Lecture 3, July 31, 2019

Robustness of Deep Learning Systems against Deception

Ling Liu

Professor

School of Computer Science

Georgia Institute of Technology

lingliu@cc.gatech.edu

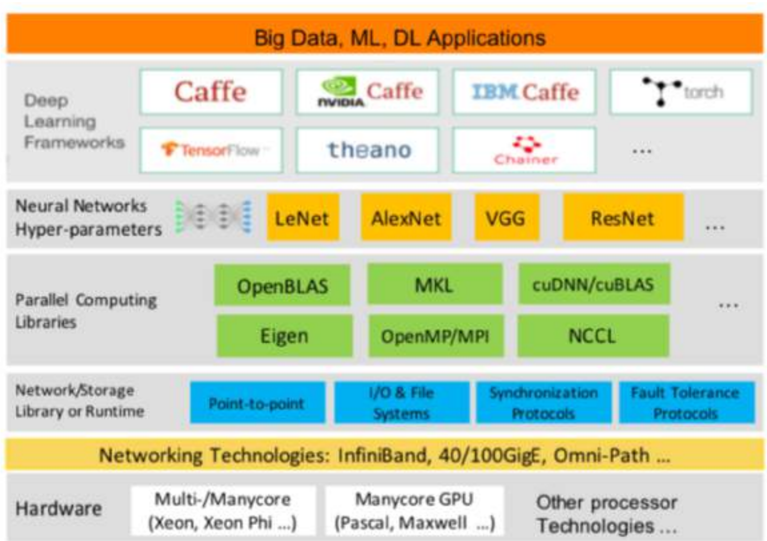
Outline

- **Security of Deep Learning**
  - Adversarial Attacks in Deep Learning:
    - What, How and why
- **Applying deep learning ensemble to tackle security challenges**
  - Attack-Defense Arms Race
  - Strategic Teaming Defense: Our Approach

Deep Learning Software Frameworks

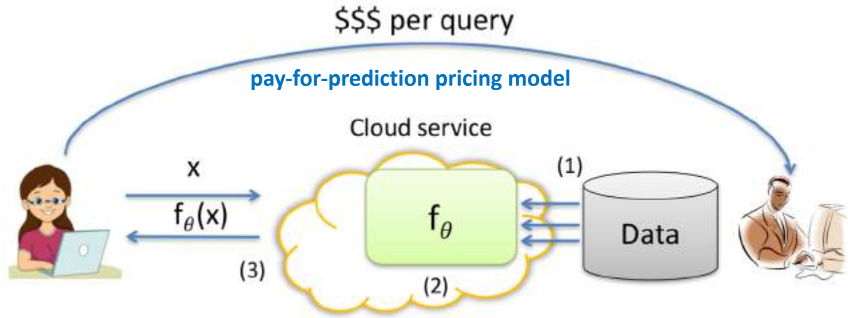
TensorFlow
Caffe
Chainer
Microsoft CNTK
Caffe2
theano
torch
PYTORCH
mxnet
Keras
R
Purine
GLUON
MatConvNet
MINERVA
julia
DL4J
Deeplearning4j
NVIDIA

Deep Learning Software Frameworks



Yanzhao Wu, Ling Liu, Calton Pu, Wenqi Cao, Semih Sahin, Wenqi Wei, Qi Zhang: A Comparative Measurement Study of Deep Learning as a Service Framework, Aug. 2018, A conference version in Proceedings of IEEE 2018 International Conference on Big Data. Seattle, USA, Dec. 10-13, 2018.

Machine Learning as a Service Cloud Platform



(1) Data owner uploads data
(2) Requests training of model f_θ from data
(3) Data owner can make f_θ available for others (its own consumers) to query via a public API while keeping trained model f_θ private

Machine Learning as a Service

Amazon Machine Learning

Google Cloud Platform

Microsoft Azure Machine Learning

bigml

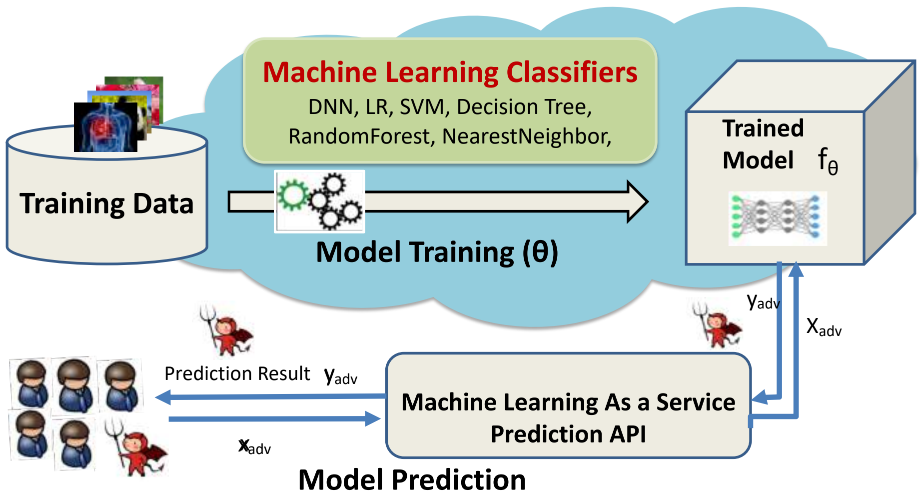
ALGORITHMIA

Service	Model types / training algorithms
Amazon	Logistic regression
Google	Logistic regression, (convolutional / recurrent) neural networks, ...
Microsoft	Logistic regression, decision trees, neural networks, SVM
BigML	Logistic regression, decision trees
Algorithmia	Custom training algorithms (from third party developers)

Object Recognition of Moving Objects

https://www.pyimagesearch.com/2018/02/19/real-time-object-detection-on-the-raspberry-pi-with-the-movidius-ncs/

Adversarial Attacks at Prediction

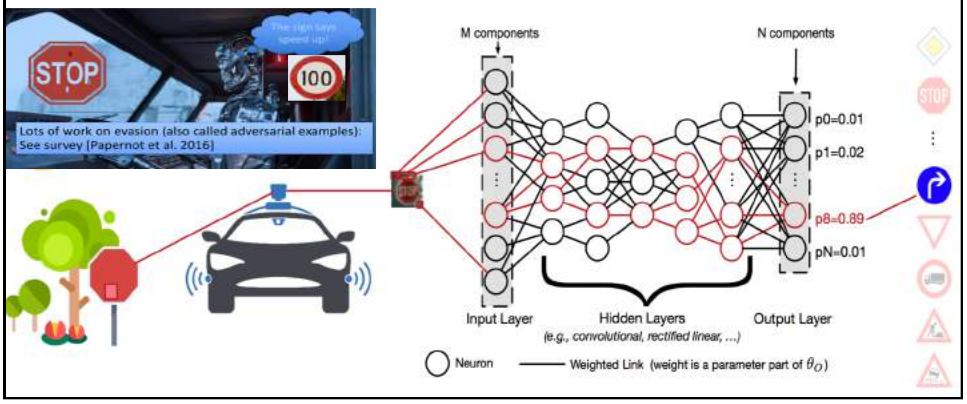


Given: f_θ , x , target prediction y

Find: x_{adv} s.t. x_{adv} and x are "similar", and $f_\theta(x_{adv}) = y_{adv}$

