INSTANCE ADAPTIVE ADVERSARIAL TRAINING: IMPROVED ACCURACY TRADEOFFS IN NEURAL NETS

Yogesh Balaji[1,2*] **Tom Goldstein** [1,2] **Judy Hoffman** [1,3]
[1]Facebook AI Research [2]University of Maryland [3]Georgia Institute of Technology

ABSTRACT

Adversarial training is by far the most successful strategy for improving robustness of neural networks to adversarial attacks. Despite its success as a defense mechanism, adversarial training fails to generalize well to unperturbed test set. We hypothesize that this poor generalization is a consequence of adversarial training with uniform perturbation radius around every training sample. Samples close to decision boundary can be morphed into a different class under a small perturbation budget, and enforcing large margins around these samples produce poor decision boundaries that generalize poorly. Motivated by this hypothesis, we propose instance adaptive adversarial training – a technique that enforces sample-specific perturbation margins around every training sample. We show that using our approach, test accuracy on unperturbed samples improve with a marginal drop in robustness. Extensive experiments on CIFAR-10, CIFAR-100 and Imagenet datasets demonstrate the effectiveness of our proposed approach.

1 INTRODUCTION

A key challenge when deploying neural networks in safety-critical applications is their poor stability to input perturbations. Extremely tiny perturbations to network inputs may be imperceptible to the human eye, and yet cause major changes to outputs. One of the most effective and widely used methods for hardening networks to small perturbations is "adversarial training" (Madry et al., 2018), in which a network is trained using adversarially perturbed samples with a fixed perturbation size. By doing so, adversarial training typically tries to enforce that the output of a neural network remains nearly constant within an ℓ_p ball of every training input.

Despite its ability to increase robustness, adversarial training suffers from poor accuracy on clean (natural) test inputs. The drop in clean accuracy can be as high as 10% on CIFAR-10, and 15% on Imagenet (Madry et al., 2018; Xie et al., 2019), making robust models undesirable in some industrial settings. The consistently poor performance of robust models on clean data has lead to the line of thought that there may be a fundamental trade-off between robustness and accuracy (Zhang et al., 2019; Tsipras et al., 2019), and recent theoretical results characterized this tradeoff (Fawzi et al., 2018; Shafahi et al., 2018; Mahloujifar et al., 2019).

In this work, we aim to understand and optimize the tradeoff between robustness and clean accuracy. More concretely, our objective is to improve the clean accuracy of adversarial training for a chosen level of adversarial robustness. Our method is inspired by the observation that the constraints enforced by adversarial training are *infeasible*; for commonly used values of ϵ, it is not possible to achieve label consistency within an ϵ-ball of each input image because the balls around images of different classes overlap. This is illustrated on the left of Figure 1, which shows that the ϵ-ball around a "bird" (from the CIFAR-10 training set) contains images of class "deer" (that do not appear in the training set). If adversarial training were successful at enforcing label stability in an $\epsilon = 8$ ball around the "bird" training image, doing so would come at the *unavoidable* cost of misclassifying the nearby "deer" images that come along at test time. At the same time, when training images lie far from the decision boundary (eg., the deer image on the right in Fig 1), it is possible to enforce stability with large ϵ with no compromise in clean accuracy. When adversarial training on CIFAR-10, we see that $\epsilon = 8$ is too large for some images, causing accuracy loss, while being unnecessarily small for others, leading to sub-optimal robustness.

