Learning to Detect Adversarial Examples Based on Class Scores

Tobias Uelwer, Felix Michels, and Oliver De Candido

44th German Conference on Artificial Intelligence (KI)

October 1st, 2021

TUM Uhrenturm

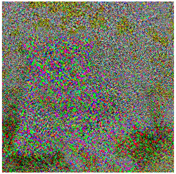
Adversarial Attacks

image

“lion”

$+ \; 0.01 \; \cdot$

perturbation



$=$

adv. image

“broccoli”

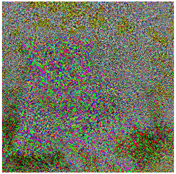
Adversarial Attacks

image

“lion”

$+ \; 0.01 \; \cdot$

perturbation



$=$

adv. image

“broccoli”

- General problem in deep learning methods
- Dangerous in safety-critical applications

Adversarial Attacks

Problem Formulation

1. Trained classification network

$$f_{\mathrm{NN}}(\boldsymbol{X}; \{\boldsymbol{\theta}\}) = \arg\max_{i} \underbrace{\operatorname{softmax}(\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}))_i}_{\text{class scores: } F(\boldsymbol{X})}$$

Adversarial Attacks

Problem Formulation

1. Trained classification network

$$f_{\mathrm{NN}}(\boldsymbol{X}; \{\boldsymbol{\theta}\}) = \arg\max_i \underbrace{\mathrm{softmax}(\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}))_i}_{\text{class scores: } F(\boldsymbol{X})}$$

2. Adversarial perturbation

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{\Delta} \in [0,1]^{N \times N}$$

Adversarial Attacks

Problem Formulation

1. Trained classification network

$$f_{\mathrm{NN}}(\boldsymbol{X};\{\boldsymbol{\theta}\}) = \arg\max_i \underbrace{\mathrm{softmax}(\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}))_i}_{\text{class scores: } F(\boldsymbol{X})}$$

2. Adversarial perturbation

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{\Delta} \in [0,1]^{N \times N}$$

3. How do we find $\boldsymbol{\Delta}$?

$$\min_{\boldsymbol{\Delta}} \| \underbrace{\tilde{\boldsymbol{X}} - \boldsymbol{X}}_{\boldsymbol{\Delta}} \|_p \quad \text{s.t.} \quad f_{\mathrm{NN}}(\tilde{\boldsymbol{X}};\{\boldsymbol{\theta}\}) \neq f_{\mathrm{NN}}(\boldsymbol{X};\{\boldsymbol{\theta}\})$$

with $p = 0, 1, 2, \infty$

Adversarial Attacks

Categorization

- Targeted vs untargeted attacks

$$f_{\mathrm{NN}}(\tilde{\boldsymbol{X}}; \{\boldsymbol{\theta}\}) = \hat{y}$$

Adversarial Attacks

Categorization

- Targeted vs untargeted attacks

$$f_{\mathrm{NN}}(\tilde{\boldsymbol{X}};\{\boldsymbol{\theta}\}) = \hat{y}$$

- One-shot vs iterative attacks

Adversarial Attacks

Categorization

- Targeted vs untargeted attacks

$$f_{\mathrm{NN}}(\tilde{\boldsymbol{X}}; \{\boldsymbol{\theta}\}) = \hat{y}$$

- One-shot vs iterative attacks
- White-box vs black-box attacks

Adversarial Attacks

Categorization

- Targeted vs untargeted attacks

$$f_{\mathrm{NN}}(\tilde{\boldsymbol{X}}; \{\boldsymbol{\theta}\}) = \hat{y}$$

- One-shot vs iterative attacks
- White-box vs black-box attacks

In This Work:

1. Fast Gradient Sign Method (FGSM):	(un-)targeted, one-shot, white-box
2. Basic Iterative Method (BIM):	(un-)targeted, iterative, white-box
3. Boundary:	(un-)targeted, iterative, black-box
4. Carlini-Wagner (CW):	(un-)targeted, iterative, white-box

Fast Gradient Sign Method (FGSM) Attack [1]

