Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope

Author: Eric Wong, J. Zico Kolter, Summary: Sungyoon Lee

I. Introduction

딥러닝은 최근 여러 분야에서 뛰어난 성능을 보여주고 있다. 그러나 입력값에 작은 노이즈를 주어 모델의 성능을 낮추거나 원치 않은 방향으로의 선택을 유도하는 공격이 가능하다. 이를 adversarial attack이라고 부르고 adversarial attack을 받은 입력값을 adversarial examples이라 한다. adversarial attack이 가능함이 알려지고 attack과 defense 방향으로의 다양한 연구들이 진행되어 왔다. 그러나 최근의 연구들은 기존의 연구 방향들의 문제점을 지적하며, 가능한 attack의 집합을 정해두고 그 attack들에 대해 robust한 defense를 개발하는 방향이 합리적이다고 역설하고 있다. 이러한 흐름에서 robust optimization를 적용하여 adversarial attack을 방어하고자 하는 다양한 시도들이 이루어지고 있다.

II. Problems

본 논문에서는 norm-bounded adversarial attack 집합에 대해 robustness가 보장되는 ReLU기반의 딥러닝 모델을 학습하는 방법을 제안한다. 이를 위해 input space, $\mathcal{X}$ 상에서의 기존 input $x \in \mathcal{X}$와 adversarial example이 가능한 영역인 norm-ball, $B(x,\epsilon) \subset \mathcal{X}$를 생각한다. 이때 classifier 모델에서 마지막 softmax layer를 제거한 logit space, $\mathcal{Z}$로의 함수 $f_\theta : \mathcal{X} \to \mathcal{Z}$를 생각하고 이 함수에 대한 $B(x,\epsilon)$의 image, $\mathcal{Z}(x,\epsilon) \equiv f_\theta(B(x,\epsilon)) \subset \mathcal{Z}$를 생각하면 piecewise-linear한 ReLU기반의 모델의 경우, $\mathcal{Z}(x,\epsilon)$는 다면체(polytope) 형태이고, 이를 adversarial polytope라 부른다. 본 논문의 목적은 해당 adversarial polytope가 decision boundary를 벗어나지 않는, 즉 다른 class로 분류되지 않는 robust한 모델을 얻고자하는 것이다. 구체적으로 adversarial polytope 안에서 x의 true class에 해당하는 $\hat{z}_{true}$의 값이 다른 class에 해당하는 $\hat{z}_{false}$의 값보다 항상 크길 바라는 것이다. 따라서 $\hat{z}_{false} - \hat{z}_{true}$의 logit difference를 loss와 같이 생각할 수 있다.

III. Convex Formulation

해당 문제를 optimization 문제로 구성하기 위해 먼저 objective를 worst case logit difference, $W(\theta) = \max_{\hat{z} \in \mathcal{Z}(x,\epsilon)} \hat{z}_{false} - \hat{z}_{true}$로 생각하면 다음 식과 같은 optimization problem이 얻어진다.

$$\textbf{(Opt.)} \quad \min_{\theta \in \Theta} W(\theta) = \min_{\theta \in \Theta} \max_{\hat{z} \in \mathcal{Z}(x,\epsilon)} \hat{z}_{false} - \hat{z}_{true}$$

모델의 파라미터 θ에 대한 outer minimization은 딥러닝에서 주로 사용하는 SGD방법을 통해 optimization 가능하다. 따라서 inner maximization 문제가 convex formulation을 하기 위한 대상이다. 이때 $\hat{z}_{false} - \hat{z}_{true} = (e_{false} - e_{true})^T \hat{z} = c^T \hat{z}$이므로 optimization variable $\hat{z}$에 대해 linear한 objective를 가지는 문제로 볼 수 있다. 그러나 convex optimization 문제로 볼 수 없는 이유는 constraint에 해당하는 $\hat{z} \in \mathcal{Z}(x,\epsilon)$에서 adversarial polytope, $\mathcal{Z}(x,\epsilon)$는 일반적으로 convex하지 않기 때문이다. 따라서 우리는 $\mathcal{Z}(x,\epsilon)$를 포함하는 convex set인 convex outer bound, $\hat{\mathcal{Z}}(x,\epsilon)$로 바꾸어 convex constraint를 얻는 대신에, maximization 문제의 upper bound를 구하는 relaxation을 적용한다. 결과적으로 우리는 LP 문제를 얻을 수 있다.

