Adversarial Training (Universal Adversarial Perturbation)

M. Parsa Dini [1] **Amin Kiani** [2] **M. Reze Rahmani** [1] **M. Hossein Yassaee** [1]
mp.dini@sharif.edu, mr.rahmani@sharif.edu,
amin.kiani@sharif.edu, yassaee@sharif.edu

Abstract

In this paper, we explore an attack method for a deep neural network classifier and propose a technique to enhance its robustness. We then modify the attack algorithm and compare its performance against state-of-the-art adversarial attack methods. Additionally, we leverage tools from high-dimensional probability to establish a theoretical bound for the proposed setting. We introduce an algorithm, propose our own method, and evaluate the model's accuracy through various measurements.

1. Introduction

Deep neural networks have been crucial in many tasks, particularly in relatively simple applications such as image classification. However, these classifiers are vulnerable to adversarial attacks, which can manipulate their predictions. Most attacks fall under the category of white-box attacks, where the adversary has access to the model's gradients and weights. Our goal is to find perturbed input vectors that, when fed into the model, lead to misclassification with high probability.

1.1. Literature/Formulation

Given an image space χ, an input image $X \in \mathbb{R}^D \subset \chi$ from an image dataset $\mathcal{D} = \{X^1, \dots, X^M\}$, and a label space $\mathcal{C} = \{1, \dots, k\}$, we define a classifier $f : \chi \to \mathcal{C}$ with accuracy at least $1 - \delta$. Let $y(X)$ denote the true label of the image X. In other words, we have:

$$\mathbb{P}[f(X) = y(X)] \geq 1 - \delta.$$

Furthermore, let $H(c)$ denote the binary representation of label c in Hamming space. We aim to bound the difference in Hamming distance between the label of an image, $f(X)$, and the label of its perturbed version, $f(X + t)$, where t is a perturbation vector. Additionally, let $d_{\mathcal{H}}(x, y)$ denote the Hamming distance between labels x and y and $\phi(c)$ is a function that maps the labels into its corresponding element in hamming space of labels.

2. Methodology

2.1. Randomized Smoothing in Adversarial Perturbation

Consider a classification problem from $\mathbb{R}^D$ to a set of classes $\mathcal{C}$. Randomized smoothing is a method for constructing a new, smoothed classifier $\hat{f}$ from an arbitrary base classifier f. When queried at X, the smoothed classifier $\hat{f}$ returns the class that the base classifier f is most likely to output when X is perturbed by isotropic Gaussian noise:

$$\hat{f}(X) = \arg\max_{c \in \mathcal{C}} P(f(X + t) = c)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. The noise level σ is a hyperparameter of the smoothed classifier $\hat{f}$ that controls the robustness/accuracy tradeoff.
Suppose that when the base classifier f classifies $\mathcal{N}(X, \sigma^2 I_D)$, the most probable class c_A is returned with probability $\overline{P_A}$, and the "runner-up" class is returned with probability $\underline{P_B}$. $\overline{P_A}$ is a lower bound for $\overline{P_A}$ and $\underline{P_B}$ is a lower bound for $\underline{P_B}$.

Theorem 2.1. *Let $f : \mathbb{R}^D \to \mathcal{C}$ be any deterministic or random function, and let $t \sim \mathcal{N}(0, \sigma^2 I_D)$. Let $\hat{f}$ be defined as above. Suppose $c_A \in \mathcal{C}$ and $\overline{P_A}, \underline{P_B} \in [0, 1]$ satisfy:*

$$P(f(X+t) = c_A) \geq \overline{P_A} \geq \underline{P_B} \geq \max_{c \neq c_A} P(f(X+t) = c).$$

Then, $\hat{f}(X + \delta) = c_A$ for all $\|\delta\|_2 \leq R^*$, where

$$R^* = \frac{\sigma}{2} \left(\Phi^{-1}(\overline{P_A}) - \Phi^{-1}(\underline{P_B}) \right),$$

and Φ^{-1} is the inverse of the standard Gaussian cumulative distribution function (CDF). Furthermore, we will show that $\hat{f}$ is lipschitz with the constant $L_{\hat{f}} \leq \frac{\sqrt{D}}{\sigma}$.