- Cost function used to train the NN (e.g., cross entropy loss)

$$J(\boldsymbol{X}, y_{\text{true}}, \boldsymbol{\theta})$$

[1] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)

FGSM Attack [1]

- Cost function used to train the NN (e.g., cross entropy loss)

$$J(\boldsymbol{X}, y_{\text{true}}, \boldsymbol{\theta})$$

- Calculate perturbation

$$\boldsymbol{\Delta} = \varepsilon \, \mathrm{sign}(\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, y_{\text{true}}, \boldsymbol{\theta}))$$

with gradient w.r.t. input image $\boldsymbol{X}$ and hyperparameter $\varepsilon > 0$

[1] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)

FGSM Attack [1]

- Cost function used to train the NN (e.g., cross entropy loss)

$$J(\boldsymbol{X}, y_{\text{true}}, \boldsymbol{\theta})$$

- Calculate perturbation

$$\boldsymbol{\Delta} = \varepsilon \operatorname{sign}(\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, y_{\text{true}}, \boldsymbol{\theta}))$$

  with gradient w.r.t. input image $\boldsymbol{X}$ and hyperparameter $\varepsilon > 0$

- Adversarial example

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} + \underbrace{\varepsilon \operatorname{sign}(\nabla_{\boldsymbol{X}} J(\boldsymbol{X}, y_{\text{true}}, \boldsymbol{\theta}))}_{\boldsymbol{\Delta}} \in [0, 1]^{N \times N}$$

[1] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)

Basic Iterative Method (BIM) Attack [2]

- Iterative extension of FGSM

$$\tilde{\boldsymbol{X}}_0 = \boldsymbol{X}$$

$$\tilde{\boldsymbol{X}}_{t+1} = \mathcal{P}_\varepsilon\left(\tilde{\boldsymbol{X}}_t + \underbrace{\alpha\,\mathrm{sign}(\nabla_{\tilde{\boldsymbol{X}}_t} J(\tilde{\boldsymbol{X}}_t, y_{\text{true}}, \boldsymbol{\theta}))}_{\boldsymbol{\Delta}_t}\right)$$

for $t = 0, \ldots, T$ and step-size $\alpha > 0$ with $\alpha T = \varepsilon$

[2] Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)

BIM Attack [2]

- Iterative extension of FGSM

$$\tilde{\boldsymbol{X}}_0 = \boldsymbol{X}$$

$$\tilde{\boldsymbol{X}}_{t+1} = \mathcal{P}_\varepsilon \left(\tilde{\boldsymbol{X}}_t + \underbrace{\alpha \operatorname{sign}(\nabla_{\tilde{\boldsymbol{X}}_t} J(\tilde{\boldsymbol{X}}_t, y_{\text{true}}, \boldsymbol{\theta}))}_{\boldsymbol{\Delta}_t} \right)$$

  for $t = 0, \dots, T$ and step-size $\alpha > 0$ with $\alpha T = \varepsilon$
- $\mathcal{P}_\varepsilon$ projects the current iterate back onto a ε-L_p ball around $\boldsymbol{X}$

[2] Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)

Boundary Attack [3]

- Black-box attack (no gradients necessary, only model evaluations)
- Iterative method starting with $[\boldsymbol{\Delta}_0]_{i,j} \sim \mathcal{U}(0,1)$

$$\tilde{\boldsymbol{X}}_0 = \boldsymbol{\Delta}_0 \text{ (must be misclassified)}$$
$$\tilde{\boldsymbol{X}}_{t+1} = \tilde{\boldsymbol{X}}_t + \boldsymbol{\Delta}_{t+1}$$

for $t = 0, \ldots, T-1$

[3] Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=SyZI0GWCZ

Boundary Attack [3]