논문: https://arxiv.org/pdf/1711.00851.pdf,
공개 code: https://github.com/locuslab/convex_adversarial

그러나 모든 training examples, x_i에 대해 반복해서 LP를 직접 푸는 것은 computationally impractical하기 때문에, 한번 더 relaxation을 진행한다. 두번째 relaxation은 기존 LP 문제에 대해 dual 문제를 구성하고 적당한 feasible solution을 대입해 얻은 dual value, $\hat{d}$를 다시 한번 upper bound로 사용하는 방법이다. 두 과정을 간단한 식으로 나타내면 아래와 같다.

$$\textbf{(Rlx.)} \quad \max_{\hat{z} \in \mathcal{Z}(x,\epsilon)} c^T \hat{z} \le \max_{\hat{z} \in \hat{\mathcal{Z}}(x,\epsilon)} c^T \hat{z} = p^* \le d^* \le \hat{d}$$

이제 이 **(Rlx.)** 식을 이용하여 inner maximization 대신 $\hat{d}$의 값을 사용하고, outer minimization을 진행하면 근사적으로 기존의 문제에 대한 풀이가 가능하다.

전체적인 틀은 위와 같지만 logit difference 대신에 cross entropy loss, L을 이용하고 아래 **(Thm.)** 식에 표현된 정리를 이용하여 inner maximization의 upper bound에 해당하는 우변을 SGD를 이용하여 minimize하는 방식으로 robust training을 진행한다. 이때 **(Thm.)** 식에서 우변을 robust loss라 한다.

$$\textbf{(Thm.)} \quad \max_{x' \in B(x,\epsilon)} L(f_\theta(x'), y) \le L(-J_\epsilon(x, g_\theta(e_y 1^T - I)), y)$$

IV. Experimental Results

저자가 공개한 code를 바탕으로 수정하여 실험을 재현해 본 결과, 2D toy data의 경우 Fig. 1에 표현한 바와 같이 normal training에서는 input space에서 box로 표현된 norm-ball이 decision boundary를 벗어난 경우가 많은 것과 비교하여 이 논문에서 제안한 robust training을 통해 adversarial attack에 대한 robustness가 증가한 것을 볼 수 있다.

또한 MNIST에 대해, 논문과 같은 learning parameter를 사용하여 반복 실험해본 결과 논문에서 주장하는 5.82%에 근접하는 6.42%(7.86, 6.59, 6.54, 6.42, 7.04, 7.86, 6.47)의 robust error를 얻을 수 있었다. Fig. 1의 learning curve에서 볼 수 있듯이, normal training loss의 upper bound에 해당하는 robust loss(black)를 minimize함으로써 normal loss(red)도 함께 줄어드는 학습이 효율적으로 이루어졌음을 확인할 수 있다.

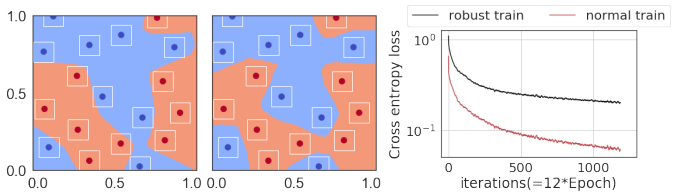

Fig. 1: (왼쪽) 2D toy data results. (1) normal training, (2) robust training 결과. (오른쪽) MNIST robust training learning curve

(message to home) 본 논문에서는 convex formulation을 위해 **convex outer bound**를 생각하여 LP문제로 변형시키고, 근사적이지만 계산적으로 효율적인 알고리즘을 위해 **dual problem**의 **feasible solution**에 대응하는 upper bound를 사용하여 numerical optimization 사용을 최소화 시켰다. 이러한 방법론은 **exact optimization**보다는 **approximate optimization**만으로도 충분한 (또는 **computational efficiency**가 더 중요한) 여러 다양한 공학적 문제에 적용할 수 있을 것으로 생각된다.

