KTH ROYAL INSTITUTE OF TECHNOLOGY

EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples

Authors : Pin-Yu Chen, Cho-Jui Hsieh, Jinfeng Yi, Huan Zhang and Yash Sharma

Presenter: Ezgi Korkmaz

Elastic Net Regularization

- Elastic-net regularization is widely used in solving high-dimensional feature selection
- Can be viewed as a regularizer linearly combines two penalty functions L_1 and L_2
- Used in following minimization problem

$$\underset{z \in Z}{\text{minimize}} \quad f(z) + \lambda_1 ||z||_1 + \lambda_2 ||z||_2^2$$

- where z is a vector of p optimization variables
- Z the set of feasible solutions
- $f(z)$ denotes a loss function
- $||z||_q$ denotes L_q norm
- $0 \leq \lambda_1, \lambda_2$ are the regularization parameters

Elastic Net Regularization

$$\underset{z \in Z}{\text{minimize}} \quad f(z) + \lambda_1 ||z||_1 + \lambda_2 ||z||_2^2$$

- $0 \leq \lambda_1, \lambda_2$ are the regularization parameters
- The elastic-net regularization yields LASSO when $\lambda_2 = 0$
- Becomes ridge regression formulation when $\lambda_1 = 0$
- Elastic-net regularization is able to select a group of highly correlated features overcomes the shortcoming of high dimensional feature selection

EAD Formulation and Generalization

- Same loss function with Carlini and Wagner
- Given an image x_0 and correct label denoted by t_0
- Adversarial example x of image x_0 with a target $t \neq t_0$
- The loss function $f(x)$ defined as

$$f(x,t) = \max\{\max_{j \neq t}[\textbf{Logit}(x)]_j - [\textbf{Logit}(x)]_t, -\kappa\} \qquad (1)$$

- where $\textbf{Logit} = [[\textbf{Logit}]_1, .., [\textbf{Logit}]_K] \in \mathbb{R}^K$ is the layer before the softmax
- K is the number of classes
- $0 \leq \kappa$ is the confidence parameter guarantees a constant gap between $\max_{j \neq t}[\textbf{Logit(x)}]_j$ and $[\textbf{Logit(x)}]_t$

EAD Formulation and Generalization

- $[\textbf{Logit}]_t$ is proportional to the probability of predicting x as label t

$$Prob(Label(x) = t) = \frac{\text{exp}([\textbf{Logit}]_t}{\sum_{j=1}^{K} exp([\textbf{Logit}]_j)} \tag{2}$$

- The loss function aims to render label t the most probable outcome for x and the parameter κ controls the separation between t and most likely prediction other than t.
- For untargeted,

$$f(x) = \max\{[\textbf{Logit}(x)]_{t_o} - \max_{j \neq t}[\textbf{Logit}(x)]_j, -\kappa\} \tag{3}$$

- Focus of the paper is targeted
- Can be applied to targeted by replacing $f(x,t)$ by $f(x)$

EAD Formulation and Generalization

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c \cdot f(x,t) + \beta ||x - x_0||_1 + ||x - x_0||_2^2 \\ \text{subject to} \quad & x \in [0,1]^p \end{aligned}$$

- where $0 \leq c, \beta$ are regularization parameters of the loss function and L_1 penalty respectively
- The box constraint $x \in [0,1]^p$ restricts x to be a properly scaled image space
- EAD formulation aims to find
  - An adversarial example x
  - Classified as target class t
  - While minimizing the distortion $\delta = x - x_0$

EAD Formulation and Generalization

- δ is the loss
  - $\beta||\delta||_1 + ||\delta||_2^2$
  - Linear combination of L_1 and L_2 metrics
- CW is a special case of EAD
  - when $\beta = 0$ disregards the L_1 penalty on δ
- L_1 penalty is an intuitive regularizer
- $||\delta||_1 = \sum_{i=1}^{p} |\delta_i|$ represents the total variation of the perturbation
- Surrogate function for promoting sparsity
- Improves attack transferability
- Complements adversarial learning

EAD Algorithm

- CW used change-of-variable (COV) approach via $tanh$ transformation on x remove the box constraint $x \in [0, 1]^p$
- When $0 < \beta$ COV is not effective
- Since the corresponding adversarial examples are insensitive to the changes in β
- L_1 penalty is non-differentiable, yet piece-wise linear function
- The failure of COV approach can be explained by inefficiency in subgradient based optimization problems.
- Paper propose iterative shrinkage thresholding algorithm (ISTA)
- ISTA regular first order optimization algorithm with an additional shrinkage thresholding step on each iteration

Iterative Shrinkage Thresholding algorithm (ISTA)

- Let $g(x) = c \cdot f(x) + ||x - x_0||_2^2$ and $\nabla g(x)$ be the numerical gradient computed by DNN
- At $k+1$ iteration the adversarial example x^{k+1} of x_0

$$x^{(k+1)} = S_\beta(x^{(k)} - \alpha_k \nabla g(x^{(k)})) \tag{4}$$

- α_k denotes the step size at the $k+1$ iteration
- $S_\beta : \mathbb{R}^p \to \mathbb{R}^p$ element-wise projected shrinkage thresholding function

$$[S_\beta(z)]_i = \begin{cases} \min\{z_i - \beta, 1\} & z_i - x_{0i} > \beta \\ x_{0i} & |z_i - x_{0i}| \leq \beta \\ \max\{z_i + \beta, 0\} & z_i - x_{0i} < -\beta \end{cases}$$

