Introduction of
Machine Learning Security

Lecture 02

Evasion Attack

Patrick Chan
patrickchan@scut.edu.cn

Agenda

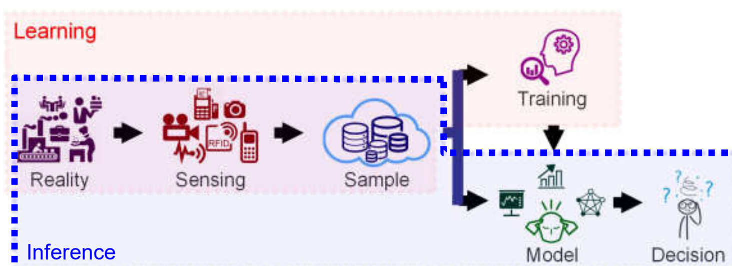
- Formulation: Attack Loss / Attack Cost
- Attack Sample Crafting
- Imperfect Knowledge
  - Surrogate Model
  - Query Attack
- Defense
  - Pre-processing
  - Robust Model

Evasion Attack

- Bypass a defensive system by modifying samples
- General speaking, evasion attack misleads trained systems by camouflaging samples in the inference phase

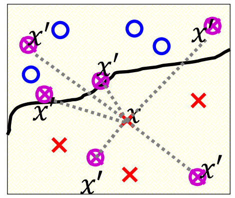
Evasion Attack

- How to mislead a trained model?



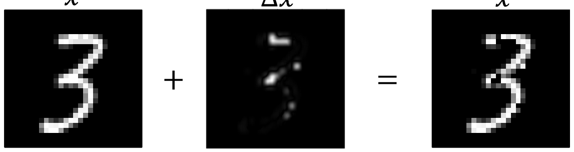
Evasion Attack

- Mislead the decision of a trained classifier by manipulating a sample in the inference phase
- How to determine Δx?



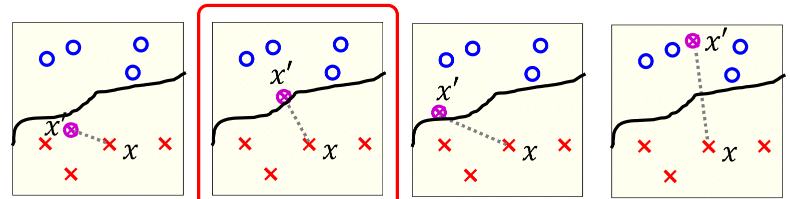
3 VS 7 Classification

	Original Sample x		Δx		Attack Sample x'
		+		=	
	Classified as 3				Classified as 7



Evasion Attack

- Which attack is better?



Successful Attack?	No	Yes	Yes	Yes
Manipulation	Small Perturbation	Small Perturbation	Large Perturbation	Very Large Perturbation

Objective Function

- Two factors of sample crafting
  - ↑ **Attack Impact**: Influence to the output
  - ↓ **Attack Cost**: Change on a sample
- Formulate as a multi-objective optimization

$$\min_{\Delta x}\left(L(x+\Delta x, t, f_w), \|\Delta x\|\right)$$

Loss between the output of the target model on attack sample and the target class (How close to your expected attack)

Change of a sample

Δx : manipulation $\quad$ f_w: the target model

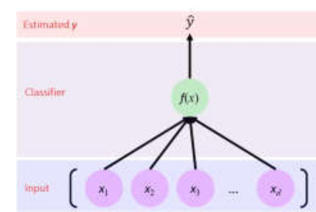
t : the target class $\quad$ x : the original sample

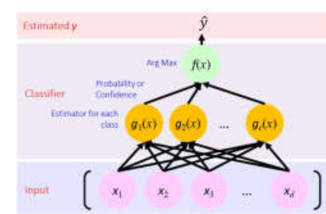
Objective Function

Attack Loss

$$\min_{\Delta x}\left(L(x+\Delta x, t, f_w), \|\Delta x\|\right)$$

- **2-Class problem**
  - The target class is obvious
    - Class 1 > Class 2 or Class 2 > Class 1
- **Multi-Class problem**
  - Generic Attack
    - Misclassification
      - Any class different from the original one
      - Usually the class which is most easily misled
  - Class-specific Attack
    - Selected target class





Objective Function
Attack Loss

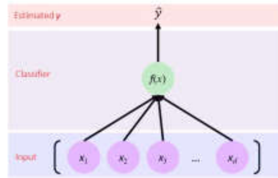
$\min_{\Delta x}(L(x+\Delta x, t, f_w), \|\Delta x\|)$

- **Loss function (L)**
  1. Confidence value of the target class
     $$-\big(g_t(x+\Delta x)\big)$$
  2. Difference between the confidence values between the target class and another one with the largest confidence value
     $$-\Big(g_t(x+\Delta x) - \max_{i \neq t} g_i(x+\Delta x)\Big)$$

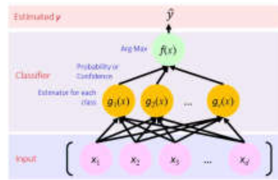
t: the target class

g_i: the estimated confidence output of the class i

Two-Class Problem



Multi-Class Problem

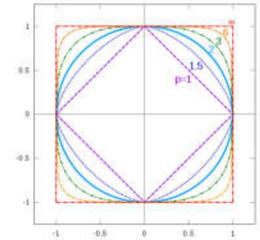


Objective Function
Attack Cost

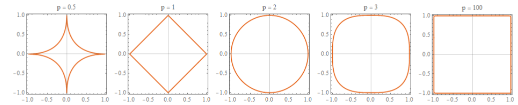
$\min_{\Delta x}(L(x+\Delta x, t, f_w), \|\Delta x\|)$

- ***p*-norms $\|x\|_p$**

$$\|x\|_p = \left(\sum_{i=1}^{d} |x_i|^p\right)^{\frac{1}{p}} \quad \text{where } p \geq 1$$



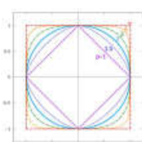
- $\|x\|_0$ = number of non-zero elements (not convex)
  - limit the number of attack feature (sparse attack)
- $\|x\|_1 = |x_1| + |x_2| + \ldots + |x_d|$
- $\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \ldots + |x_d|^2}$
- $\|x\|_\infty = \max_{1 \le i \le d} |x_i|$
  - Minimize the maximum change to any features (dense attack)

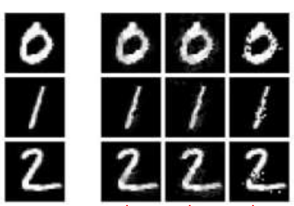


Objective Function
Attack Cost

$\min_{\Delta x}(L(x+\Delta x, t, f_w), \|\Delta x\|)$

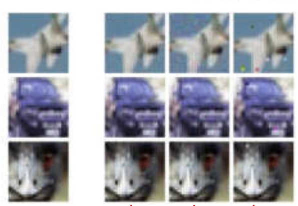
- Different *p*-norm functions on the adversarial noise ($\Delta x = \|x - x'\|$) generate different x'





