KTH ROYAL INSTITUTE
OF TECHNOLOGY

Boosting Adversarial Attacks with Momentum

Authors : Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu and Jianguo Li

Presenter: Ezgi Korkmaz

- With the knowledge of the structure and parameters of a given model
  - Optimization based methods box-constraint L-BFGS
  - One-step gradient method FGSM
  - Iterative variants of gradient methods
- Transferability the adversarial examples crafted for one model remains adversarial for other
- Black-box attack becomes practical
- Different machine learning models learn similar decision boundaries

Background

- $f(x) : x \in X \rightarrow y \in Y$
- For correctly classified input x with ground truth label y
  - Non-targeted $f(x^*) \neq y$
  - Targeted $f(x^*) = y^*$
  - y^* is the target label
- L_p norm required to be less than an allowed value $||x^* - x||_p \leq \epsilon$

One-step gradient-based approach: FGSM

- Find an x^* by maximizing the loss function $J(x^*, y)$
- Where J is the cross-entropy loss
- FGSM generates adversarial examples to meet L_∞ norm bound $||x^* - x||_\infty \leq \epsilon$

$$x^* = x + \epsilon \cdot sign(\nabla_x J(x, y)) \tag{1}$$

- The fast gradient method (FGM) generalization of FGSM to meet L_2 norm bound $||x^* - x||_2 \leq \epsilon$

$$x^* = x + \epsilon \cdot \frac{\nabla_x J(x, y)}{||\nabla_x J(x, y)||_2} \tag{2}$$

Iterative Methods: IFGM

- Iteratively apply fast gradient multiple times witha small step size α

$$x_0^* = x, \quad x_{t+1}^* = x_t^* + \alpha \cdot sign(\nabla_x J(x_t^*, y)) \tag{3}$$

- Clip x_t^* into the ϵ or set $\alpha = \frac{\epsilon}{T}$ where T is the number of iterations.
- Stronger white-box adversaries with the cost of worst transferability.

Optimization-based Methods: L-BFGS

$$\min_{x^*} \lambda \cdot ||x^* - x||_p - J(x^*, y) \tag{4}$$

- Optimization based methods lack the efficacy in black-box attacks just like iterative

Defense Methods

- Adversarial training to increase the robustness of DNNs
- Injecting adversarial examples into the training procedure trained models learns to resist the perturbation in the gradient direction of the loss function
- Does not confer robustness to black-box attacks
- Ensemble adversarial training robust against one-step attacks and black-box attacks
- Because it produces adversarial samples model being trained and other hold-out models

MI-FGSM

Algorithm 1 MI-FGSM

input: A classifier f, with loss function J, a real example x and ground-truth label y
input: The size of perturbation ϵ, iteration T and decay factor μ
output: An adversarial example x^* with $||x^* - x||_\infty \leq \epsilon$
$\alpha = \epsilon / T$;
$g_0 = 0; x_0^* = x$;
for t = 0:T-1 **do**
  Input x_t^* to f and obtain the gradient $\nabla_x J(x_t^*, y)$;
  Update g_{t+1} by accumulating the velocity vector in the gradient direction as
  $g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_t^*, y)}{||\nabla_x J(x_t^*, y)||_1}$
  Update $x_{t+1}^* = x_t^* + \alpha \cdot sign(g_t + 1)$;
end for
return $x^* = x_T^*$

MI-FGSM

- Accelerate the gradient descent algorithms by accumulating a velocity vector in the direction of the gradient of the loss function
- Momentum is effective in stochastic gradient descent to stabilize the updates
- Gradient-based approach seek the adversarial example by solving the constrained optimization problem

$$\underset{x^*}{\operatorname{argmax}} J(x^*, y), \quad \text{s.t.} \quad ||x^* - x||_\infty \leq \epsilon \tag{5}$$

- FGSM generates an adversarial example by applying the sign of the gradient only once
- By the assumption of linearity of the decision boundary around the data point