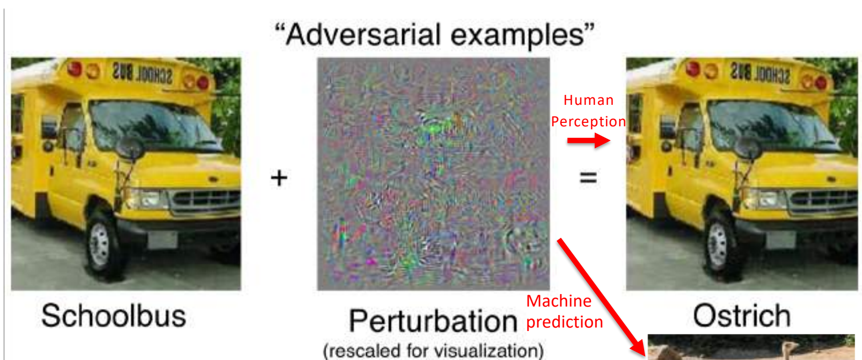
Definition of Adversarial Inputs

- An adversarial example refers to a test input that deliberately modifies the corresponding benign (natural) example to cause the DNN model to produce incorrect classification
  - **Target attacks**: Change the true class prediction to a targeted wrong class prediction
  - **Untargeted attacks**: Change the true class prediction to a different class prediction



Good Model can surprisingly misbehave

Adversarial examples can be formed by using gradient-based optimization to perturb a naturally captured image with small and *imperceptible* changes to increase and maximize the probability of a specific class.



Intriguing properties of neural networks (https://arxiv.org/abs/1312.6199**)** Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus

Adversarial Input Attacks

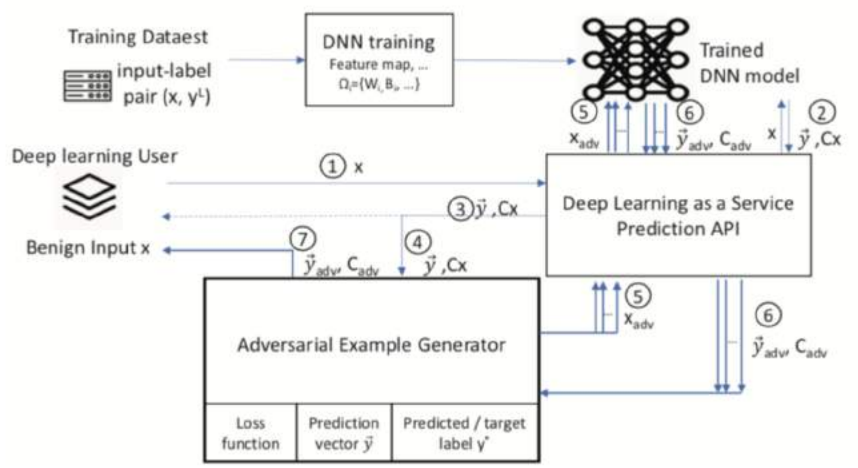


Figure 1: Outsider Adversarial Attack Workflow

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

Adversarial Perturbation: General Formulation

$x_{adv} = x + \Delta x$

Perturbation threshold

Attack objective function

$$\Delta_x = dist(x, x_{adv}) \leq \theta$$

$$s.t. \quad \min \beta\Delta_x + att(x_{adv})(1-\beta)g(\vec{y}_{adv}, y^*)$$

$$x \in X,\ x_{adv} \notin X$$

$$C_{x_{adv}} \neq C_x,\ C_x \in Y,\ C_{x_{adv}} \in Y/C_x$$

$$HC_{x_{adv}} = HC_x$$

Human Imperceptibility

$att(x_{adv})=1$ if attack is targeted and $att(x_{adv})=-1$ if the attack is untargeted

$\boldsymbol{\beta}$: The relative importance of the perturbation and the objective function

X: benign input example — $\vec{y}$: prediction vector of benign input example

X_{adv}: Adversarial example — $\vec{y}_{adv}$: prediction vector of adversarual input example

C_x: Predicted class of x — C_{adv}: Predicted class of x_{adv} — y^T : attack target class

Adversarial Perturbation: Basic Principle

controls how much to perturb at a time

Crafting Rule (location + constraint)

$$x^t_{adv} = \begin{cases} x + \theta R(\frac{\partial h(\vec{y}, y^*)}{\partial x}), \ t = 1, & \text{One step} \\ x_{adv}{}^{t-1} + \theta R(\frac{\partial h(\vec{y}^{\,t-1}_{adv}, y^*)}{\partial x_{adv}{}^{t-1}}), \ t > 1, & \text{Multi step} \end{cases}$$

Untargeted attack: maximize the distance between $\vec{y}_{adv}$ and C_x

Targeted attack: minimize the distance between $\vec{y}_{adv}$ and y^T

Untargeted attack:

$$y^* = C_x$$

Targeted attack:

$$y^* = y^T.$$

One-step attack fast but excessive noise may be added, and it puts more weight on the objective function and less on minimizing the amount of perturbation

Multi-step iterative attack is computationally more expensive, the attack is more strategic with high SR and less perturbation.