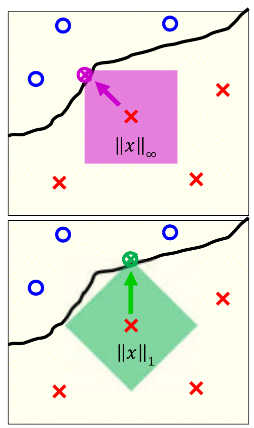
L_∞ L_2 L_0

Original Image | Dense Attack | Sparse Attack



L_∞ L_2 L_0

Original Image | Dense Attack | Sparse Attack



Objective Function
Attack Cost

$\min_{\Delta x}(L(x+\Delta x, t, f_w), \|\Delta x\|)$

- One-Pixel / Few-Pixel Attack

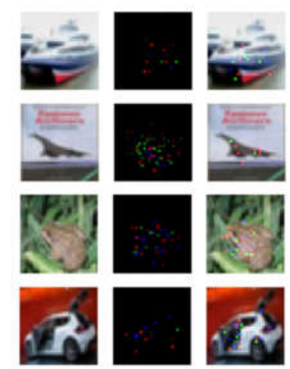
$$\min \quad f(x + e(x))$$
$$\text{s.t.} \quad \|e(x)\|_0 \leq d$$

$\|e(x)\|_0$: count non-zero elements

$e(x) = (e_1, e_2, \ldots, e_n)$: n is number of features

d: number of modified features

One-Pixel Attack
***d* = 1**



Few-Pixel Attack
***d* > 1**

J Su, DV Vargas, K Sakurai(2019) One pixel attack for fooling deep neural networks. In: IEEE Transactions on Evolutionary Computation

Objective Function
Attack Cost

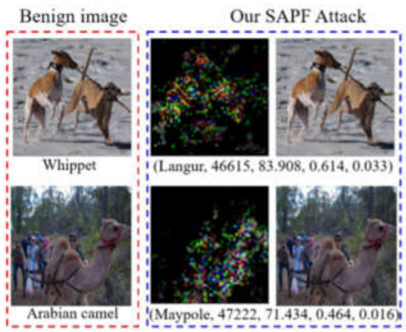
$\min_{\Delta x}(L(x + \Delta x, t, f_w), \|\Delta x\|)$

- Control the attack features and their magnitudes separately and precisely

$$\min_{\boldsymbol{\delta},\mathbf{G}} \quad \underbrace{\| \boldsymbol{\delta} \odot \mathbf{G} \|_2^2}_{\substack{\text{Adversarial Noise} \\ (\Delta x = \boldsymbol{\delta} \odot \mathbf{G})}} + \lambda \mathcal{L}(f(\mathbf{x} + \boldsymbol{\delta} \odot \mathbf{G}), y_t),$$

$$s.t. \quad \underbrace{\mathbf{1}^{\top}\mathbf{G} = k}_{\substack{\text{no more than} \\ \text{k features}}}, \ \mathbf{G} \in \{0,1\}^d$$

- $\delta \in R^d$: vector of perturbation magnitudes
- $G \in \{0,1\}^d$:vector of perturbed positions

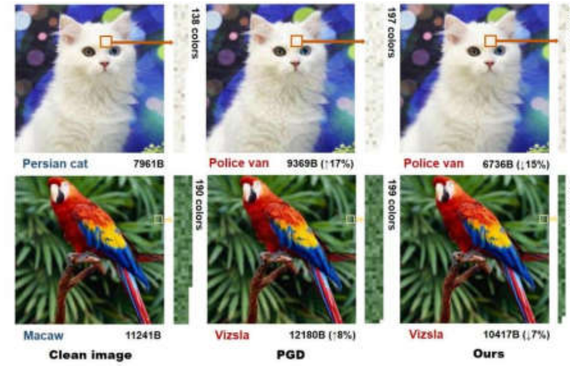


Y Fan, B Wu, T Li(2020) Sparse Adversarial Attack via Perturbation Factorization. In: ECCV

Objective Function
Attack Cost

$\min_{\Delta x}(L(x + \Delta x, t, f_w), \|\Delta x\|)$

- Most adversarial attacks add extra disturbing information on clean images explicitly
- AdvDrop attacks by dropping existing information of images

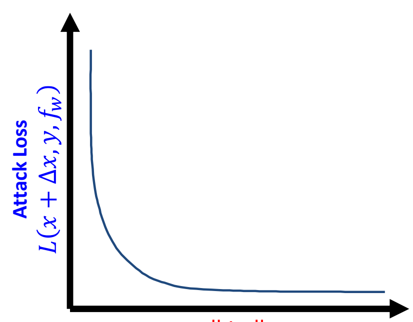


Duan, R., Chen, Y., Niu, D., Yang, Y., Qin, A. K., & He, Y. (2021). Advdrop: Adversarial attack to dnns by dropping information. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7506-7515)

Objective Function
Tradeoff

$\min_{\Delta x}(L(x + \Delta x, t, f_w), \|\Delta x\|)$

- **Attack Impact** and **Attack Cost** are correlated
  - Smaller sample change yields larger attack loss, vice versa
  - Smaller attack loss yields larger sample change, vice versa

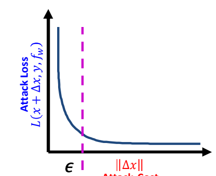


Objective Function
Formulation

$\min_{\Delta x}(L(x + \Delta x, t, f_w), \|\Delta x\|)$

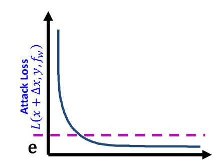
- Multi-objective problem can be formulated as
  1. **Minimize Attack Loss**
     - Maximize damage with a fixed attack cost

$$\begin{aligned} \min \quad & L(x + \Delta x, t, f_w) \\ \text{s.t.} \quad & \|\Delta x\| \ \epsilon \geq \end{aligned}$$



  2. **Minimize Attack Cost**
     - Minimize the attack cost for an attack damage

$$\begin{aligned} \min \quad & \|\Delta x\| \\ \text{s.t.} \quad & L(x + \Delta x, t, f_w) \ \geq \text{e} \end{aligned}$$



Objective Function

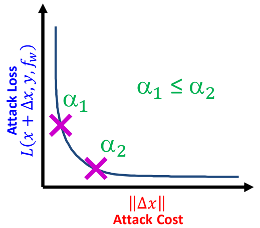
Formulation

$\min_{\Delta x}(L(x + \Delta x, t, f_w), \|\Delta x\|)$

- Multi-objective problem can be formulated as
  3. **Tradeoff Solution**
     - Maximize damage with a fixed attack cost

$$\min \ \alpha \, L(x + \Delta x, t, f_w) + (1 - \alpha) \, \|\Delta x\|$$

α : a tradeoff parameter ($0 \leq \alpha \leq 1$)