Before Proving the theorem, the certified radius R^* goes to ∞ as $\overline{P_A} \to 1$ and $\underline{P_B} \to 0$. This should sound reasonable: the Gaussian distribution is supported on all of $\mathbb{R}^d$, so the only way that $f(X+t) = c_A$ with probability 1 is if $f = c_A$ almost everywhere.

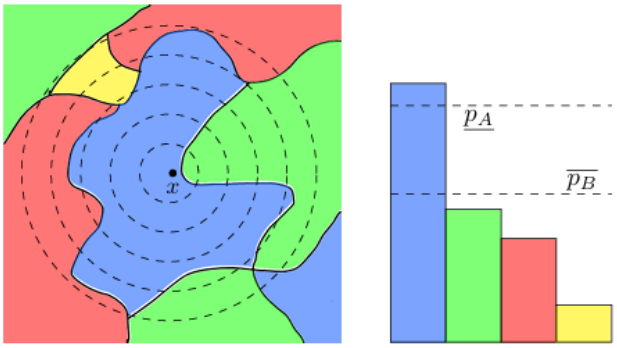

Figure 1. Evaluating the smoothed classifier at an input x.
Left: The decision regions of the base classifier f are shown in different colors. The dotted lines represent the level sets of the Gaussian distribution $\mathcal{N}(x, \sigma^2 I)$.
Right: The distribution $f(\mathcal{N}(x, \sigma^2 I))$. As discussed below, $\overline{p_A}$ is a lower bound on the probability of the top class, and $\underline{p_B}$ is an upper bound on the probability of each other class. Here, $\overline{f}(x)$ is represented as "blue".

Proof. For the time being let $f : \mathbb{R}^{\mathbb{D}} \to [0, 1]$. Since the perturbed image is $Z = X + t$, then we can say $P_T * P_X = P_Z$ which suggests that: $\hat{f}(X) = f(X) * g_\sigma(X)$ where g_σ is the pdf of zero-centered gaussian with covariance $\sigma^2 I_D$. Furthermore, we have:

$$\begin{aligned}\hat{f}(X) &= \mathbb{E}_{z\sim\mathcal{N}(0,I_D)}[f(X+\sigma z)] \\ &= \int_{\mathbb{R}^{\mathbb{D}}} \frac{f(X+\sigma z)}{(2\pi)^{D/2}} \exp\left(-\frac{\|z\|_2^2}{2}\right) dz.\end{aligned}$$

Generally speaking, since convolution is a linear operator and since only g depends on X, therefore:

$$\begin{aligned}\nabla_X \hat{f}(X) &= \nabla_X (f * g_\sigma) \\ &= f * \nabla_X g_\sigma \\ &= \mathbb{E}_{z\sim\mathcal{N}(0,I_D)}\left[f(X+\sigma z)\frac{z}{\sigma}\right]\end{aligned}$$

However, we know that the lipschitz constant of $\hat{f}$ is less than $||\nabla_X f(X)||_2$, or in other words: $L_{\hat{f}} \leq ||\nabla_X f(X)||_2$. Hence:

$$L_{\hat{f}} \leq \left\|\mathbb{E}_{z\sim\mathcal{N}(0,I_D)}[f(X+\sigma z)]\frac{z}{\sigma}\right\|_2 \leq \frac{1}{\sigma}\mathbb{E}_{z\sim\mathcal{N}(0,I_D)}[\|z\|_2]$$

Now, Since $h(x) = x^2$ is a convex function, for a random variable X, we have Jensen's Inequality:

$$h(\mathbb{E}[g(X)]) \leq \mathbb{E}[h(g(X))] \quad \text{or} \quad \mathbb{E}[g(X)] \leq \sqrt{\mathbb{E}[g^2(X)]}$$