- Black-box attack (no gradients necessary, only model evaluations)
- Iterative method starting with $[\boldsymbol{\Delta}_0]_{i,j} \sim \mathcal{U}(0,1)$

$$\tilde{\boldsymbol{X}}_0 = \boldsymbol{\Delta}_0 \text{ (must be misclassified)}$$
$$\tilde{\boldsymbol{X}}_{t+1} = \tilde{\boldsymbol{X}}_t + \boldsymbol{\Delta}_{t+1}$$

for $t = 0, \ldots, T-1$

- Perturbations calculated by random walk along the boundary with conditions
  1. $\tilde{\boldsymbol{X}}_{t+1} \in [0,1]^{N \times N}$
  2. $\frac{\|\boldsymbol{\Delta}_{t+1}\|_F}{d(\tilde{\boldsymbol{X}}_t, \boldsymbol{X})} = \gamma$ (The perturbation has a specific relative size.)
  3. $\frac{d(\tilde{\boldsymbol{X}}_t, \boldsymbol{X}) - d(\tilde{\boldsymbol{X}}_{t+1}, \boldsymbol{X})}{d(\tilde{\boldsymbol{X}}_t, \boldsymbol{X})} = \nu$ (The distance is decreased by a realtive amount.)

with a distance metric d and hyperparameters $\gamma, \nu > 0$.

[3] Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=SyZI0GWCZ

Carlini-Wagner (CW) Attack [4]

- Solve constrained optimization problem

$$\min_{\boldsymbol{\Delta}} \|\boldsymbol{\Delta}\|_p + c \cdot \max\left\{\max_{j \neq y_{\text{true}}} [\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}+\boldsymbol{\Delta})]_j - [\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X})]_{y_{\text{true}}}, -\kappa\right\}$$
$$\text{s.t.} \quad \boldsymbol{X}+\boldsymbol{\Delta} \in [0,1]^{N \times N}$$

  where $c, \kappa > 0$
- Control confidence with κ

[4] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)

CW Attack [4]

- Solve constrained optimization problem

$$\min_{\boldsymbol{\Delta}} \|\boldsymbol{\Delta}\|_p + c \cdot \max\left\{\max_{j \neq y_{\text{true}}} [\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}+\boldsymbol{\Delta})]_j - [\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X})]_{y_{\text{true}}}, -\kappa\right\}$$

$$\text{s.t.} \quad \boldsymbol{X}+\boldsymbol{\Delta} \in [0,1]^{N \times N}$$

  where $c, \kappa > 0$
- Control confidence with κ
- Introduce auxiliary variable $\boldsymbol{W}$ where

$$\boldsymbol{\Delta} = \frac{1}{2}(\tanh(\boldsymbol{W}) + 1) - \boldsymbol{X}$$

[4] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)

CW Attack [4]

- Solve constrained optimization problem

$$\min_{\boldsymbol{\Delta}} \|\boldsymbol{\Delta}\|_p + c \cdot \max\left\{\max_{j \neq y_{\text{true}}} [\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}+\boldsymbol{\Delta})]_j - [\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X})]_{y_{\text{true}}}, -\kappa\right\}$$
$$\text{s.t.} \quad \boldsymbol{X}+\boldsymbol{\Delta} \in [0,1]^{N \times N}$$

  where $c, \kappa > 0$
- Control confidence with κ
- Introduce auxiliary variable $\boldsymbol{W}$ where

$$\boldsymbol{\Delta} = \frac{1}{2}(\tanh(\boldsymbol{W}) + 1) - \boldsymbol{X}$$

- Solve unconstrained optimization problem w.r.t. $\boldsymbol{W}$

[4] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)

Adversarial Defences

Categorization

- Gradient masking: prevent gradient computation with respect to the inputs by training models to obfuscate gradients

Adversarial Defences

Categorization

- Gradient masking: prevent gradient computation with respect to the inputs by training models to obfuscate gradients
- Adversarial training: robustify models by including adversarial images in the training set

Adversarial Defences

Categorization

- Gradient masking: prevent gradient computation with respect to the inputs by training models to obfuscate gradients
- Adversarial training: robustify models by including adversarial images in the training set
- Adversarial example detection: train an additional classifier to decide whether an input image is adversarial or not