Type of Attack Targets

- Targeted attack
  - Make the target model to misclassify by predicting the adversarial example as a intended target class

$$x^*: argmin_{x^*} L(x, x^*) \text{ s.t. } f(x^*) = y^*$$

- Untargeted attack
  - Make the target model to misclassify by predicting the adversarial example as a class other than the original class.

$$x^*: argmin_{x^*} L(x, x^*) \text{ s.t. } f(x^*) \neq y$$

Targeted Attack

- Three Representative Types
  - Most likely target (Most)
    - y^T = arg max $_{y<>Cx}$ $\vec{y}$
  - Least Likely target (LL)
    - y^T = arg min $_{y<>Cx}$ $\vec{y}$
  - Next Class Target (next)

Type of distance measure

- There are three ways to measure the distortion L_0, L_2, L_∞
  - L_0 represents the sum of the number of changed pixels

$$\sum_{i=0}^{n} ||x_i - x_i^*||$$

  - L_2 represents standard Euclidean norm

$$\sum_{i=0}^{n} \sqrt{(x_i - x_i^*)^2}$$

  - L_∞ is the maximum distance value between x and $x*$ (x_{adv})

Example Targeted Adversarial Attacks

Given: f_θ , x , target prediction y

Find: x_{adv} s.t. x_{adv} and x are "similar", and $f_\theta(x_{adv}) = y_{adv}$

Similarity: $x_{adv} \sim x\ (+ \Delta x)$

Adversarial Attack Methods:

Generating Adversarial Examples

attack family	attack in this paper	norm	goal	iteration	magic
FGSM	FGSM	L_∞	untargeted	one	θ
	BIM	L_∞	untargeted	multiple	$\theta,\ I_{max}$
	TFGSM	L_∞	targeted	one	θ
	TBIM	L_∞	targeted	multiple	$\theta,\ I_{max}$
Deepfool	Deepfool	L_2	untargeted	multiple	I_{max}
CW	CW_i	L_∞	targeted	multiple	I_{max}
	CW_2	L_2	targeted	multiple	I_{max}
	CW_0	L_0	targeted	multiple	I_{max}
JSMA	JSMA	L_0	targeted	multiple	I_{max}

norm	definition
L_∞	the maximum change to any pixel of input x.
L_2	the Euclidean distance between input x and x_{adv}.
L_0	total number of pixels of x that are changed.

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

Adversarial Attacks in Model Prediction

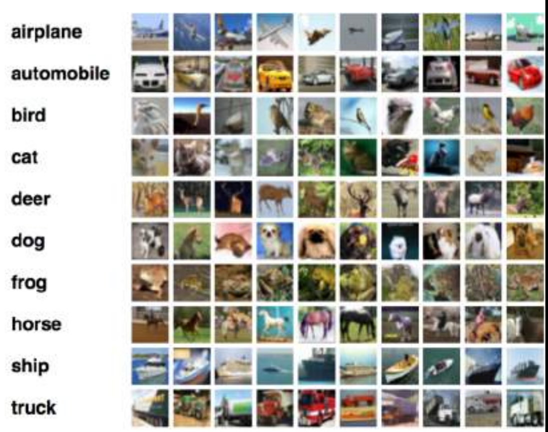
- Fast Gradient Sign Method (FGSM) Attack
  - Untargeted Attack: Source Misclassification
  - Reference
    - Goodfellow et.al. Explaining and Harnessing Adversarial Examples
    - Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus. *Intriguing properties of neural networks*. ICLR 2014
- **J**acobian-based **S**aliency **M**ap **A**pproach (JSMA) Attack
  - Targeted Attack: Source-Target Misclassification
  - Reference
    - Papernot et al. The Limitations of Deep Learning in Adversarial Settings
- Optimization based Attacks
  - Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In Security and Privacy (SP), 2017 IEEE Symposium on. IEEE, 39–57.
  - Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. 2017. Robust physical-world attacks on machine learning models. arXiv preprint arXiv:1707.08945 (2017).

MNIST dataset

http://yann.lecun.com/exdb/mnist/

CIFAR dataset

https://www.cs.toronto.edu/~kriz/cifar.html



Methods of Adversarial Attack

- **Fast-gradient sign method (FGSM)**
  - Take a step in the direction of the gradient of the loss function, simple and good performance.

$$x^* = x + \epsilon \cdot sign(\nabla loss_{F,t}(x))$$

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.
Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017.

Fast Gradient Sign Method (FGSM) Attack

[Goodfellow et.al 2014]

$$x_{adv} = x + \theta sign\left(\frac{\partial J(\vec{y}, y^{C_x})}{\partial x}\right), \quad \text{untargeted}$$

$$x_{adv} = x - \theta sign\left(\frac{\partial J(\vec{y}, y^{T})}{\partial x}\right), \quad \text{targeted}$$

Maximize Attack objective function

$$\theta sign\left(\frac{\partial J(\vec{y}, y^{C_x})}{\partial x}\right).$$

Subject to $|| x_{adv} - x ||_\infty \le \theta$

Minimize amount of perturbation

θ controls the amount of injected noise

$$x_{adv} = x + \Delta x$$
$$\Delta x = dist(x, x_{adv})$$

Crafting Rule

For **untargeted attack**, pixel values should be decreased if $\frac{\partial J(\vec{y}, y^{C_x})}{\partial x} < 0$ and pixel values should be increased if $\frac{\partial J(\vec{y}, y^{C_x})}{\partial x} > 0$, aim at increasing (maximizing) the loss function between $\vec{y}_{adv}$ and C_x. so that the prediction moves away from the source class

For **targeted attack**, the loss function for targeted attack is defined between $\vec{y}$ and the target class of the attack y^T. The direction of change is to decrease (minimize) the loss function so that the prediction moves towards the target class.

Untargeted FGSM on MINST

0.771 ∗ 1032 = 796
digit 2 successful

0.995 ∗ 1135 = 1129
digit 1 successful

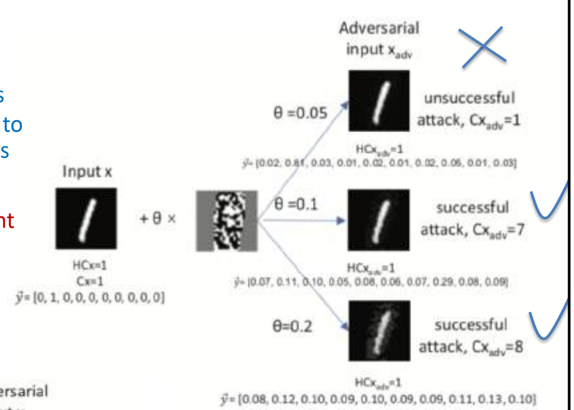
S\D	0	1	2	3	4	5	6	7	8	9	SR	# image
0	0.058	0.018	0.067	0.004	0.015	**0.691**	0.037	0.027	0.082	0.001	0.942	980
1	0.068	0.005	0.123	0.092	0.015	0.078	0.006	0.068	**0.524**	0.021	0.995	1135
2	0.010	0.001	0.229	0.052	0.049	**0.341**	0.018	0.065	0.208	0.027	0.771	1032
3	0.032	0.052	0.192	0.178	0.022	**0.272**	0	0.031	0.093	0.128	0.822	1010
4	0.004	0.017	0.110	0.063	0.048	0.045	0.014	0.049	**0.571**	0.079	0.952	982
5	0.047	0.006	0.142	0.100	0.074	0.135	0.029	0.016	**0.341**	0.110	0.865	892
6	0.056	0.070	0.217	0.013	0.116	0.156	0.041	0.004	**0.307**	0.02	0.959	958
7	0.031	0.030	0.314	0.047	0.018	0.018	0.001	0.117	**0.385**	0.039	0.883	1028
8	0.020	0.033	0.201	0.031	0.094	**0.380**	0.015	0.040	0.158	0.028	0.842	974
9	0.005	0.016	**0.336**	0.013	0.046	0.194	0.003	0.210	0.168	0.009	0.991	1009

Table 2: Untargeted FGSM Attack (θ=0.2): the cell at i^{th} row and j^{th} column represents the fraction of adversarial inputs misclassifies source class in i^{th} row to destination class in j^{th} column.