Therefore, we can deduce that:

$$L_{\hat{f}} \leq \frac{1}{\sigma}\mathbb{E}_{z\sim\mathcal{N}(0,I_D)}[\|z\|_2] \leq \frac{1}{\sigma}\sqrt{\mathbb{E}_{z\sim\mathcal{N}(0,I_D)}[\|z\|_2^2]} = \frac{\sqrt{D}}{\sigma}$$

Thus, the lipschitz constant of f is at least $\frac{\sqrt{D}}{\sigma}$. However what we proved is a general case where we have an interval. But in this context we had a finite set $\mathcal{C}$, which is ok since this finite set is a specific case of the whole case.

After the dissemination of this work by Cohen et al, a more general result was published in Levine et al. (2019); Salman et al. (2019): If $h : \mathbb{R}^d \to [0, 1]$ is a function and h is the "smoothed" version,

$$h(X) = \mathbb{E}_{z\sim\mathcal{N}(0,\sigma^2 I_D)}[h(X+\sigma z)],$$

then the function $X \mapsto 1(h(X))$ is 1-Lipschitz. Theorem 1 can be proved by applying this result to the functions $f_c(X) = 1[f(X) = c]$ for each class c.

□

Without loss of generality, let us assume that each image has only one channel and that each pixel is represented using 8 bits. When perturbing an image X with a perturbation vector t, if we consider all possible perturbation vectors $t \in \mathcal{V}$, we can observe that the space exhibits periodic behavior. This periodicity arises because adding a sufficiently large value to a pixel causes an overflow, effectively wrapping around within a finite set of values. Consequently, the image space can be viewed as a periodic space with a finite number of elements.

We can visualize this image space using the depiction in Figure 2. Given this, we can partition the image space χ into $k = |\mathcal{C}|$ disjoint subsets as follows:

$$\chi = \bigcup_{i=1}^{k} \psi_i$$

where

$$\forall i \in \mathcal{C} : \psi_i = \{x \in \chi \mid f(x) = i\}.$$

So this part is very tricky. We make a transformation from the label space to the hamming space such that for each class, the neighboring class that have boundary regions with

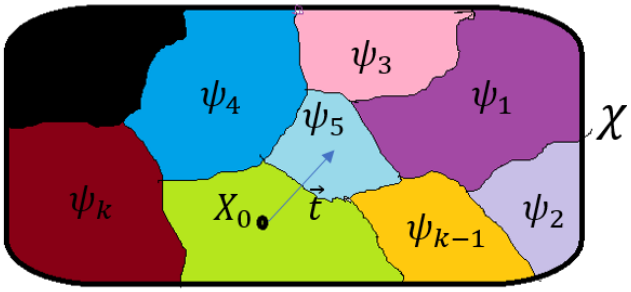

Figure 2. As shown, the image space χ is partitioned as $\bigcup_{i=1}^{k} \psi_i$, where $\psi_i \cap \psi_j = \emptyset$ for all $i \neq j$. Furthermore, the decision boundaries between different classes are clearly visible. It is also important to note that the image space χ is finite.

eachother have hamming distance of 1.

Assumption 2.2. Let $\tilde{t} = \phi(f(X_0)$, let's assume that he classifier $\hat{f}$ has the following property: for perturbed vectors t, s and an image X_0 from image space χ :

$$d_{\mathcal{H}}(f(X+t), f(X+s)) \leq cL_{\hat{f}}||t-s||_p$$

Thus, it is reasonable to introduce **Assumption 1**, which states that our trained classifier performs well in the sense that it rarely assigns a class to an image unless the class has a decision boundary in close proximity. As a result, when an image X_0 is perturbed by a vector t, the Hamming distance between X_0 and its perturbed version $X_0 + t$ is highly likely to be either *0* or *1*. This means that, with high probability, small perturbations will not significantly alter the classification outcome, reinforcing the robustness of our classifier under minor changes.
Therefore, given the assumption stated above, we can now redefine the partitioning of the image space χ in terms of the Hamming space, leading to the following new partitioning:

$$\chi = \bigcup_{i=1}^{k} \hat{\psi}_i$$

where

$$\forall i \in \mathcal{C}: \quad \hat{\psi}_i = \{x \in \chi \mid \phi(f(x)) = \phi(i)\}.$$

In this new space, the Hamming distance serves as a meaningful metric, as all images sharing the same label have a Hamming distance of **zero**, while images belonging to neighboring classes have a Hamming distance of **one**. This redefinition aligns well with our goal of measuring classification robustness within a discrete, structured framework.