- When $\alpha = 1$, only $L(x + \Delta x, t, f_w)$ is focused
- When $\alpha = 0$, only $\|\Delta x\|$ is focused

Attack Sample Crafting

Gradient Descent

- Algorithm

$\Delta x_0 = 0$ *Initialize delta x*

i = 0 *Initialize counter i*

Do

i = i + 1 *Counting*

$\Delta x_{i+1} = \Delta x_i - \alpha \nabla L(x + \Delta x_i, t, f_w)$ *Update Δx_{i+1} according to gradient at $x + \Delta x_i$*

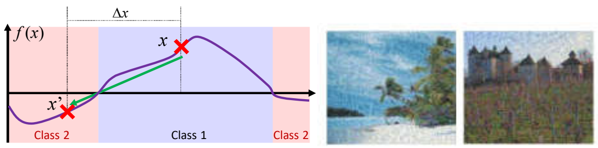
$\Delta x_{i+1} = \text{constraint}(\Delta x_{i+1})$ *Limit Δx_{i+1} by constraints*

While i <= n *Update n times*

Attack Sample Crafting

Gradient Descent

- **One-Step Method**
  Fast Gradient Sign Method (FGSM)
  - n = 1 : Cost ↓



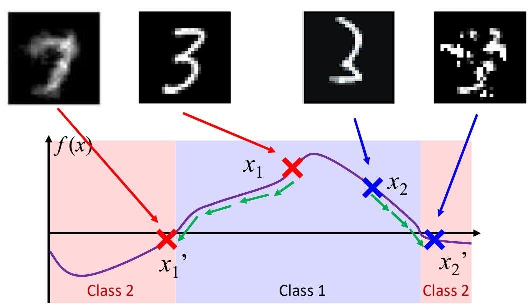
- **Multi-Step Method**
  Projected Gradient Descent(PGD)
  - n > 1 : Cost ↑

IJ Goodfellow, J Shlens, C Szegedy(2014) Explaining and harnessing adversarial examples. In: arXiv
A Madry, A Makelov, L Schmidt(2017) Towards Deep Learning Models Resistant to Adversarial Attacks. In: arXiv

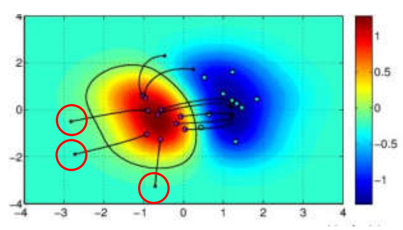
Attack Sample Crafting

Look Natural?

- Changing $g(x)$ is good enough?



Crafted samples are very different from the real samples of another class



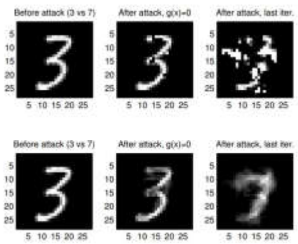
Attack Sample Crafting

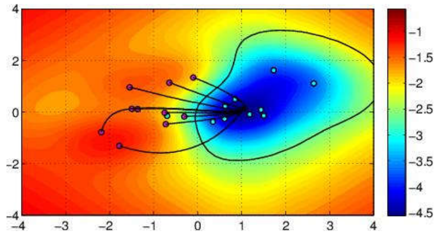
Look Natural? Density Estimator

- Not only cross the decision boundary but also close to the target class
  - Penalize x' in low density regions

Mislead the model to classify +1 as -1

$$\begin{aligned} \min \quad & f(x') - \boxed{\hat{p}(x'|y=-1)} \\ \text{s.t.} \quad & \|x - x'\| \le \varepsilon \end{aligned}$$



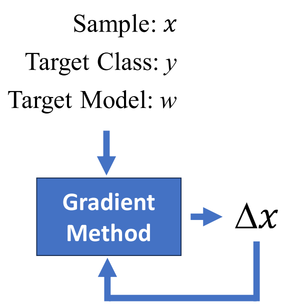


B Biggio, I Corona, D Maiorca(2013) Evasion Attacks against Machine Learning at Test Time. In: ECML

Attack with Limited Knowledge

Limited Knowledge

- Gradient method for crafting is available only if the target model w is known (which is the most secure information)
  - i.e. $\nabla_{\Delta x} L_w(x+\Delta x, t)$
- What can we do when the detail of target model is unknown?
  - No gradient can be computed

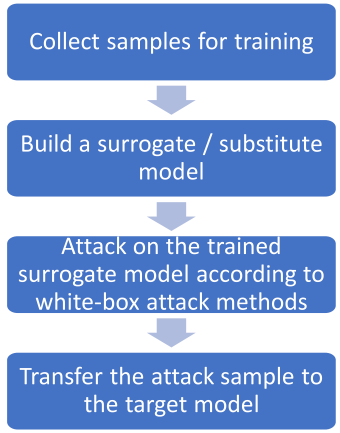


Methods

1. **Surrogate Model + Attack Transfer**
   - Assumption: Samples for training can be somehow obtained
   - Train a surrogate model to replace the target one
2. **Query Attack**
   - Assumption: the target model can be accessed
   - Craft attack samples by querying the target models

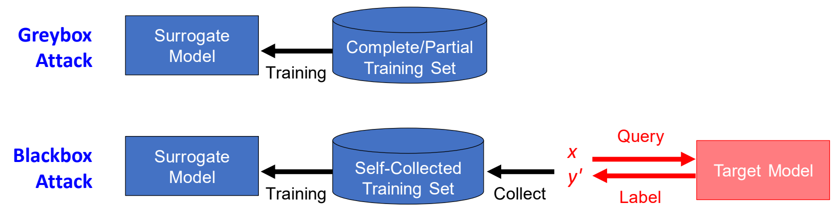
Procedure

- Aim to approximate the target model
- The surrogate model should be convenient for crafting
  - i.e. differentiable
- More information yields better approximation
- Key Challenge: Transferability



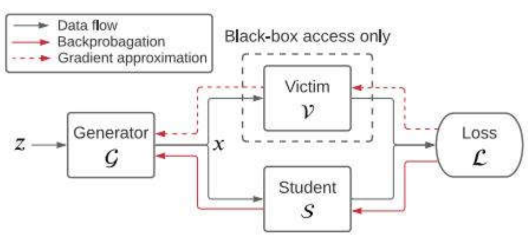
Incomplete Training Set

- Build Surrogate / Substitute model by incomplete training set
  - Greybox Attack: Complete / Partial training set
  - Blackbox Attack: Self-collected data in the same application (but maybe different from the training samples used in target model)
    - Query of the targeted model is allowed



No Training Set

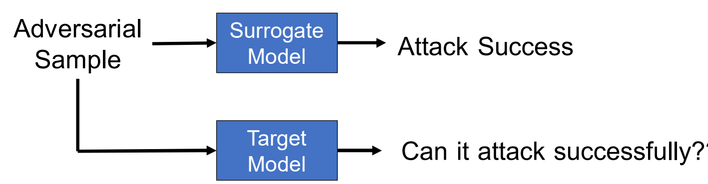
- Data-Free Model Extraction
  - **Student** aims to approximate to Victim (target model) : reduce Loss
  - **Generator** generates samples to increase the difference between victim and student : increase Loss
  - Student and Generator are in opposition



Truong, J. B., Maini, P., Walls, R. J., & Papernot, N. (2021). Data-free model extraction. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.