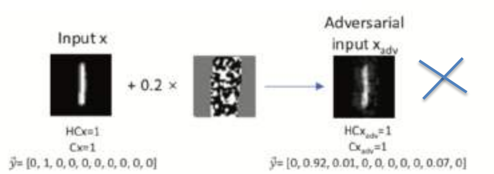
The destination class of untargeted attacks is not uniformly random.

FGSM Attack: Characterization

- Different levels of noise (varying θ) may lead to different attack effectiveness for the same input
- Two different but successful attacks to the same benign input may lead to inconsistent misclassification results
- Different input images of the same class (e.g., digit 1) may have different attack effectiveness even with the same level of noise (same θ)



An illustration of loss function based attack (θ =0.05, 0.1, 0.2), pixel values are added in light area of the perturbation, are deducted in dark area, and do not change in gray area.



An unsuccessful loss function based attack (θ =0.2)

Untargeted Attack with FGSM

- It is not easy for attacker to
  – select the right level of noise θ in one shot
  – find the right amount of perturbation $\epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{x})\right)$

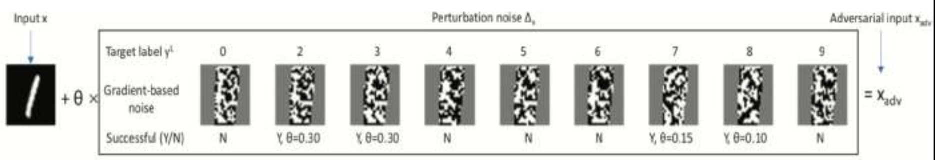
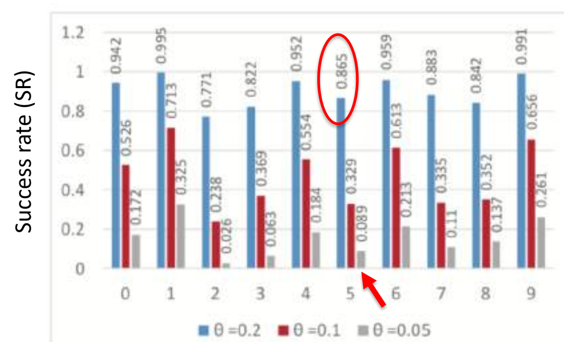


Fig. 4: Visualization of Loss Function-Based Noise Injection for targeted FGSM attack

Characterization of untargeted FGSM
(attacking images of class 0)

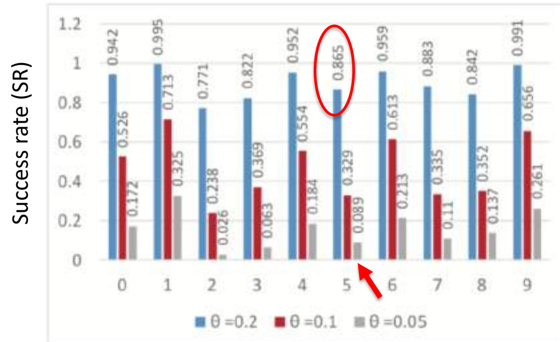


SR of untargeted (one step) FGSM with Different θ: x-axis denotes the 10 classes

iter \ S	0	1	2	3	4	5	6	7	8	9
1	0.172	0.325	0.026	0.063	0.184	0.089	0.213	0.11	0.137	0.261
3	0.796	0.921	0.751	0.789	0.897	0.793	0.903	0.843	0.775	0.960
5	0.988	0.997	0.924	0.959	0.971	0.935	0.982	0.976	0.937	0.998

Table 5: SR of Multi-step FGSM (θ = 0.005).

Characterization of untargeted FGSM



SR of untargeted (one step) FGSM with Different θ: x-axis denotes the 10 classes

iter \ S	0	1	2	3	4	5	6	7	8	9
1	0.172	0.325	0.026	0.063	0.184	0.089	0.213	0.11	0.137	0.261
3	0.796	0.921	0.751	0.789	0.897	0.793	0.903	0.843	0.775	0.960
5	0.988	0.997	0.924	0.959	0.971	0.935	0.982	0.976	0.937	0.998

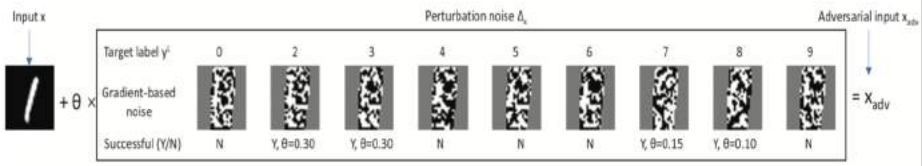
Table 5: SR of Multi-step FGSM ($\theta = 0.005$).

Untargeted FGSM Attack (Random Source Misclassification)

S	Easy 1	Easy 2	Easy 3	Hard 1	Hard 2	Hard 3
0	5/0.691	8/0.082	2/0.067	9/0.001	3/0.004	4/0.015
1	8/0.524	2/0.123	3/0.092	6/0.006	4/0.015	9/0.021
2	5/0.341	8/0.208	3/0.052	1/0.001	0/0.01	6/0.016
3	5/0.272	2/0.192	9/0.128	6/0.0	4/0.025	7/0.031
4	8/0.571	2/0.11	9/0.079	0/0.004	6/0.014	1/0.017
5	8/0.341	2/0.142	3,9/0.11	1/0.006	7/0.016	6/0.029
6	8/0.307	2/0.217	5/0.156	7/0.004	3/0.013	0/0.055
7	8/0.385	2/0.314	3/0.047	6/0.001	4/0.018	5/0.018
8	5/0.38	2/0.201	4/0.094	6/0.015	9/0.028	0/0.033
9	2/0.336	7/0.21	5/0.194	6/0.003	0/0.005	3/0.013

TABLE 4: Top 3 Easy & Hard Attacks under untargeted FGSM: each cell indicates the destination class digit and the fraction of adversarial examples being misclassified into that destination class.