Lemma 2.3. *Let $\mathcal{I}$ be an index space with metric $d(.,.)$. Then, for any $\epsilon > 0$, the following bound holds:*

$$\mathbb{E}\left[\sup_{i \in \mathcal{I}} f_i\right] \leq \int_0^{\infty} \sqrt{\log N(\epsilon, d, \mathcal{I})}\, d\epsilon$$

where $N(\epsilon)$ denotes the covering number of the index space $\mathcal{I}$ at scale ϵ, and the integral provides a bound on the growth of the supremum over the index space.

However, there is a better lemma that we are going to use:

Lemma 2.4. *Let $\mathcal{I}$ be an indexing set, and let $d(.,.)$ be a metric. Let $N(\epsilon, d, \mathcal{I})$ be the covering number of $\mathcal{I}$ at scale ϵ, i.e., the smallest number of ϵ-balls required to cover $\mathcal{I}$. Then we have:*

$$\mathbb{E}\left[\sup_{i \in \mathcal{I}} f_i\right] \leq \sup_{\epsilon} \left(\epsilon \sqrt{\log N(\epsilon, d, \mathcal{I})}\right).$$

We also need another useful lemma:

Lemma 2.5. *Assume that $\mathcal{K}$ is the set of binary strings of length n. Let $\mathcal{N}(\mathcal{K}, d_{\mathcal{H}}, m)$ and $\mathcal{P}(\mathcal{K}, d_{\mathcal{H}}, m)$ denote the covering number and packing number of $\mathcal{K}$ with respect to the Hamming distance $d_{\mathcal{H}}$ at scale m, respectively. Then we have :*

$$\mathcal{N}(\mathcal{K}, d_{\mathcal{H}}, m) \leq P(K, d_{\mathcal{H}}, m) \leq \frac{2^n}{\left(\sum_{k=0}^{\lfloor \frac{m}{2} \rfloor} \binom{n}{k}\right)}$$

Proof. Let $\{a_1, \ldots, a_n\}$ where $m = P(K, d_{\mathcal{H}}, m)$ be an m-packing for the set $\mathcal{K} = \{0,1\}^n$. We can see that $\{a_1, \ldots, a_n\}$ is an m-covering as well, otherwise there would be a point a_{n+1} that has a distance less than m with all $\{a_i\}_{i=1}^n$, which can't happen due to the maximality of packing $\{a_1, \ldots, a_n\}$. Which suggests:

$$\mathcal{N}(\mathcal{K}, d_{\mathcal{H}}, m) \leq \mathcal{P}(\mathcal{K}, d_{\mathcal{H}}, m)$$

Now we consider an m-packing $\{a_1, \ldots, a_m\}$ of binary strings $\{0,1\}^n$ of length n. Now since each a_i, a_j have at least m different binary digits; if we take each a_i and alter $l \leq \lfloor \frac{m}{2} \rfloor$ digits of a_i, we will still get distinct elements.

$$\mathcal{C} = \{x \ : \ d_h(x, a_i) \leq \lfloor \frac{m}{2} \rfloor; \{a_i\}_{i=1}^m \text{ is } m\text{-packing}\}.$$

Since the total number of strings of length n is 2^n, we have:

$$|\mathcal{C}| = P(K, d_{\mathcal{H}}, m) \cdot \left(\sum_{k=0}^{\lfloor \frac{m}{2} \rfloor} \binom{n}{k}\right) \leq 2^n = |K| \checkmark$$