Adversarial Defences

Categorization

- Gradient masking: prevent gradient computation with respect to the inputs by training models to obfuscate gradients
- Adversarial training: robustify models by including adversarial images in the training set
- Adversarial example detection: train an additional classifier to decide whether an input image is adversarial or not

Benefits of Adversarial Attack Detection

- Post-hoc approach: no influence on model training
- Easy to implement

Detecting Adversarial Examples

$$f_{\mathrm{NN}}(\boldsymbol{X}; \{\boldsymbol{\theta}\}) = \arg\max_i \underbrace{\mathrm{softmax}(\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}))_i}_{\text{class scores: } F(\boldsymbol{X})}$$

Detecting Adversarial Examples

$$f_{\mathrm{NN}}(\boldsymbol{X};\{\boldsymbol{\theta}\}) = \arg\max_i \underbrace{\mathrm{softmax}(\boldsymbol{z}^{(\mathrm{NN})}(\boldsymbol{X}))_i}_{\text{class scores: } F(\boldsymbol{X})}$$

Our Detection Method

1. Construct adversarial image set $\mathcal{X}_{\text{adv}} = \{\tilde{\boldsymbol{X}}_1, \ldots, \tilde{\boldsymbol{X}}_M\}$ from training set $\mathcal{X}_{\text{train}} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M\}$
2. Train Support Vector Machine (SVM) T_{SVM} on normalized class scores training set

$$\mathcal{X}_{\text{scores}} = \{(F(\boldsymbol{X}_1), +1), \ldots, (F(\boldsymbol{X}_M), +1), \\ (F(\tilde{\boldsymbol{X}}_1), -1), \ldots, (F(\tilde{\boldsymbol{X}}_M), -1)\}$$

3. At test time use T_{SVM} to detect adversarial examples based on class scores

Evaluation

Experimental Setup

- CIFAR 10 dataset
- Three pre-trained classification models (VGG-Net, GoogLeNet, ResNet)
- Four untargeted adversarial attacks (FGSM, BIM, Boundary, CW)
- Combinations of two attacks (CW+BIM, CW+FGSM, Boundary+BIM, Boundary+FGSM)

[5] Kwon, H., Kim, Y., Yoon, H., Choi, D.: Classification score approach for detecting adversarial example in deep neural network. Multimedia Tools and Applications80(7), 10339–10360 (2021)

Evaluation

Experimental Setup

- CIFAR 10 dataset
- Three pre-trained classification models (VGG-Net, GoogLeNet, ResNet)
- Four untargeted adversarial attacks (FGSM, BIM, Boundary, CW)
- Combinations of two attacks (CW+BIM, CW+FGSM, Boundary+BIM, Boundary+FGSM)

Reference Algorithm

- Kwon et al. [5] : Threshold the difference between the largest and second largest normalized class scores. Threshold is learned using a decision stump.

---

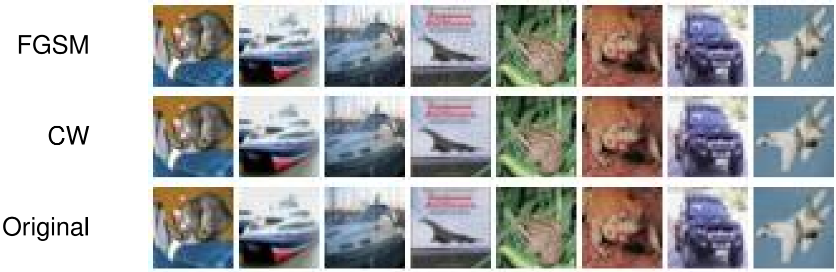
[5] Kwon, H., Kim, Y., Yoon, H., Choi, D.: Classification score approach for detecting adversarial example in deep neural network. Multimedia Tools and Applications80(7), 10339–10360 (2021)

