ . \tag{3}$$

From the vanilla loss landscape to the adversarial loss landscape, continuity is preserved but smoothness is not. Non-smoothness arises from the dependence of the adversarial perturbation δ on the model parameters θ. Arbitrarily small changes in θ can lead to large changes in δ.

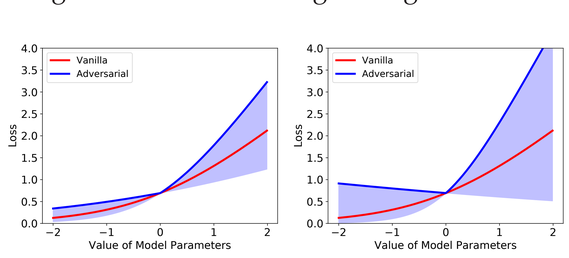

Figure 1: A plot of the vanilla (red lines) and the adversarial loss landscape (blue lines) when $g(x, \theta) = \log(1 + \exp(\theta x))$. Left: $\epsilon = 0.6$; Right: $\epsilon = 1.2$.

The non-smooth nature of the adversarial loss landscape is unfavorable for both optimization and generalization.

- For optimization, there is no longer a guarantee that SGD converges to a critical point, making training less stable.
- For generalization, the minima of $\mathcal{L}_\epsilon$ on θ become sharper with the increase of ϵ.

A warmup of ϵ in adversarial training enables us to start with an "easy" loss landscape and then gradually switch to the target loss landscape. It helps in 1) making the training less sensitive to the learning rate, and in 2) improving the final performance. Linear and cosine schedulers are two classical warmup scheduling schemes.

Selected Experiments

Sub-figures are labeled in the alphabet order from the left to the right.

Gradient Analysis

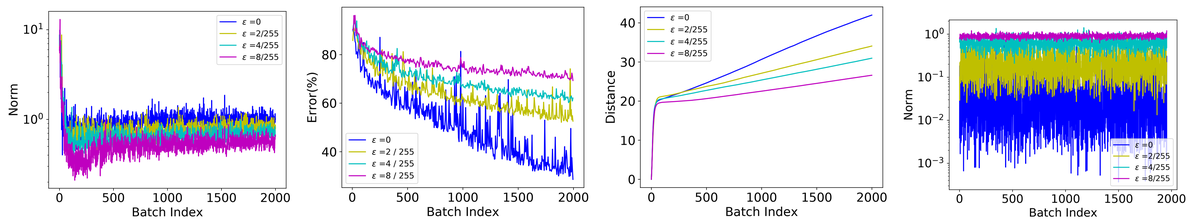

Figure 2: Adversarial training of ResNet18 on CIFAR10 with different values of ϵ. (a - c) show the first 2000 mini-batches while (d) shows the last 2000 mini-batches. (a) and (d) illustrate the gradient's norm; (b) shows the test error and (c) shows the distance the model parameters have moved from initialization.

Hessian Analysis

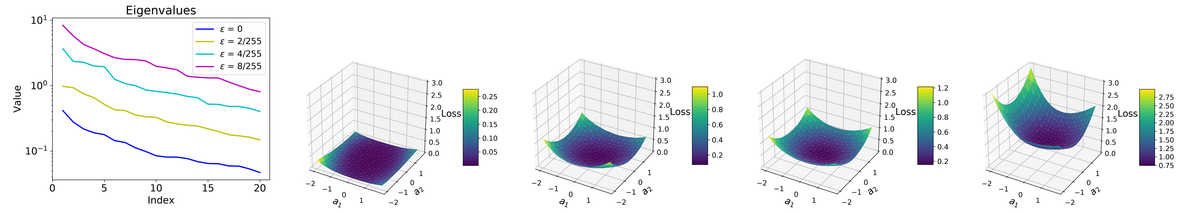

Figure 3: Adversarial training of ResNet18 on CIFAR10 with different values of ϵ. (a) Numerical estimation of the top 20 eigenvalues of the Hessian matrix $\nabla_\theta^2 \mathcal{L}_\epsilon$. (b - e) Landscape visualization in the directions of the top 2 Hessian eigenvectors. $\epsilon = 0$, $\epsilon = 2/255$, $\epsilon = 4/255$, and $\epsilon = 8/255$ from the left to the right, respectively.

Warmup of ϵ

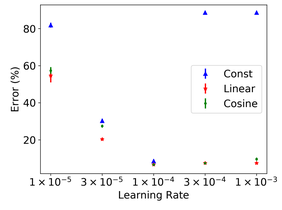

Figure 4: Mean and std of the test error under different learning rates and ϵ schedulers on MNIST when $\epsilon = 0.4$

ϵ Scheduler	Clean Error (%)	Robust Error (%)			
		PGD100 (%)	APGD100 CE (%)	APGD100 DLR (%)	Square5K (%)
Constant	18.62(6)	54.97(9)	57.26(13)	56.60(25)	50.59(19)
Cosine	18.43(26)	53.85(21)	56.16(18)	55.77(24)	49.60(18)
Linear	18.55(14)	53.41(10)	55.69(17)	55.45(22)	49.66(28)

Table 1: Comparison between different ϵ schedulers with ResNet18 on CIFAR10. The number between brackets indicates the standard deviation across different runs. For example, $1.56(17)$ stands for 1.56 ± 0.17. More results in the paper.

Takeaway Message

With the increase of the adversarial budget's size ϵ, the adversarial loss landscape in the model parameter space becomes **unfavorable for optimization**, including 1) **non-smoothness** 2) **less-connected minima** 3) **hardness to escape from the initial suboptimal region**. Using a warm-up ϵ scheduling scheme, like a linear or cosine scheduler, improves performance and makes adversarial training less sensitive to the learning rate.