Transferability

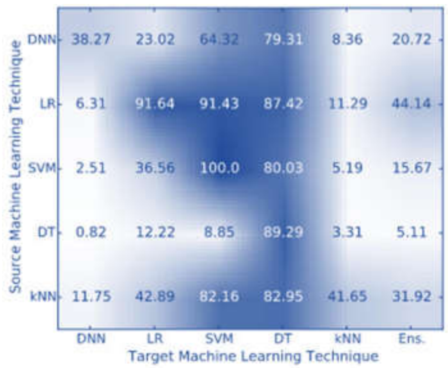
- **Attack Transferability** is important requirement
  - Reduce the influence of difference between the real and surrogate models on attack performance



Attack Method: Surrogate Model

Transferability

- Experimental results on **attack performance** of attack samples generated by Surrogate (Source) model on Target model
  - DNN: Deep Neural Network
  - LR: Linear Regression
  - SVM: Support Vector Machine
  - DT: Decision Tree
  - kNN: k nearest neighborhood
- Different models behave differently



Papernot et al., Practical Black-box Attacks against Machine Learning, AsiaCCS 2017

Attack Method: Surrogate Model

Transferability

- Transferability Factors
  - **Size of Input Gradient (↑)**
    - Larger gradient yields large modification, usually generate larger attack impact
  - **Gradient Alignment (↑)**
    - Larger similarity of the input gradients of the loss function of the target and surrogate models is better for attack
  - **Variability of the loss landscape (↓)**
    - Surrogate loss functions that are stabler and lower variance may find better a local optima (better attack)

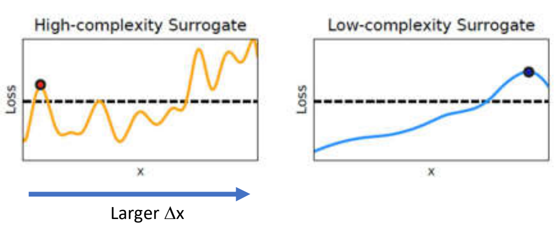
Demontis, Biggio et al., Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks, USENIX 2019

Attack Method: Surrogate Model

Transferability

- Model complexity is important to transferability
- **Two main factors**
  - the complexity of the target model
  - the complexity of the surrogate model

Red Point: Attack Sample crafted according to High-Complexity Surrogate
Blue Point: Attack Sample crafted according to Low-Complexity Surrogate



Successful Attack
Failed Attack

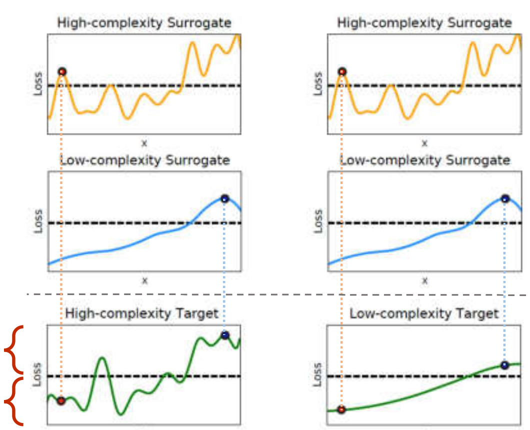
Below a certain threshold (i.e., the dotted line), the point is correctly classified, otherwise it is misclassified

Demontis, Biggio et al., Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks, USENIX 2019

Attack Method: Surrogate Model

Transferability

- Attack Sample crafted according to High-Complexity Surrogate may not able to fool the target model



Red Point: Attack Sample crafted according to High-Complexity Surrogate
Blue Point: Attack Sample crafted according to Low-Complexity Surrogate

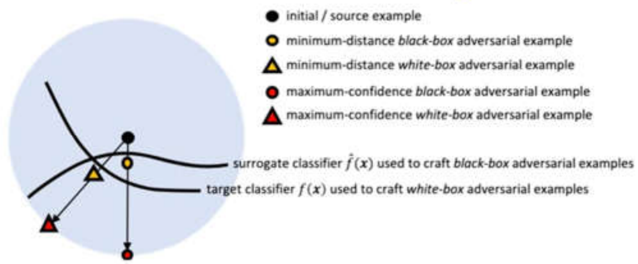
Successful Attack
Failed Attack

Demontis, Biggio et al., Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks, USENIX 2019

Attack Method: Surrogate Model

Transferability

- **Maximum Confidence Attacks**
  might better transfer, but more perturbation is needed
- **Minimum Distance Attacks**
  are likely to fail because decision boundary is different



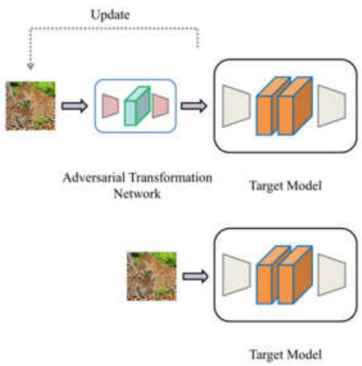
Demontis, Biggio et al., Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks, USENIX 2019

Attack Method: Surrogate Model: Transferability

Enhance Transferability

- Image transformation of a model may reduce attack ability
- Adversarial Transformation Network (ATN) is built aiming to lose effect of image transformation on sample attack
- Attack sample is crafted to attack a model with / without ATN

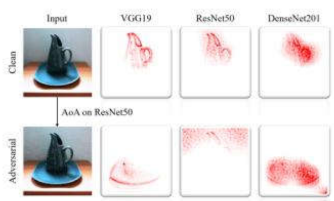
$$L_{attack} = J(f(\mathbf{x}^{adv}), y) + \gamma J(f(T(\mathbf{x}^{adv})), y)$$