Attack Results

	Accuracy on adversarial examples			Average perturbation norm		
Attack	VGG19	GoogLeNet	ResNet18	VGG19	GoogLeNet	ResNet18
FGSM	39.97%	39.85%	40.18%	17.6232	0.2575	2.7183
BIM	5.17%	4.29%	4.49%	8.9903	0.0484	0.2303
Boundary	8.99%	25.75%	1.39%	0.0515	0.0209	0.0849
CW	4.75%	0.55%	0.30%	0.2461	0.0140	0.0559
Orig. Acc.	93.95%	92.85%	93.07%	–	–	–

Attack Results

	Accuracy on adversarial examples			Average perturbation norm		
Attack	VGG19	GoogLeNet	ResNet18	VGG19	GoogLeNet	ResNet18
FGSM	39.97%	39.85%	40.18%	17.6232	0.2575	2.7183
BIM	5.17%	4.29%	4.49%	8.9903	0.0484	0.2303
Boundary	8.99%	25.75%	1.39%	0.0515	0.0209	0.0849
CW	4.75%	0.55%	0.30%	0.2461	0.0140	0.0559
Orig. Acc.	93.95%	92.85%	93.07%	–	–	–



Detection Results (Single Attacks)

Model	Attack	Accuracy		F_1 score	
		Kwon et al. [5]	Ours	Kwon et al. [5]	Ours
VGG19	FGSM	71.60%	**82.08%**	68.43%	**82.05%**
	BIM	85.20%	**98.70%**	84.47%	**98.69%**
	Boundary	**97.53%**	96.30%	**97.44%**	96.25%
	CW	89.90%	**90.05%**	89.99%	**90.16%**
GoogLeNet	FGSM	72.60%	**76.05%**	73.69%	**74.48%**
	BIM	81.50%	**83.60%**	77.88%	**82.38%**
	Boundary	**96.50%**	95.50%	**96.35%**	95.45%
	CW	93.65%	**93.80%**	93.58%	**93.76%**
ResNet18	FGSM	70.40%	**72.58%**	69.23%	**71.37%**
	BIM	85.48%	**89.48%**	83.68%	**88.96%**
	Boundary	**97.20%**	96.28%	**97.10%**	96.19%
	CW	93.53%	**93.58%**	93.63%	**93.65%**

[5] Kwon, H., Kim, Y., Yoon, H., Choi, D.: Classification score approach for detecting adversarial example in deep neural network. Multimedia Tools and Applications80(7), 10339–10360 (2021)

Detection Results (Multiple Attacks)

Model	Attack	Accuracy: Kwon et al. [5]	Accuracy: Ours	F_1 Score: Kwon et al. [5]	F_1 Score: Ours
VGG19	CW+BIM	67.38%	**89.90%**	54.80%	**90.08%**
	CW+FGSM	80.75%	**83.65%**	79.90%	**83.14%**
	Boundary+BIM	73.45%	**95.88%**	63.73%	**95.85%**
	Boundary+FGSM	82.45%	**85.80%**	81.92%	**84.85%**
GoogLeNet	CW+BIM	70.93%	**84.08%**	59.66%	**83.92%**
	CW+FGSM	79.68%	**82.35%**	79.28%	**81.37%**
	Boundary+BIM	73.60%	**84.93%**	63.89%	**84.57%**
	Boundary+FGSM	78.93%	**80.93%**	78.53%	**79.58%**
ResNet18	CW+BIM	70.45%	**88.30%**	60.49%	**88.40%**
	CW+FGSM	78.68%	**79.33%**	79.03%	**79.52%**
	Boundary+BIM	72.73%	**90.05%**	62.16%	**89.61%**
	Boundary+FGSM	77.93%	**78.85%**	**78.31%**	77.76%

[5] Kwon, H., Kim, Y., Yoon, H., Choi, D.: Classification score approach for detecting adversarial example in deep neural network. Multimedia Tools and Applications 80(7), 10339–10360 (2021)

Conclusion

Conclusion

- Detecting adversarial attacks only by looking at the class score distribution
- Empirical evaluation of various state-of-the-art adversarial attacks on different classification models
- Improved class score based adversarial attack detection
- The proposed detection method can detect mixtures of attacks

Thank you for your attention!

Any questions?