Iterative Shrinkage Thresholding Algorithm (ISTA)

$$[S_\beta(z)]_i = \begin{cases} \min\{z_i - \beta, 1\} & z_i - x_{oi} > \beta \\ x_{oi} & |z_i - x_{oi}| \leq \beta \\ \max\{z_i + \beta, 0\} & z_i - x_{oi} < -\beta \end{cases}$$

- For any $i \in \{1, 2, .., p\}$
- Shrinks to element z_i by β when $z_i - x_{oi} > \beta$
- Thresholds z_i by setting $[S_\beta(z)]_i = x_{oi}$ when $|z_i - x_{oi}| \leq \beta$
- $g(x)$ is the attack objective function of the C&W method
- ISTA can be seen as robust version of C&W
- Where shrinks a pixel value of the adversarial example if the deviation to the original image is greater than β and keeps a pixel value unchanged if the deviation is less than β

Elastic-Net Attacks to DNNs (EAD)

Algorithm 1 Elastic-Net Attacks to DNNs (EAD)

input: Original labeled image x_0, t_0, target attack class t, attack transferability parameter κ, L_1 regularization parameter β, step size α_k, # of iteration I
output: An adversarial example x
Initialization: $x^{(0)} = y^{(0)} = x_0$
for k = 0:I-1 **do**
$x^{(k+1)} = S_\beta(y^{(k)} - \alpha_k \nabla g(y^{(k)})$
$y^{(k+1)} = x^{(k+1)} + \frac{k}{k+3}(x^{k+1} - x^{(k)})$
end for
Decision rule: determine x from successful examples in $\{x^{(k)}\}_{k=1}^{I}$ (EN rule or L_1) rule

- The slack vector $y^{(k)}$ incorporates the momentum $x^{(k)}$ for acceleration
- Learning rate set to $\alpha_0 = 0.01$
- During the EAD iterations $x^{(k)}$ considered as successful adversarial example of x_0
- If the model predicts its most likely class to be the target class t
- The final adversarial example is selected from all successful examples based on distortion metrics.
- Authors consider two decision rules for selecting x the least elastic-net and L_1 relative to x_0

Evaluation

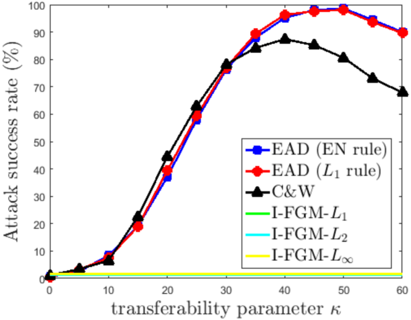
- EAD performance similar to C&W performance since C&W is the special case of EAD
- Compared to existing L_1 based FGM and I-FGM
  - Significantly lower L_1 distortion
  - Better attack success rate
- L_1 based adversarial examples crafted by EAD improves attack transferability and complements adversarial training.

Evaluation

	MNIST				CIFAR10				ImageNet			
Attack method	ASR	L_1	L_2	L_∞	ASR	L_1	L_2	L_∞	ASR	L_1	L_2	L_∞
C&W (L_2)	**100**	22.46	**1.972**	0.514	**100**	13.62	**0.392**	0.044	**100**	232.2	**0.705**	0.03
FGM-L_1	39	53.5	4.186	0.782	48.8	51.97	1.48	0.152	1	61	0.187	0.007
FGM-L_2	34.6	39.15	3.284	0.747	42.8	39.5	1.157	0.136	1	2338	6.823	0.25
FGM-L_∞	42.5	127.2	6.09	0.296	52.3	127.81	2.373	0.047	3	3655	7.102	0.014
I-FGM-L_1	**100**	32.94	2.606	0.591	**100**	17.53	0.502	0.055	77	526.4	1.609	0.054
I-FGM-L_2	**100**	30.32	2.41	0.561	**100**	17.12	0.489	0.054	**100**	774.1	2.358	0.086
I-FGM-L_∞	**100**	71.39	3.472	**0.227**	**100**	33.3	0.68	**0.018**	**100**	864.2	2.079	**0.01**
EAD (EN rule)	**100**	**17.4**	2.001	0.594	**100**	**8.18**	0.502	0.097	**100**	**69.47**	1.563	0.238
EAD (L_1 rule)	**100**	**14.11**	2.211	0.768	**100**	**6.066**	0.613	0.17	**100**	**40.9**	1.598	0.293

- Authors show that adversarial examples crafted by EAD can be successful as the state-of-the-art L_2 and L_∞ attacks in breaking and undefended and defensively distilled networks.
- Furthermore, it improves the attack transferability and complements the adversarial training.

Evaluation



- Attack transferability (average case) from the undefended network to the defensively distilled network on MNIST by varying Î°. EAD can attain nearly 99% attack success rate (ASR) when $\kappa = 50$, whereas the top ASR of the C&W attack is nearly 88 % when $\kappa = 50$.

Evaluation

Attack method	Adversarial training	Average case ASR	L_1	L_2	L_∞
C&W (L_2)	None	100	22.46	1.972	0.514
	EAD	100	26.11	2.468	0.643
	C&W	100	24.97	2.47	0.684
	EAD + C&W	100	27.32	2.513	0.653
EAD (L_1 rule)	None	100	14.11	2.211	0.768
	EAD	100	17.04	2.653	0.86
	C&W	100	15.49	2.628	0.892
	EAD + C&W	100	16.83	2.66	0.87

- Adversarial training using the C&W attack and EAD (L_1 rule) on MNIST. ASR means attack success rate. Incorporating L_1 examples complements adversarial training and enhances attack difficulty in terms of distortion.