Attack Method: Surrogate Model: Transferability

Enhance Transferability

- Many studies indicate that attention heatmaps of models on an object are similar
- Aim at manipulating the heatmap
  - Magnitude suppression $L_{\text{supp}}(x) = \|h(x, y_{\text{ori}})\|_1$
  - Distraction $L_{\text{dstc}}(x) = -\left\| \frac{h(x, y_{\text{ori}})}{max(h(x, y_{\text{ori}}))} - \frac{h(x_{\text{ori}}, y_{\text{ori}})}{max(h(x_{\text{ori}}, y_{\text{ori}}))} \right\|_1$
  - Decrease the gap between 1st and 2nd largest classes $L_{\text{bdry}}(x) = \|h(x, y_{\text{ori}})\|_1 - \|h(x, y_{\text{sec}}(x))\|_1$

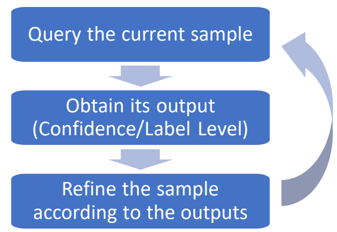


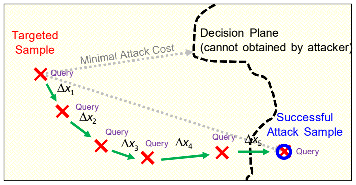
Sizhe Chen, Zhengbao He, Chengjin Sun(2020) Universal adversarial attack on attention and the resulting dataset damagenet. In: IEEE Transactions on Pattern Analysis and Machine Intelligence

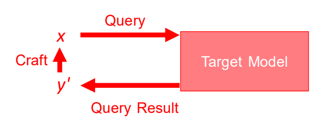
Attack Method: Query Attack

Procedure

- Assume the target model can be queried
- More information provided by the model (Confidence output > Label output) enhance the crafting
- Key Challenges:
  - Reduce the query number
  - Minimize the modification

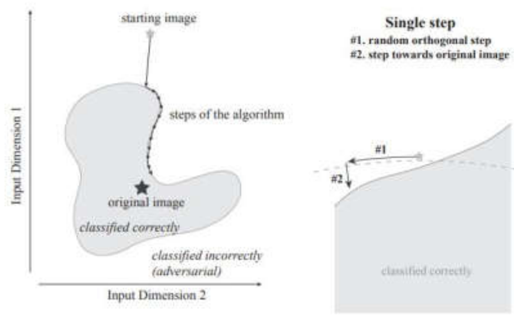




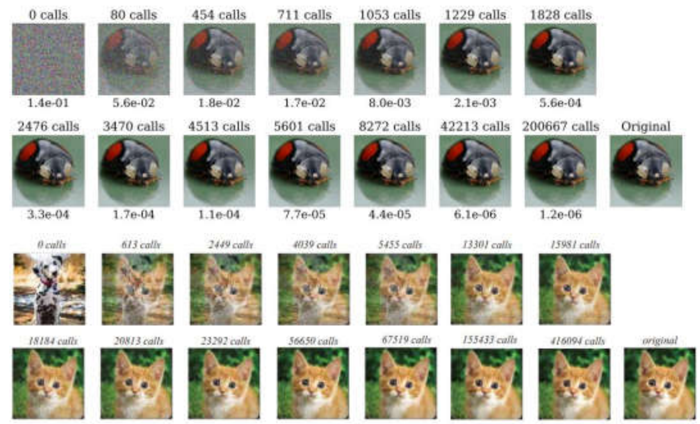


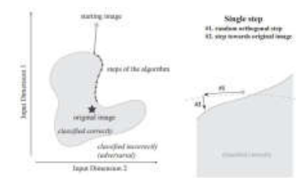
Decision-Based Adversarial Attacks

- Start from a randomly selected a point in the target class
- Moving the point toward to the original image until touch the boundary
- Get closer to the original image by searching on the surface of the target class region



Decision-Based Adversarial Attacks





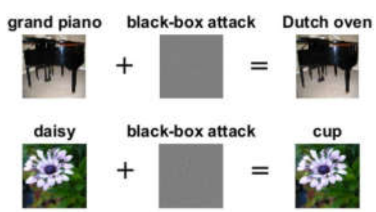
ZOO: Zeroth order attack

- Estimate the second-order partial derivative without surrogate
- Querying the model around a very small proximity for each dimension

First-Order Partial Derivative $\hat{g}_i := \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f(\mathbf{x}+h\mathbf{e}_i) - f(\mathbf{x}-h\mathbf{e}_i)}{2h},$

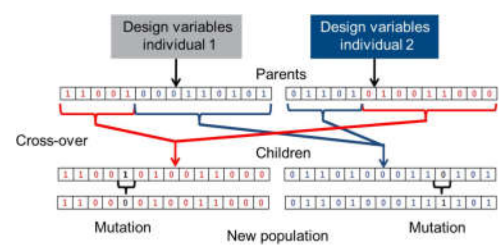
Second-Order Partial Derivative $\hat{h}_i := \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_{ii}^2} \approx \frac{f(\mathbf{x}+h\mathbf{e}_i) - 2f(\mathbf{x}) + f(\mathbf{x}-h\mathbf{e}_i)}{h^2}.$

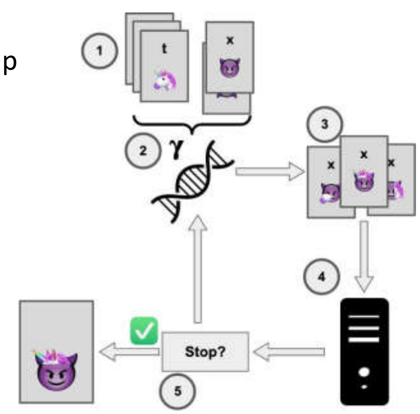
h : a small constant (e.g. h = 0.0001)
e_i : a standard basis vector with only the i-th component as 1



Genetic Algorithms

- Maintain M best samples as a pool
  - Cross-over two randomly selected samp
  - Mutation
  - Query scores for new samples
  - Stop until a solution is found





Attack Evaluation

Attack Loss

- **Attack success rate (ASR)**
  - Ratio of successful evasion when the sample change is fixed
  - Larger ASR, better attack performance

$$ASR = \frac{SAN}{AN}$$

SAN: Successful Attack Number
AN : Attack Number

- **Average Confidence for Adversarial Class (ACAC)**
  - The average probability of all misclassification categories for all samples when the attack is successful
  - Larger ACAC, better attack performance

$$ACAC = \frac{1}{n}\sum_{i=1}^{n} P(X_i^a)_{F(X_i^a)}$$

$F(X_i^a)$: the i-th sample being classified as category a
$P(X_i^a)$: the probability that the i-th sample is classified as category a