Characterization of targeted FGSM



Visualization of Loss Function-Based Noise Injection for targeted FGSM attack

The pixel position whose value is to be increased when $\frac{\partial J(\vec{y}, y^{C_x})}{\partial x} < 0$ (dark area) and decrease whe $\frac{\partial J(\vec{y}, y^{C_x})}{\partial x} > 0$ (light area)

Takeaway: Boosting small θ iteratively may not improve attack success rate ASR when the attack under large θ is not successful. In addition to tuning θ , the crafting rule may also need to be refined iteratively to boost attack SR.

Attack Methods: Characterization

MNIST dataset

MNIST		attack effect		attack confidence		cost		
attack		ASR	MR	DistACBC	AdvConf	DistPerturb	DistPercept	Time(s)
FGSM	UA	0.46	0.46	0.8673	0.9214	2.436	118.6	0.002
BIM		0.91	0.91	0.9941	0.9959	2.189	88.05	0.009
TFGSM	most	0.61	0.86	0.8633	0.8998	2.470	126.6	0.002
	LL	0.1	0.8	0.7329	0.7636	2.460	128.4	0.002
TBIM	most	0.97	0.97	0.9775	0.9885	2.045	76.27	0.009
	LL	0.64	0.8	0.8850	0.9097	2.114	79.53	0.009
CW_∞	most	1	1	0.9999	0.9999	1.825	61.29	61.73
	LL	1	1	0.9998	0.9998	2.144	86.28	49.95
CW_2	most	1	1	0.9999	0.9999	1.468	23.68	0.432
	LL	1	1	0.9998	0.9999	1.791	37.74	0.378
CW_0	most	1	1	0.9999	0.9999	0.599	17.21	81.99
	LL	1	1	0.9999	0.9999	2.255	34.05	74.55
JSMA	most	0.96	0.96	0.4845	0.7186	1.916	16.67	0.286
	LL	0.49	0.6	0.5175	0.5896	2.346	28.69	0.976

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu.
``Adversarial Examples in Deep Learning: Characterization and Divergence", arXiv, April, 2018.

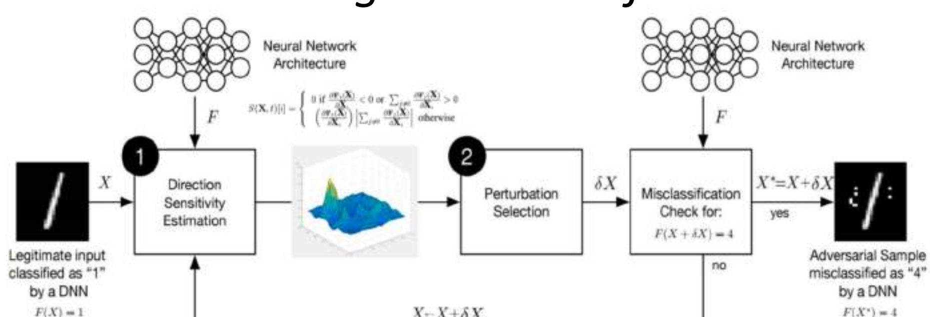
Attack Methods: Characterization

CIFAR-10

CIFAR-10		attack effect		attack confidence		cost		
attack		ASR	MR	DistACBC	AdvConf	DistPerturb	DistPercept	Time(s)
FGSM	UA	0.85	0.85	0.8326	0.8647	0.93	47.63	0.021
BIM	UA	0.92	0.92	0.9484	0.9645	0.607	18.96	0.154
TFGSM	ML	0.82	0.89	0.9090	0.9310	0.93	47.7	0.02
	LL	0.05	0.73	0.5812	0.6331	0.928	47.56	0.019
TBIM	ML	0.94	0.94	0.9547	0.9766	0.604	18.72	0.151
	LL	0.39	0.46	0.7214	0.7923	0.598	18.43	0.155
DF	UA	0.98	0.98	0.5727	0.7388	0.488	7.827	0.283
CW_∞	ML	1	1	0.9820	0.9889	0.571	15.98	235.5
	LL	1	1	0.9721	0.9779	0.726	26.45	243.2
CW_2	ML	1	1	0.9777	0.9867	0.455	6.92	5.772
	LL	1	1	0.9659	0.9732	0.598	13	7.441
CW_0	ML	1	1	0.9838	0.9904	1.251	8.003	355.4
	LL	1	1	0.9695	0.9757	1.587	18.11	356.7
JSMA	ML	1	1	0.2428	0.5366	1.934	27.12	4.894
	LL	0.99	1	0.2206	0.3920	2.338	53.48	9.858

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu.
``Adversarial Examples in Deep Learning: Characterization and Divergence'', arXiv, April, 2018.

Jacobian-Based Iterative Approach: *source-target misclassification*



Crating Rule

w. objective function

$$S(x,T)[\lambda] = \begin{cases} 0, & if \quad \frac{\partial y^T}{\partial x}[\lambda] < 0 \quad or \quad \sum_{j \neq T} \frac{\partial y^j}{\partial x}[\lambda] > 0, \\ \frac{\partial y^T}{\partial x}[\lambda] \left| \sum_{j \neq T} \frac{\partial y^j}{\partial x}[\lambda] \right|, & otherwise, \end{cases}$$

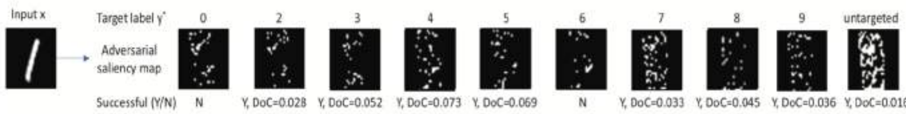
#pixels changed: 15%

$$A = \sum_{i \in \{p,q\}} \frac{\partial y^T}{\lambda_i}, \quad B = \sum_{i \in \{p,q\}} \sum_{j \neq T} \frac{\partial y^j}{\lambda_i},$$

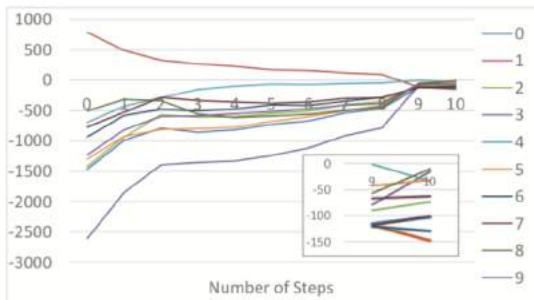
A pixel pair with the largest value on −A × B when A > 0 and B < 0 is chosen to be crafted.
NOTE: A represents to what extent changing these two pixels will change the prediction on the target class.
B denotes the impact of changing the two pixels on classes other than the target.