V. Appendix

1) Experimental Results와 관련한 모든 code는 `https://github.com/sungyoon-lee/convex_adversarial` 에 upload 하였다. 추가적으로 전체적인 이해를 돕기위한 ppt도 upload하였다.

2) 전체적인 framework는 아래와 같다.

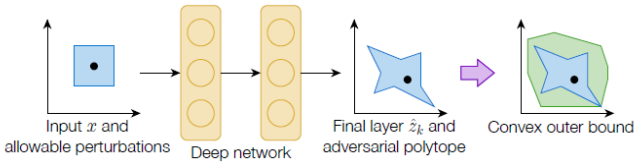

Fig. 2: norm-ball이 딥러닝 네트워크를 통해 adversarial polytope를 만들고, 본 논문에서는 이를 포함하는 convex outer bound를 생각하여 convex formulation을 진행한다.

3) (**Opt.**) 식에서 $W(\theta)$의 계산까지 SGD를 통해 alternatively 계산하면 minimax optimization의 instable한 문제로 인해 성능이 좋지 못할 것이다.

4) e_i는 i번째 성분만 1이고 나머지는 0인 vector를 표현한 것으로, basis vector로 이해할 수 있다.

5) convex outer bound를 얻기 위해서 기존의 ReLU activation 대신에 아래 식으로 표현되는 convex envelops를 사용한다.

$$z \geq 0, z \geq \hat{z}, z \leq \frac{u}{u-l}(\hat{z} - l)$$

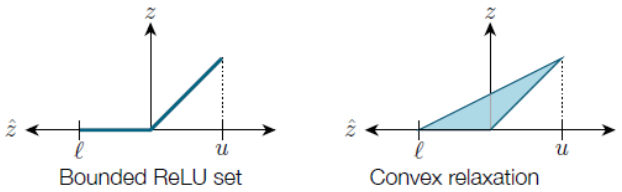

Fig. 3: (왼쪽) ReLU activation. (오른쪽) convex envelops

6) adversarial polytope, $\mathcal{Z}(x, \epsilon) \equiv f_\theta(B(x, \epsilon))$가 θ에 의존하는 것을 표현하기 위해 $\mathcal{Z}_\theta(x, \epsilon)$로 쓸 수 있다. 마찬가지로 $p^*_\theta, d^*_\theta, \hat{d}_\theta$로 쓸 수 있고 (**Rlx.**)를 새로 표현하면

(**Rlx1.**) $\max_{\hat{z} \in \mathcal{Z}_\theta(x,\epsilon)} c^T \hat{z} \leq \max_{\hat{z} \in \hat{\mathcal{Z}}_\theta(x,\epsilon)} c^T \hat{z} = p^*_\theta \leq d^*_\theta \leq \hat{d}_\theta$

7) MNIST의 경우, Titan XP를 사용하여 robust training은 4시간 15분, normal training은 12분 정도 소요되어 약 20배정도의 차이가 나는 것을 확인하였다. normal training은 실제로는 3 epochs만에 충분히 수렴하므로 그 차이는 더 크다고 할 수 있다.

8) (**Thm.**) 식에서 우변에 나타난 J_ϵ, g_θ는 dual formulation에서 나오는 dual network에 대한 식이다. 식이 복잡하기에 따로 표현하지는 않았지만 J_ϵ는 dual loss에 해당하고, g_θ는 dual network에 해당한다. 자세한 식은 논문을 참고하기 바란다.