Attack Loss

- **Peak signal-to-noise ratio (PSNR)**
  - Calculates the signal-to-noise ratio of original and attack images after successful evasion
  - Larger PSNR, better attack performance

$$PSNR = 10\, log_{10} \frac{M^2}{MSE(x, x')}$$

M : the maximum pixel value
x : the clean image
X’ : the generated image
MSE : the mean square error

Attack Cost

- **Frechet Inception Distance (FID)**
  - Distance between the original and attack image when successful evasion
  - Lower FID, better attack performance

$$FID = \|\mu_x - \mu_{x'}\|_2^2 + Tr\left(\Sigma_x + \Sigma_{x'} \pm 2(\Sigma_x\Sigma_{x'})^{\frac{1}{2}}\right)$$

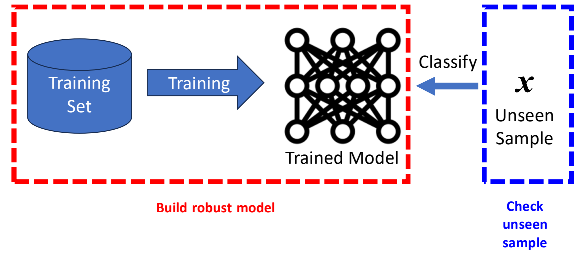
x : the clean image
x’ : the generated image
μ_a : mean of a
Σ_a : covariance matric of a
Tr : sum of the diagonal elements of the matrix

- **Learned Perceptual Image Patch Similarity (LPIPS)**
  - Distance between the original and attack image in the feature space at multiple layers of a pretrained network.
  - Lower LPIPS, higher perceived similarity, better attack performance

Defense Methods

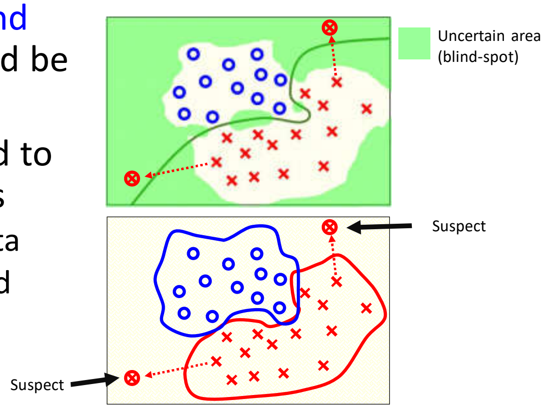
Defense

- Pre-processing
  - Detection
  - Cleaning
- Robust Learning
  - Parameter Updating
  - Structure Selection



Defense: Pre-Processing

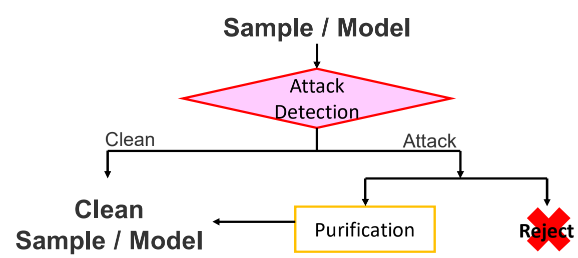
- Characteristics of clean and adversarial samples should be different
- Adversarial examples tend to occur in uncertain regions
  - Far away from training data
  - No information is provided



Defense: Pre-Processing

Detection: Sample

- Explicit Method
  - Evaluation criteria
- Implicit Method
  - Rely on a classifier



Explicit Detection: Sample

- Trained Autoencoder (AE) is used to reconstruct a sample by clean samples
  - **Reconstruction Error**
    - Reconstruction Error on an adversarial sample is larger
    - x is an attack when the error is large
  - **Probability Divergence**
    - Jensen-Shannon divergence (JSD) quantifies the similarity between x and its reconstruction in the feature space (f)
    - x is an attack when JSD is large

$$\|x - AE(x)\|_p$$

$$JSD\left(f(x) \parallel f\big(AE(x)\big)\right)$$

$$JSD(P \parallel Q) = \frac{1}{2}D(P \parallel M) + \frac{1}{2}D(M \parallel Q)$$

Kullback–Leibler Divergence $D_{KL}(P \parallel Q) = \sum_{x \in X} P(x) log\left(\frac{P(x)}{Q(x)}\right)$

a mixture distribution of P and Q $M = \frac{1}{2}(P + Q)$

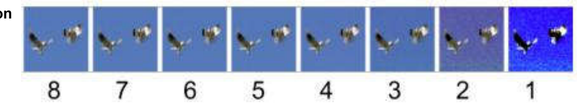
Meng D, Chen H(2017) Magnet: a two-pronged defense against adversarial examples. In: ACM SIGSAC conference on computer and communications security

Explicit Detection: Sample

- Squeezed Output Difference
  - Difference of output on original and transformed images
  - Two kinds of feature squeeze techniques are considered
  - x is an attack when $\max\left(d(x, x_{squeeze1}), d(x, x_{squeeze2})\right)$ is large, where $d(x, x_{squeeze}) = \left\|f(x) - f(x_{squeeze})\right\|_1$

$squeeze1$

Color Depth Reduction
Number of bits in each color channel (RBG)



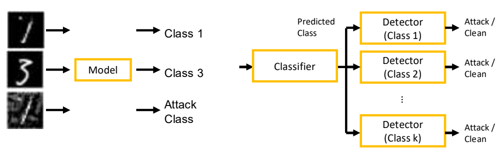
$squeeze2$

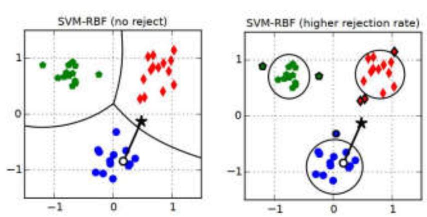
Spatial Smooth

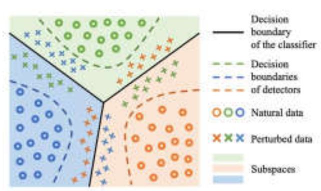
Xu W, Evans D, Qi Y(2017) Feature squeezing: Detecting adversarial examples in deep neural networks. In: NDSS

Implicit Detection: Sample

- Reject uncertain samples
  - Train a detector
    - For whole model
    - For each class
  - Set confident value thresholds



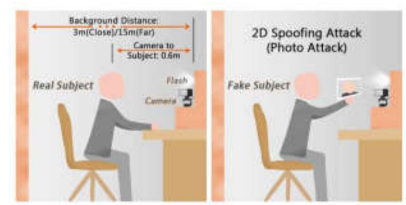


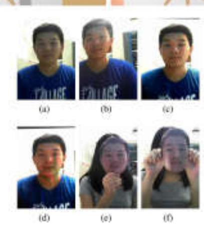


Grosse, Kathrin, et al(2017) On the (statistical) detection of adversarial examples. In: arXiv
Yin, Xuwang, Soheil Kolouri, Gustavo K. Rohde(2019) Adversarial example detection and classification with asymmetrical adversarial training. In: arXiv
Melis M, Demontis A, Biggio B, et al. (2017)Is Deep Learning Safe for Robot Vision? Adversarial Examples Against the iCub Humanoid. In: ICCV Workshops