*Work done during an internship at Facebook AI Research

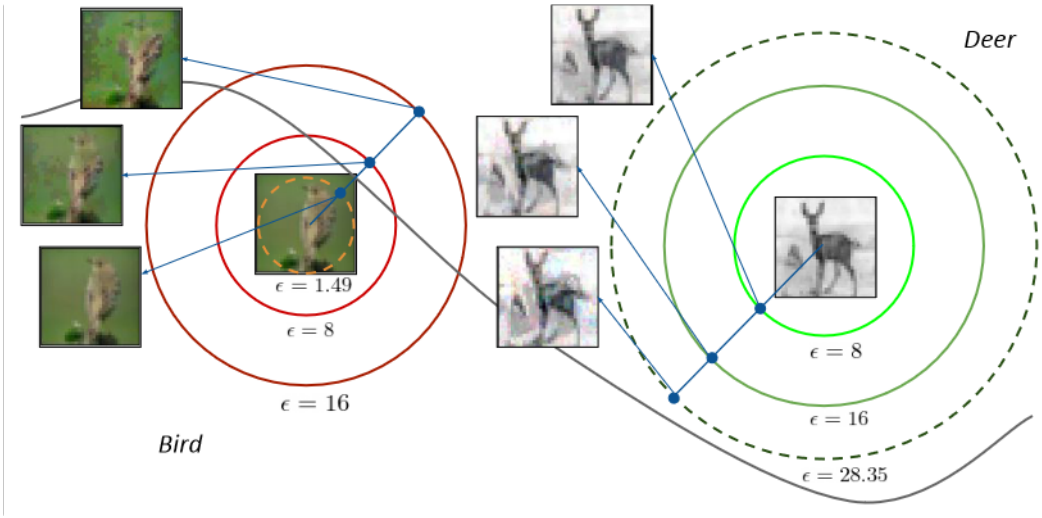

Figure 1: Overview of instance adaptive adversarial training. Samples close to the decision boundary (bird on the left) have nearby samples from a different class (deer) within a small L_p ball, making the constraints imposed by PGD-8 / PGD-16 adversarial training infeasible. Samples far from the decision boundary (deer on the right) can withstand large perturbations well beyond $\epsilon = 8$. Our adaptive adversarial training correctly assigns the perturbation radius (shown in dotted line) so that samples within each L_p ball maintain the same class.

The above observation naturally motivates adversarial training with *instance adaptive* perturbation radii that are customized to each training image. By choosing larger robustness radii at locations where class manifolds are far apart, and smaller radii at locations where class manifolds are close together, we get high adversarial robustness where possible while minimizing the clean accuracy loss that comes from enforcing overly-stringent constraints on images that lie near class boundaries. As a result, instance adaptive training significantly improves the tradeoff between accuracy and robustness, breaking through the pareto frontier achieved by standard adversarial training. Additionally, we show that the learned instance-specific perturbation radii are interpretable; samples with small radii are often ambiguous and have nearby images of another class, while images with large radii have unambiguous class labels that are difficult to manipulate.

Parallel to our work, we found that Ding et al. (2018) uses adaptive margins in a max-margin framework for adversarial training. Their work focuses on improving the adversarial robustness, which differs from our goal of understanding and improving the robustness-accuracy tradeoff. Moreover, our algorithm for choosing adaptive margins significantly differs from that of Ding et al. (2018).

2 Background

Adversarial attacks are data items containing small perturbations that cause misclassification in neural network classifiers (Szegedy et al., 2014). Popular methods for crafting attacks include the fast gradient sign method (FGSM) (Goodfellow et al., 2015) which is a one-step gradient attack, projected gradient descent (PGD) (Madry et al., 2018) which is a multi-step extension of FGSM, the C/W attack (Carlini & Wagner, 2017), DeepFool (Moosavi-Dezfooli et al., 2016), and many more. All these methods use the gradient of the loss function with respect to inputs to construct additive perturbations with a norm-constraint. Alternative attack metrics include spatial transformer attacks (Xiao et al., 2018), attacks based on Wasserstein distance in pixel space (Wong et al., 2019), etc.

Defending against adversarial attacks is a crucial problem in machine learning. Many early defenses (Buckman et al., 2018; Samangouei et al., 2018; Dhillon et al., 2018), were broken by strong attacks. Fortunately, *adversarially training* is one defense strategy that remains fairly resistant to most existing attacks.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote the set of training samples in the input dataset. In this paper, we focus on classification problems, hence, $y_i \in \{1, 2, \dots N_c\}$, where N_c denotes the number of classes. Let $f_\theta(\mathbf{x}) : \mathbb{R}^{c \times m \times n} \to \mathbb{R}^{N_c}$ denote a neural network model parameterized by θ. Classifiers are often trained by minimizing the cross entropy loss given by