MI-FGSM

- Linearity assumption may not hold if the distortion is large
- FGSM might underfit the model
- Limiting its attack ability
- On the contrary, I-FGSM greedily moves the adversarial example in the direction of the sign of the gradient in each iteration
- Adversarial example can into poor local maxima, "overfit" the model
- Means transferability across different models reduced
- Integrate momentum into I-FGSM and stabilize the update direction
- Escape from poor local maxima
- Alleviates the trade-off between the attack ability and the transferability

MI-FGSM

- Linearity assumption may not hold if the distortion is large
- FGSM might underfit the model
- Limiting its attack ability
- On the contrary, I-FGSM greedily moves the adversarial example in the direction of the sign of the gradient in each iteration
- Adversarial example can into poor local maxima, "overfit" the model
- Means transferability across different models reduced
- Integrate momentum into I-FGSM and stabilize the update direction
- Escape from poor local maxima
- Alleviates the trade-off between the attack ability and the transferability

Attacking Ensemble of Models

- If an example remains adversarial for multiple models, it may capture an intrinsic direction that always fools these models and more likely to transfer
- A powerful black-box attack
- Has been proposed to attack multiple models whose logit activations are fused
- To attack an ensemble of K models,
  - $l(x) = \sum_{k=1}^{K} w_k l_k(x)$
  - where $l_k(x)$ are the logits of the kth model
- w_k is the ensemble weight
  - $0 \leq w_k$
  - $\sum_{k=1}^{K} w_k = 1$

Attacking Ensemble of Models

- The loss function $J(x, y)$ is defined as the softmax cross-entropy loss between y and $l(x)$ where $\mathbb{1}_y$ is the one hot encoding of y

$$J(x, y) = \mathbb{1}_y \cdot \log(softmax(l(x))) \tag{6}$$

- For comparison K model can be averaged in predictions.
  - $p(x) = \sum_{k=1}^{K} w_k p_k(x)$
  - where $p_k(x)$ is the predicted probability of the kth model given input x
- K models can also be averaged in loss
- $J(x, y) = \sum_{k=1}^{K} w_k J_k(x, y)$

MI-FGSM for an ensemble of models

Algorithm 2 :MI-FGSM for an ensemble of models

input: The logits of K classifiers $l_1, l_2, l_3 ... l_K$; ensemble weights $w_1, w_2 ..., w_K$ a real example x and ground-truth label y
input: The size of perturbation ϵ, iteration T and decay factor μ
output: An adversarial example x^* with $||x^* - x||_\infty \leq \epsilon$
$\alpha = \epsilon / T$;
$g_0 = 0; x_0^* = x$;
for t = 0:T-1 **do**
    Input x_t^* and output $l_k(x_t^*)$ for $k = 1, 2.., K$
    Fuse the logits as $l(x_t^*) = \sum_{k=1}^{K} w_k k_k(x_t^*)$;
    Get softmax cross-entropy loss $J(x_t^*, y) basedonl(x_t^*)$
    Obtain the gradient $\nabla_x J(x_t^*, y)$;
    Update g_{t+1}
    Update x_{t+1}^*
end for
return $x^* = x_T^*$

- Empirical result ensemble in logits performs better than ensemble in predictions.

Extensions

- M-IFGSM for L_2 distance

$$x^*_{t+1} = x^*_t + \alpha \cdot \frac{g_{t+1}}{||g_{t+1}||_2} \tag{7}$$

- where,

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x^*_t, y)}{||\nabla_x J(x^*_t, y)||_1} \tag{8}$$

- For targeted,

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x^*_t, y^*)}{||\nabla_x J(x^*_t, y^*)||_1} \tag{9}$$

Extensions

- M-IFGSM for L_2 distance

$$x^*_{t+1} = x^*_t + \alpha \cdot \frac{g_{t+1}}{||g_{t+1}||_2} \tag{10}$$

- M-IFGSM with an L_∞ norm bound,

$$x^*_{t+1} = x^*_t - \alpha \cdot sign(g_{t+1}) \tag{11}$$