Implicit Detection: Sample

- Face Liveness Detection
- One camera, no additional hardware
  - No depth information
- Flash is applied to enhance the difference between real and fake persons
- Only available for 2D attack





PPK Chan, W Liu, D Chen(2017) Face liveness detection using a flash against 2D spoofing attack. In: IEEE Transactions on Information Forensics and Security, 13(2), 521-534

Defense: Robust Model

- Build a model aiming to reduce the influence of attack on decisions
  - Adversarial Learning
    - Evasion: Increase attack cost/difficulty
    - Poisoning: Reduce influence of contaminated samples
  - Structure
    - Feature Selection
    - Ensemble

Adversarial Learning

- Basic idea: Consider additional regularization terms in the objective function

$$E = Err + Adv$$

- Two kinds of regularization:
  - Obfuscated Gradient
  - Adversarial Term (Increase Manipulation Cost)

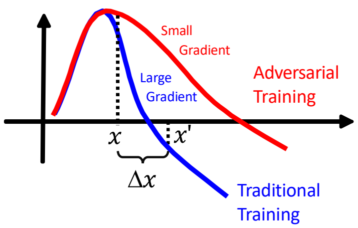
Bai T, Luo J, Zhao J (2023) Recent Advances in Adversarial Training for Adversarial Robustness. In: IJCAI
Goodfellow I J, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: ICLR

Obfuscated Gradient

- Penalizes large gradients

$$\min L(g(x), y) + \lambda \underbrace{||\nabla f(x)||}_{\text{gradient}}$$

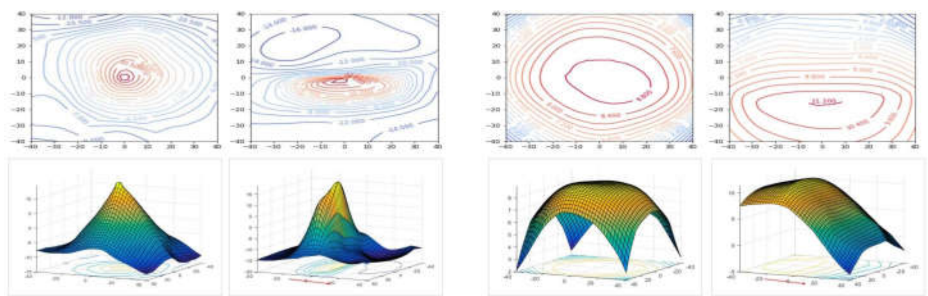
- Mislead crafting in adversarial attack



CJ Simon-Gabriel, Y Ollivier, L Bottou (2018)Adversarial Vulnerability of Neural Networks Increases with Input Dimension. In: ICLR
A Athalye, N Carlini, D Wagner (2018)Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International conference on machine learning (pp. 274-283). PMLR

Obfuscated Gradient

Normal model (Adversarial accuracy: 0.3%) **Defended model (Adversarial accuracy: 44.7%)**

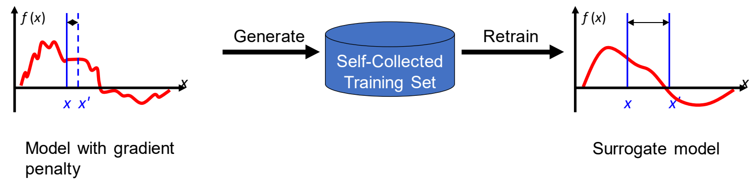


Random Perturbation | Adversary Perturbation | Random Perturbation | Adversary Perturbation

Yu et al., Interpreting and Evaluating NN Robustness, IJCAI 2019

Obfuscated Gradient

- However, obfuscation can be solved easily by training a surrogate model



CJ Simon-Gabriel, Y Ollivier, L Bottou (2018)Adversarial Vulnerability of Neural Networks Increases with Input Dimension. In: ICLR
A Athalye, N Carlini, D Wagner (2018)Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: International conference on machine learning (pp. 274-283). PMLR

Adversarial Term

- A trained model should work well on both clean and attack samples

- Attack samples is generated for each training sample
  - Time complexity of attack sample generation is large

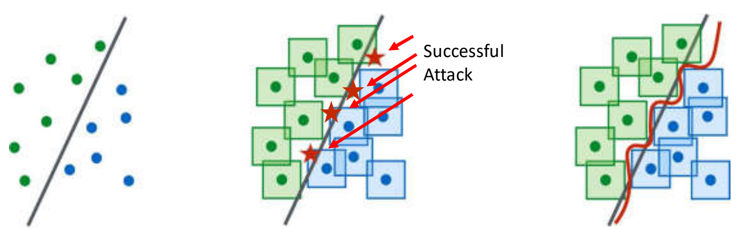
$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha \underbrace{J(\boldsymbol{\theta}, \boldsymbol{x}, y)}_{\text{Error on clean samples}} + (1-\alpha) \underbrace{J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \operatorname{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))}_{\text{Error on attack samples}} .$$

Bai T, Luo J, Zhao J (2023) Recent Advances in Adversarial Training for Adversarial Robustness. In: IJCAI
Goodfellow I J, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: ICLR

Robust Optimization

- Minimize the error and maximize the manipulation

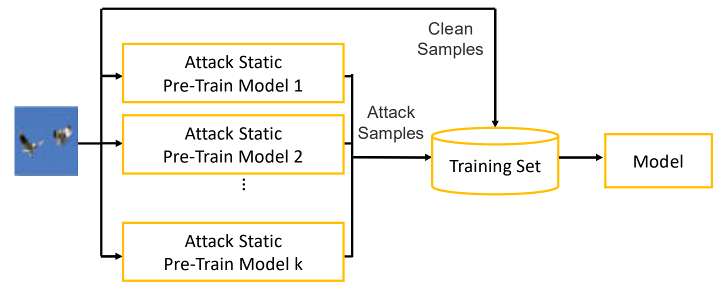
$$\min_{w} \max_{\|\Delta x\|} L_w(x + \Delta x, y)$$



Madry et al., ICLR 2018 (https://arxiv.org/pdf/1706.06083.pdf)
Fast AT (NeurIPS 2020, https://arxiv.org/abs/2007.02617)

Feature squeezing

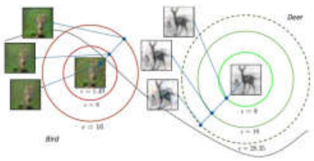
- Increase the diversity of attack samples
- Different views
  - Based on several pre-trained models



Xu W, Evans D, Qi Y(2017) Feature squeezing: Detecting adversarial examples in deep neural networks. In: NDSS