Papernot et al. The Limitations of Deep Learning in Adversarial Settings. Slide adapted from Papernot@WIFS_T2

Jacobian-Based Iterative Approach: Targeted attack



Visualization of Adversarial Saliency Map-based Noise Injection for targeted attacks. The Adversarial Saliency Map shown is from the rst iteration. The noise of digit 1 is for untargted attack.



Making the Inter-Class Distances smaller each step

Prelogits over 10 steps (source digit 1, target digit 8)

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

Targeted Attack with Jacobian

S \ T	0	1	2	3	4	5	6	7	8	9	S: avg
0		0.027	**0.970**	0.039	0.205	0.147	0.049	0.307	0.352	0.170	0.252
1	0.001		0.856	0.838	0.415	0.502	0.030	0.686	**0.970**	0.510	**0.534**
2	0.001	0.006		**0.285**	0.007	0.003	0.009	0.136	0.237	0.004	0.076
3	0.001	0.027	0.483		0.005	0.136	0.003	0.125	0.114	0.110	0.112
4	0.000	0.188	0.633	0.155		0.145	0.013	**0.768**	0.386	0.173	0.273
5	0.013	0.246	0.077	**0.592**	0.033		0.037	0.217	0.478	0.105	0.120
6	0.040	0.176	**0.815**	0.223	0.618	0.382		0.183	0.630	0.116	0.354
7	0.003	0.034	**0.636**	0.562	0.027	0.129	0.000		0.320	0.208	0.213
8	0.003	0.086	**0.858**	0.575	0.071	0.317	0.016	0.107		0.015	0.228
9	0.010	0.084	0.613	0.761	0.387	0.003	0.000	**0.944**	0.825		0.403
T: avg	0.008	0.097	**0.660**	0.448	0.196	0.196	0.017	0.386	0.479	0.157	

TABLE 7: SR of adversarial examples in Jacobian-based attack.

Target	0	2	3	4	5	6	7	8	9
DoC	0.150	0.050	0.066	0.101	0.102	0.148	0.066	0.029	0.093
Entropy	0.026	0.069	0.068	0.03	0.064	0.017	0.05	0.067	0.048

TABLE 8: DoC and entropy of 1135 images of digit 1.

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

Jacobian-Based Iterative Approach: *source-target misclassification*

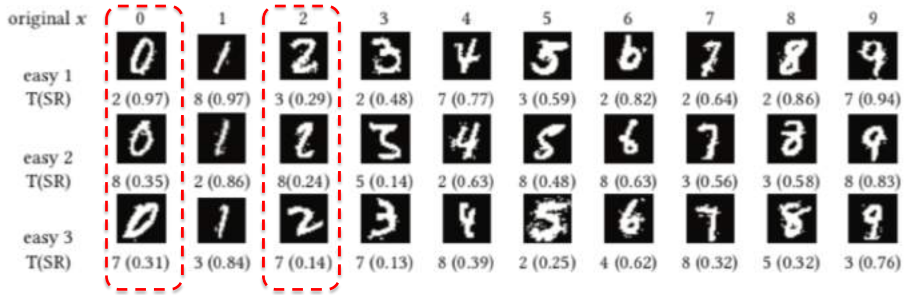
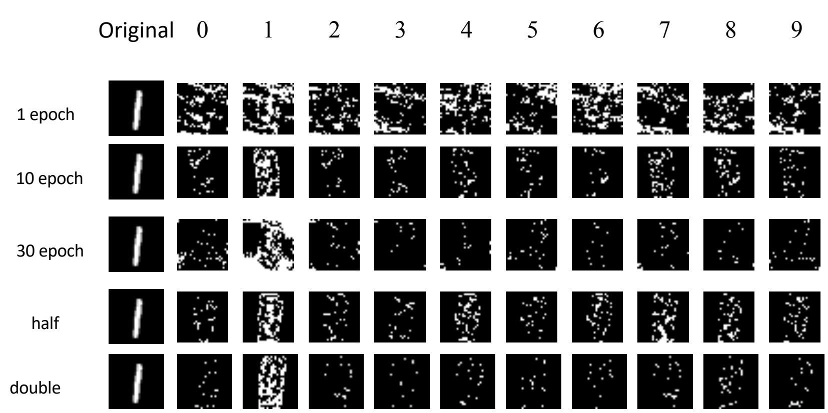


Figure 10: Top 3 easy cases per target in Jacobian-based Attack

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

Attack Effectiveness (Comparison of features)

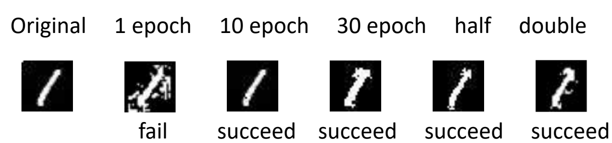
- Networks with different hyperparameters shows different learned features (Saliency map) of one image.



Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

Attack Effectiveness (Comparison of Divergence)

- Given different features captured by different training process, the adversarial attacks would behave differently, making some attack easy and some hard. Here is a demonstration of attacking digit 1 into digit 2.
- For attacks using deep learning models that are trained only 1 epoch, the attack fails to be classified as 2 after crafting 15% of the 28*28=784 pixels.



Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

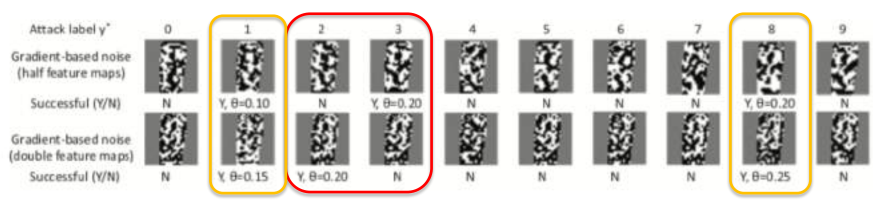
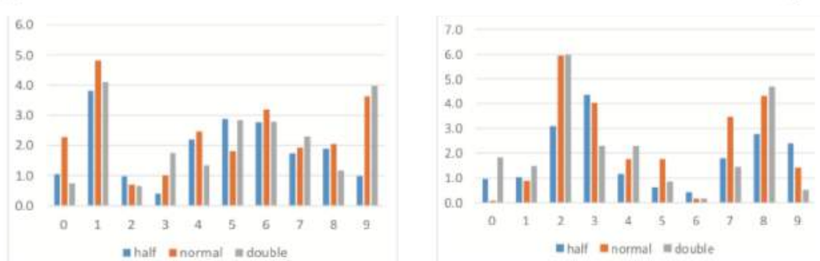


Fig. 14: Loss function-based noise with different feature maps



Vulnerability of Source

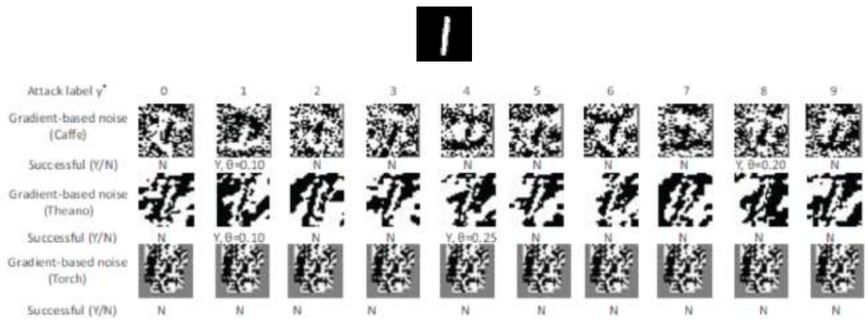
Hardness of Target

Impact of varying sizes of feature maps: higher SR, more vulnerable and lower SR, harder attack.

