Appendix: Reliably fast adversarial training via latent adversarial perturbation

Geon Yeong Park         Sang Wan Lee
KAIST
Daejeon, South Korea
(pky3436, sangwan)@kaist.ac.kr

This supplementary material is organized as follows. In section 1, we present the proof for Proposition 1, which is obtained by modifying [2], for completeness. Optimization setting and hyperparameter configurations are presented in section 2.

1. Proof

Proposition 1. Consider an L layer neural network, with the latent adversarial perturbations \( \delta_k(x) \) being applied at each layer \( k \in K \). Assuming the Hessians, of the form \( \nabla^2 h_l(x)|_{h_m(x)} \) where \( l, m \) are the index over layers, are finite. Then the perturbation accumulated at the layer \( L - 1 \), \( \delta_{L-1}(x) \), is approximated by:

\[
\delta_{L-1}(x) = \sum_{k \in K} J_k(x) \delta_k(x) + O(\gamma),
\]

where \( J_k(x) \in \mathbb{R}^{N_{L-1} \times N_k} \) represents each layer’s Jacobian; \( J_k(x)_{i,j} = \frac{\partial h_{L-1}(x)}{\partial x_k} \), given the number of neurons in layer \( L - 1 \) and \( k \) as \( N_{L-1} \) and \( N_k \), respectively. \( O(\gamma) \) represents higher order terms in \( \delta \) that tend to zero in the limit of small perturbation.

Proof. Starting with layer 0 as the input layer, the accumulated perturbation on a layer \( L - 1 \) can be approximated through recursion. Following the conventional adversarial training, suppose that the first layer index 0 is included in the set \( K \). At layer 0, we apply Taylor’s theorem on \( h_1(x + \delta_0(x)) \) around the original input \( x \). If we assume that all values in Hessian of \( h_1(x) \) is finite, i.e., \( |\partial^2 h_1(x)|/\partial x_\ell x_k| < \infty, \forall i, j, k \), the following approximation holds:

\[
h_1(x + \delta_0(x)) = h_1(x) + \frac{\partial h_1(x)}{\partial x} \delta_0(x) + O(\kappa_0),
\]

where \( O(\kappa_0) \) represents asymptotically dominated higher order terms given the small perturbation. By accommodating \( L = 2 \) as a special case, we obtain the accumulated noise \( \delta_{L-1} = \frac{\partial h_1(x)}{\partial x} \delta_0(x) + O(\kappa_0) \). Note that (2) can be generalized with an arbitrary layer index \( k + 1 \) and perturbation \( \delta_k(x) \).

Repeating this process for each layer \( k \in K \) recursively, and assuming that all Hessians of the form \( \nabla^2 h_l(x)|_{h_m(x)} \) for \( m < l \) are finite, we obtain the accumulated perturbation for a layer \( L - 1 \) as follows:

\[
\delta_{L-1}(x) = \sum_{k \in K} \frac{\partial h_{L-1}(x)}{\partial h_k(x)} \delta_k(x) + O(\gamma),
\]

where \( O(\gamma) \) represents asymptotically dominated higher order terms as the perturbation \( \delta_k(x), \forall k \in K \) is sufficiently small. Denoting \( \nabla^2 h_{L-1}(x)/\partial x_k(x) \) as the Jacobian \( J_k(x) \in \mathbb{R}^{N_{L-1} \times N_k} \) completes the proof. \( \Box \)

2. Experimental setup

We provide extended details about simulation settings for completeness. For a fair comparison, we reproduced all the other baseline results using the same back-bone architecture and the optimization settings. Every method is trained for 30 epochs except Free-AT [3] which is trained for 72 epochs to get results comparable to the other methods. Following the setup of [1], we use cyclic learning rates [4] with the SGD optimizer with momentum 0.9 and weight decay \( 5 \times 10^{-4} \). Specifically, the learning rate increases linearly from 0 to 0.2 in first 12 epochs, and then decreases linearly to 0 in left 18 epochs. We use a batch size of 128 for CIFAR-10 and CIFAR-100 experiments. For the Tiny ImageNet experiments, we use a batch size of 64 to reduce the memory consumption.

For FGSM-RS [5] on every dataset, we use a step size \( \alpha = 1.25 \gamma_0 \) following the recommendation of authors. We succeeded in reproducing the robust accuracy of FGSM-RS against PGD-50-10 attack using the experimental setup reported in [5], but found that catastrophic overfitting occurs when the epoch was increased to 30. For PGD-7 AT, we use a step size \( \alpha = 2\gamma_0/10 \) for generating a 7-step PGD adversarial attack sample.

References