$$\min_\theta \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \sim \mathcal{D}} -\tilde{\mathbf{y}}_i \big[\log(f_\theta(\mathbf{x}_i)) \big]$$

where $\tilde{\mathbf{y}}_i$ is the one-hot vector corresponding to the label y_i. In adversarial training, instead of optimizing the neural network over the clean training set, we use the adversarially perturbed training set. Mathematically, this can be written as the following *min-max* problem

$$\min_\theta \max_{\|\delta_i\|_\infty \leq \epsilon} \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \sim \mathcal{D}} -\tilde{\mathbf{y}}_i \big[\log(f_\theta(\mathbf{x}_i) + \delta_i) \big] \tag{1}$$

This problem is solved by an alternating stochastic method that takes minimization steps for θ, followed by maximization steps that approximately solve the inner problem using k steps of PGD. For more details, refer to Madry et al. (2018).

Algorithm 1 Adaptive adversarial training algorithm

Require: N_{iter}: Number of training iterations, N_{warm}: Warmup period
Require: $PGD_k(\mathbf{x}, y, \epsilon)$: Function to generate PGD-k adversarial samples with ϵ norm-bound
Require: ϵ_w: ϵ used in warmup
for t in $1 : N_{iter}$ **do**
  Sample a batch of training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{batch}} \sim \mathcal{D}$
  if $t < N_{warm}$ **then**
    $\epsilon_i = \epsilon_w$
  else
    Choose ϵ_i using Alg 2
  end if
  $\mathbf{x}_i^{adv} = PGD(\mathbf{x}_i, y_i, \epsilon_i)$
  $S_+ = \{i | f(\mathbf{x}_i)$ is correctly classified as $y_i\}$
  $S_- = \{i | f(\mathbf{x}_i)$ is incorrectly classified as $y_i\}$
  $\min_\theta \frac{1}{N_{batch}} \Big[\sum_{i \in S_+} L_{cls}(\mathbf{x}_i^{adv}, y^i) + \sum_{i \in S_-} L_{cls}(\mathbf{x}_i, y_i) \Big]$
end for

3 Instance Adaptive Adversarial Training

To remedy the shortcomings of uniform perturbation radius in adversarial training (Section 1), we propose *Instance Adaptive Adversarial Training* (IAAT), which solves the following optimization:

$$\min_\theta \max_{\|\delta_i\|_\infty < \epsilon_i} \frac{1}{N} \sum_{(\mathbf{x}_i, y_i) \sim \mathcal{D}} -\tilde{\mathbf{y}}_i \big[\log(f_\theta(\mathbf{x}_i) + \delta_i) \big] \tag{2}$$

Like vanilla adversarial training, we solve this by sampling mini-batches of images $\{\mathbf{x}_i\}$, crafting adversarial perturbations $\{\delta_i\}$ of size at most $\{\epsilon_i\}$, and then updating the network model using the perturbed images.

The proposed algorithm is distinctive in that it uses a different ϵ_i for each image $\mathbf{x}_i$. Ideally, we would choose each ϵ_i to be as large as possible without finding images of a different class within the ϵ_i-ball around $\mathbf{x}_i$. Since we have no a-priori knowledge of what this radius is, we use a simple heuristic to update ϵ_i after each epoch. After crafting a perturbation for $\mathbf{x}_i$, we check if the perturbed image was a successful adversarial example. If PGD succeeded in finding an image with a different class label, then ϵ_i is too big, so we replace $\epsilon_i \leftarrow \epsilon_i - \gamma$. If PGD failed, then we set $\epsilon_i \leftarrow \epsilon_i + \gamma$.

Since the network is randomly initialized at the start of training, random predictions are made, and this causes $\{\epsilon_i\}$ to shrink rapidly. For this reason, we begin with a warmup period of a few (usually 10 epochs for CIFAR-10/100) epochs where adversarial training is performed using uniform ϵ for every sample. After the warmup period ends, we perform instance adaptive adversarial training.

A detailed training algorithm is provided in Alg. 1.