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence", 2018, https://arxiv.org/abs/1807.00051

Attack Effectiveness (Comparison of features)

Networks trained under different DNN framework show different learned features (gradient of the loss function) of one image.



Liu, Ling, Yanzhao Wu, Wenqi Wei, Wenqi Cao, Semih Sahin, and Qi Zhang. "Benchmarking Deep Learning Frameworks: Design Considerations, Metrics and Beyond." In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018.

Attack Effectiveness (DNN frameworks)

Using untargeted FGSM

- Images in different source classes response differently against the same destination under different DNN frameworks.
- Images in the same source classes response to the attack differently under different DNN frameworks.

Theano

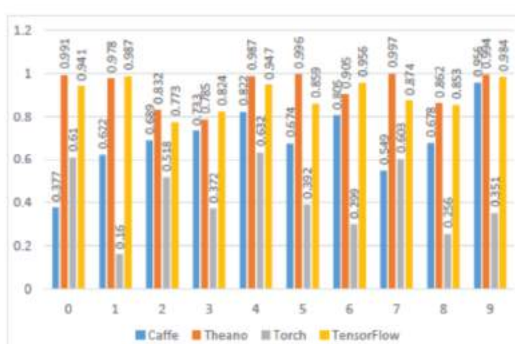
Successful as label 5

Successful as label 3

Caffe

Fail to attack

Successful as label 8

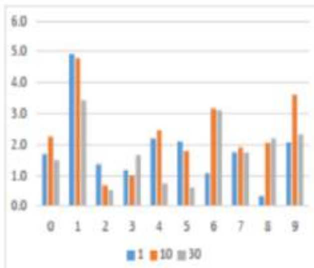


SR of untargeted FGSM with different frameworks

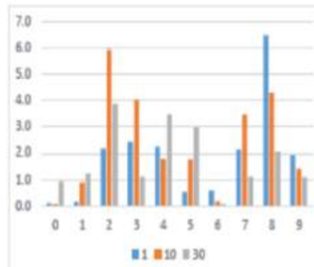
Attack Effectiveness (training epoch)

Using targeted JSMA

- Images in different source classes response differently against same target under different training epochs.
- Images in the same source classes response differently against different target under different training epochs.



Source Vulnerability



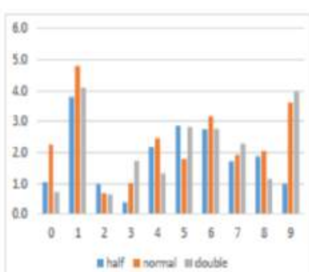
Hardness of Target

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence'', 2018, https://arxiv.org/abs/1807.00051

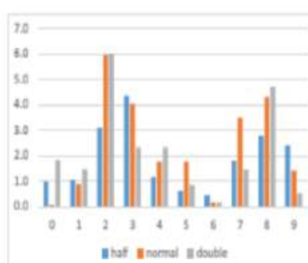
Attack Effectiveness (sizes of feature maps)

Using targeted JSMA

- Images in different source classes response differently against same target under different sizes of feature maps.
- Images in the same source classes response differently against different target under different sizes of feature maps.



Source Vulnerability



Hardness of Target

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence'', 2018, https://arxiv.org/abs/1807.00051

Attack Methods: Characterization

MNIST dataset trained by TF CNN with 99.4% accuracy

CW family of attacks are powerful with 100% attack success rate
Other attacks have high SR in some cases and low SR in other.

MNIST		attack effect		attack confidence		cost		
attack		ASR	MR	DistACBC	AdvConf	DistPerturb	DistPercept	Time(s)
FGSM	UA	0.46	0.46	0.8673	0.9214	2.436	118.6	0.002
BIM	UA	0.91	0.91	0.9941	0.9959	2.189	88.05	0.009
TFGSM	most	0.61	0.86	0.8633	0.8998	2.470	126.6	0.002
	LL	0.1	0.8	0.7329	0.7636	2.460	128.4	0.002
TBIM	most	0.97	0.97	0.9775	0.9885	2.045	76.27	0.009
	LL	0.64	0.8	0.8850	0.9097	2.114	79.53	0.009
CW_∞	most	1	1	0.9999	0.9999	1.825	61.29	61.73
	LL	1	1	0.9998	0.9998	2.144	86.28	49.95
CW_2	most	1	1	0.9999	0.9999	1.468	23.68	0.432
	LL	1	1	0.9998	0.9999	1.791	37.74	0.378
CW_0	most	1	1	0.9999	0.9999	0.599	17.21	81.99
	LL	1	1	0.9999	0.9999	2.255	34.05	74.55
JSMA	most	0.96	0.96	0.4845	0.7186	1.916	16.67	0.286
	LL	0.49	0.6	0.5175	0.5896	2.346	28.69	0.976

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu.
``Adversarial Examples in Deep Learning: Characterization and Divergence", arXiv, April, 2018.

Attack Methods: Characterization

CIFAR-10 trained by DenseNet with 94.5% training accuracy

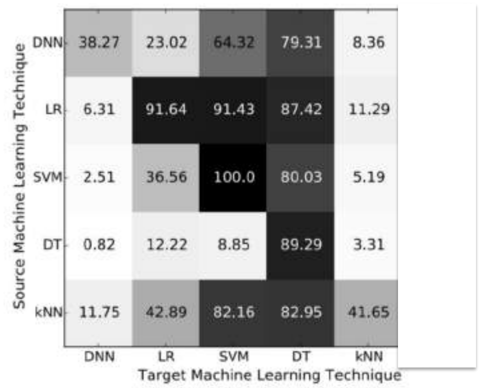
- CW family of attacks, JSMA are powerful with close to 100% attack success rate;
- BIM has 92% ASR for CIFAR-10 (91% for MNIST) but FGSM has 85% for CIFAR-10 (only 46% for MNIST);
- Other attacks still have high SR in some cases and low SR in other.