Diversity of Attack Samples

- Increase the diversity of attack samples
- Different attack strengths for samples
  - Avoid pushing an attack sample to another class

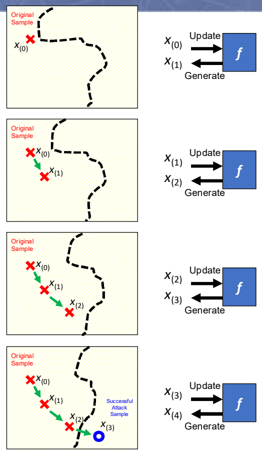


- Different attack strengths for training iterations
  - Adversarial term may dominate at beginning
    - Premature model may be evaded easily
  - Increase PGD iterations during training

Balaji Y, Goldstein T, Hoffman J(2019) Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. In: arXiv
Zhang J, Xu X, Han B(2020) Attacks which do not kill training make adversarial learning stronger. In: PMLR

Cost Reduction

- Crafting adversarial samples is **time consuming**
  - Training set is usually large
- Free Adversarial Training
  - Perturbing a sample and updating the model simultaneously
  - Using incompletely crafted samples to update the model



Shafahi A, Najibi M, Ghiasi A(2019) Adversarial training for free!. In: arXiv

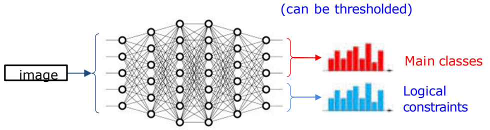
Domain Knowledge

- Improve the model by domain knowledge on the class relationships
  - A natural way to spot incoherent predictions
  - E.g.

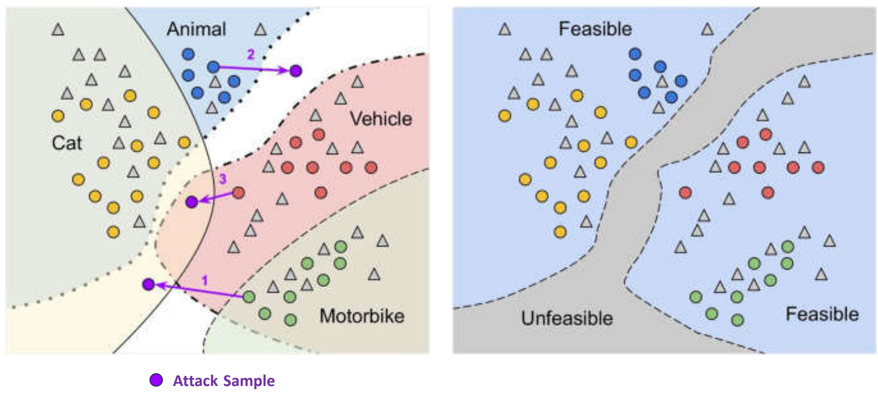
$$\begin{aligned}
&\forall x, && \mathrm{CAT}(x) \Rightarrow \mathrm{ANIMAL}(x)\,,\\
&\forall x, && \mathrm{MOTORBIKE}(x) \Rightarrow \mathrm{VEHICLE}(x)\,,\\
&\forall x, && \mathrm{VEHICLE}(x) \Rightarrow \neg\mathrm{ANIMAL}(x)\,,\\
&\forall x, && \mathrm{CAT}(x) \vee \mathrm{ANIMAL}(x) \vee \mathrm{MOTORBIKE}(x) \vee \mathrm{VEHICLE}(x)
\end{aligned}$$

Penalty (>0) associated to the T-Norm-based constraint from the h[th] formula

$$\min_{\mathbf{f}} = \underbrace{\boxed{\frac{1}{n}\sum_{i=1}^{l} L_y(\mathbf{f}(\mathbf{x}_i), \mathbf{y}_i)}}_{\text{Prediction Loss}} + \underbrace{\boxed{\sum_{j=1}^{l+u}\sum_{h=1}^{m} \lambda_m \cdot L_\phi(\phi_h(\mathbf{f}(\mathbf{x}_j)))}}_{\text{Constraint loss (can be thresholded)}} + \lambda\|\mathbf{f}\|$$

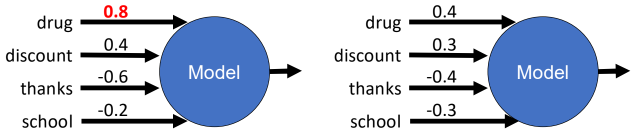


Domain Knowledge



Structure: Feature Selection

- Weight evenness increases robustness
  - Influence of feature on decisions should be even
  - More features should be manipulated to change decisions



- Feature selection may reduce robustness
  - Weight of unselected features is 0
  - Feature weights are less even
  - Evasion may requires less modification

Kolcz A, Teo CH (2009) Feature weighting for improved classifier robustness. In: 6th conference on Email and Anti-Spam (CEAS)

Structure: Feature Selection

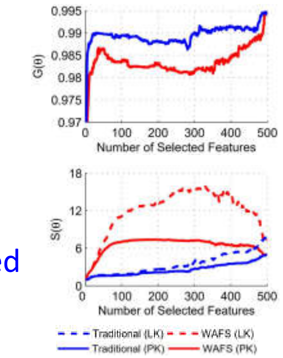
- Adversary-aware Feature Selection
  - Maximize error and robustness

$$\arg\max_{\theta\in\{0,1\}^d} Acc_\theta + \gamma R_\theta$$

$$\text{s.t. } \sum_{j=1}^{d} \theta_j = m$$

where d : feature number, m : cardinality of feature subset, R : robustness estimation, θ : feature set, Acc : accuracy, γ : tradeoff

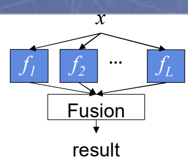
  - Most discriminative features are not selected
  - Variance of discriminative abilities of selected feature decreases, i.e. weights more even

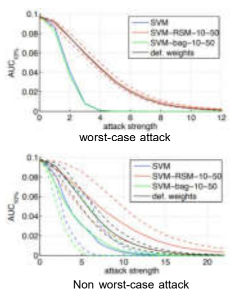


Fei Zhang, Patrick PK Chan, Battista Biggio, Daniel S Yeung, Fabio Roli(2015) Adversarial feature selection against evasion attacks. In: IEEE transactions on cybernetics

Structure: Ensemble

- Weight evenness can be enhanced by ensemble
  - Bagging (Samples)
    Each base classifier is trained with a bootstrap replication of the original training set
  - Random Subspace method (Features)
    Each base classifier is trained with different random feature subsets
  - Simple average is used a fusion





B Biggio, G Fumera, F Roli(2010) Multiple classifier systems for robust classifier design in adversarial environments. In: Journal of Machine Learning and Cybernetics