Thus we get:

$$\mathcal{N}(\mathcal{K}, d_{\mathcal{H}}, m) \leq P(K, d_{\mathcal{H}}, m) \leq \frac{2^n}{\left(\sum_{k=0}^{\lfloor \frac{m}{2} \rfloor} \binom{n}{k}\right)}$$

□

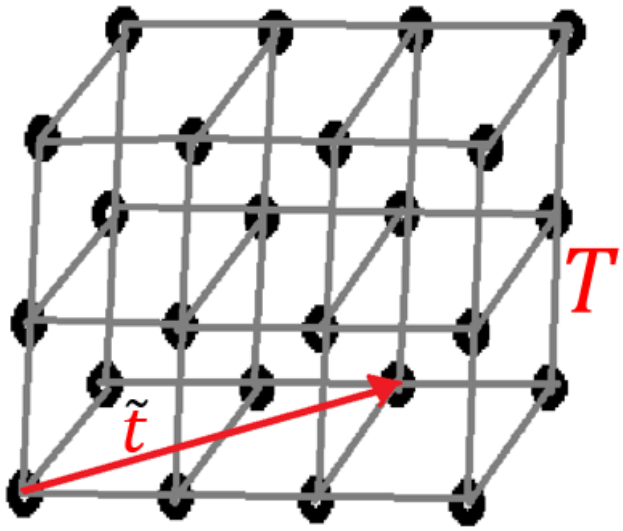

Figure 3. The Visualization of k-dimensional hamming space and the vector $\tilde{t} \in \mathcal{T}$, where $\mathcal{T}$ is used as indexing space.

Now using the lemmas above, we can easily prove that if we take the hamming difference of the detected label of the image X and the detected label of the perturbed image $X + t$ as our random vector $Y_{\tilde{t}}$ we get:

$$Y_{\tilde{t}} = d_{\mathcal{H}}(\phi(f(X+t)), \phi(f(X))) = d_{\mathcal{H}}(v + \tilde{t}, v)$$
$$\text{where } \tilde{t} = \phi(f(X+t)) - v \quad \text{and} \quad v = \phi(f(X))$$

Then since the hamming distance is a metric and if we take our indexing space as $\mathcal{T}$, then from lemmas above, we can deduce that:

$$\begin{aligned} &\mathbb{E}_{X \sim P_X}\left[d_{\mathcal{H}}(\phi(f(X+t)), \phi(f(X)))\right] \\ &\quad = \mathbb{E}_{X \sim P_X}\left[d_{\mathcal{H}}(v + \tilde{t}, \tilde{t})\right] \\ &\quad \leq \sup_m m \cdot \left(\frac{2^n}{\sum_{k=0}^{\lfloor \frac{m}{2} \rfloor} \binom{n}{k}}\right) \end{aligned}$$

which completes the proof and the bound we were looking for.

3. Experiments & Results

In this section we take a look into the work of Dezfooli et al., in this paper they tried to find perturbations that when added to the original image, fools the model with high probability. here is the algorithm proposed by Dezfooli et al. :

Algorithm 1 Computation of Universal Perturbations

Data points X, classifier $\hat{k}$, desired p-norm of the perturbation ξ, desired accuracy on perturbed samples ϵ. Universal perturbation vector v. Initialize $v \leftarrow 0$. $\text{Err}(X + v) \leq 1 - \epsilon$ each datapoint $x_i \in X$ $\hat{k}(x_i + v) = \hat{k}(x_i)$ Compute the minimal perturbation that sends $x_i + v$ to the decision boundary:

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \quad \text{s.t.} \quad \hat{k}(x_i + v + r) \neq \hat{k}(x_i).$$

Update the perturbation:

$$v \leftarrow P_{p,\xi}(v + \Delta v_i).$$

For better understanding we can take a look at this image which is depicted down below:

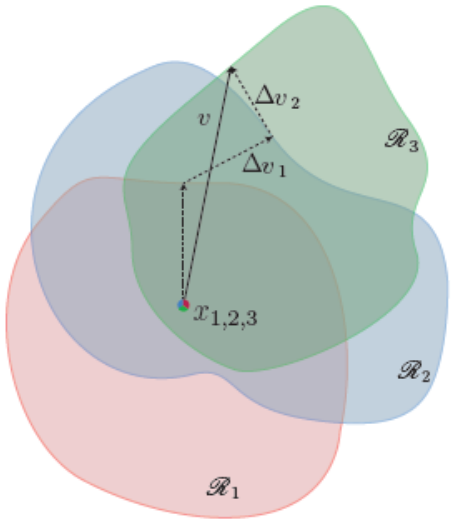

Figure 4. Schematic Representation of the Proposed Algorithm In this illustration, data points x_1, x_2, and x_3 are superimposed, and the classification regions R_i (i.e., regions of constant estimated label) are shown in different colors.
Our algorithm proceeds by sequentially aggregating the minimal perturbations that push the current perturbed points $x_i + v$ outside their corresponding classification region R_i.

Here we changed the algorithm and considered the l_∞ norm instead of l_2 norm. Since l_∞ norm measures the maximum

change to any pixel in the image, Perturbations under the l_∞ norm are also small but can be more effective in causing misclassification compared to l_2 and l_1. And As we can see, the results are as we expected:

Attack Method	Accuracy Attack	Accuracy After Attack
FGSM	94.2%	22.1%
PGD	94.2%	21.7%
L-BFGS	94.2%	27.6%
UAP	94.2%	29.1%
Ours	94.2%	38.5%

Table 1. Comparison of accuracy before and after different adversarial attacks.

Conclusion

In this study, we have examined the literature on adversarial attack algorithms and the corresponding defense mechanisms, highlighting the complexity of addressing adversarial vulnerabilities in machine learning models. By simulating a variety of attack strategies, specifically utilizing p-norms, we were able to observe significant degradation in model performance, indicating the potency of adversarial perturbations in real-world applications. Our findings underscore the ongoing challenges in securing models against such attacks, suggesting that adversarial robustness remains a crucial area of research for ensuring the reliability and safety of machine learning systems.

Future Work

There are several promising directions for future research in the domain of adversarial attacks and defenses. One potential avenue is to explore the relationship between the Wasserstein distance of perturbed and original distributions. By placing an upper bound on this distance, we can better understand the interplay between perturbation magnitude and model performance, leading to more effective defense strategies. Additionally, investigating the use of Gaussian smoothing distributions for mitigating adversarial effects could provide insights into novel defense mechanisms. This approach has the potential to improve model robustness by smoothing out high-frequency noise while preserving important features in the data. Finally, further studies could focus on adaptive adversarial attacks, where attackers dynamically adjust their strategies based on the defenses employed, in turn driving the development of more sophisticated and generalizable defense methods.

Software and Data

We used these github repositories for FGSM, UAP, PGD and L-BFGS attacks on a DNN classifier trained on MNIST dataset. We also put our codes in this github repository.

Acknowledgements

Many thanks to our mentor, Dr. Yassaee, for his invaluable guidance and support throughout this project. We also appreciate the insightful feedback and suggestions from our mentor, Dr. Rahmani, which significantly contributed to the improvement of this work. We also appreciate the team behind the *Universal Adversarial Perturbations* and *Certified Adversarial Robustness via Randomized Smoothing* paper for their contributions to this field.

References

1. Jeremy M Cohen, Elan Rosenfeld, J. Zico Kolter, "Certified Adversarial Robustness via Randomized Smoothing," ICML 2019, arXiv:1902.02918.

2. Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, "Universal Adversarial Perturbations," CVPR 2017, arXiv:1610.08401.

3. Ashutosh Chaubey, Nikhil Agrawal, Kavya Barnwal, Keerat K. Guliani, Pramod Mehta, "Universal Adversarial Perturbations: A Survey," arXiv 2020, arXiv:2005.08087.