CIFAR-10		attack effect		attack confidence		cost		
attack		ASR	MR	DistACBC	AdvConf	DistPerturb	DistPercept	Time(s)
FGSM	UA	0.85	0.85	0.8326	0.8647	0.93	47.63	0.021
BIM	UA	0.92	0.92	0.9484	0.9645	0.607	18.96	0.154
TFGSM	ML	0.82	0.89	0.9090	0.9310	0.93	47.7	0.02
	LL	0.05	0.73	0.5812	0.6331	0.928	47.56	0.019
TBIM	ML	0.94	0.94	0.9547	0.9766	0.604	18.72	0.151
	LL	0.39	0.46	0.7214	0.7923	0.598	18.43	0.155
DF	UA	0.98	0.98	0.5727	0.7388	0.488	7.827	0.283
CW_∞	ML	1	1	0.9820	0.9889	0.571	15.98	235.5
	LL	1	1	0.9721	0.9779	0.726	26.45	243.2
CW_2	ML	1	1	0.9777	0.9867	0.455	6.92	5.772
	LL	1	1	0.9659	0.9732	0.598	13	7.441
CW_0	ML	1	1	0.9838	0.9904	1.251	8.003	355.4
	LL	1	1	0.9695	0.9757	1.587	18.11	356.7
JSMA	ML	1	1	0.2428	0.5366	1.934	27.12	4.894
	LL	0.99	1	0.2206	0.3920	2.338	53.48	9.858

Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu.
``Adversarial Examples in Deep Learning: Characterization and Divergence", arXiv, April, 2018.

Cross-model Attack Transferability

- Transferability (black-box attack)
  - An adversarial example modified for a single target model is effective for other model.
  - Using a substitute model that can mimic target model to generate adversarial examples and attack the target model
  - Adversarial examples generated using ensemble-based approaches can successfully attack black box image classification.



Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples (https://arxiv.org/abs/1605.07277), Nicolas Papernot, Patrick McDaniel, Ian Goodfellow

Adversarial Attacks on real-world MLaaS Cloud systems

- Substitute network (black-box attack)
  - The attacker can create a substitute network similar to the target model By repeating the query process.
  - Once a substitute network is created, the attacker can perform a white box attack.
  - Approximately 80% attack success for Amazon and Google services

Remote Platform	ML technique	Number of queries	Adversarial examples misclassified (after querying)
MetaMind	Deep Learning	6,400	84.24%
amazon webservices	Logistic Regression	800	96.19%
Google Cloud Platform	Unknown	2,000	97.72%

All remote classifiers are trained on the MNIST dataset (10 classes, 60,000 training samples)

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017.

Adversarial Examples Beyond Imperceptible perturbation

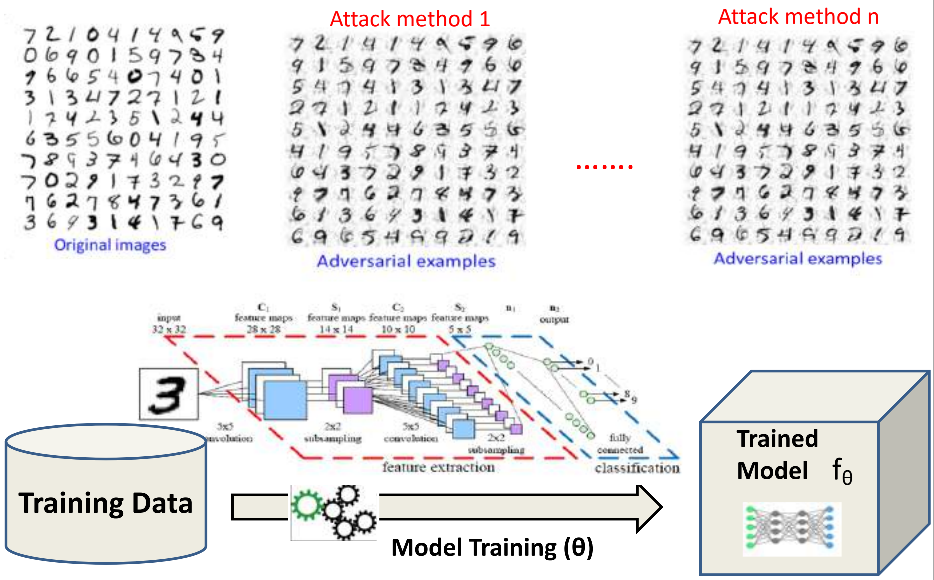
- Type 1: Deviation from human perception
  - The adversarial example generated by applying a small or imperceptible perturbation to a clean image.
- Type 2: Deviation from Training-assumed correct behavior, such as in-distribution data.
  - The adversarial example can be any out-of-distribution examples
  - Such OOD examples will fool a machine learning system due to the limitation of the trained model
- Type 3: Deviation from model output
  - Adversarial example is any type of the input that is *intended* to make the model misclassified
  - Both the above two types and more belong to the input attacks.
  - But the adversarial example attack does not necessarily succeed.
    - "error rate on adversarial examples". If adversarial examples were defined to be actually misclassified, this error rate would always be 1 by definition.

Adversarial Examples that Fool both Computer Vision and Time-Limited Humans (https://arxiv.org/abs/1802.08195)
Gamaleldin F. Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alex Kurakin, Ian Goodfellow, Jascha Sohl-Dickstein

Methods of defense

- The defense of adversarial examples have two types
  - Reactive: detect the adversarial example
  - Proactive: make deep neural networks more robust.
- **Reactive defense**
  - **Adversarial example detection using ensemble**
  - Example: Input transformation
- **Proactive defense**
  - Gradient Masking (e.g., Distillation method)
  - Adversarial training
  - Input Filtering method, including ensemble defense methods.

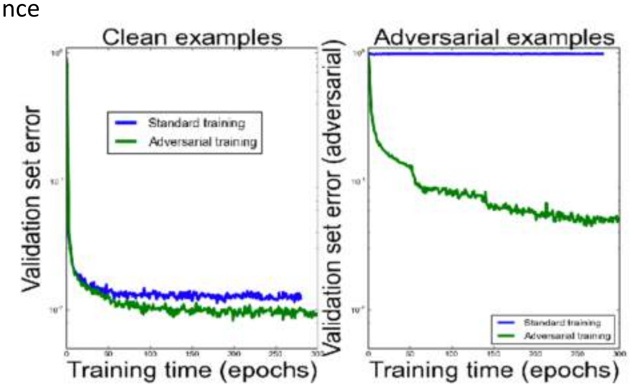
Adversarial Training Defense



Adversarial Example: When is it useful

Adversarial Learning

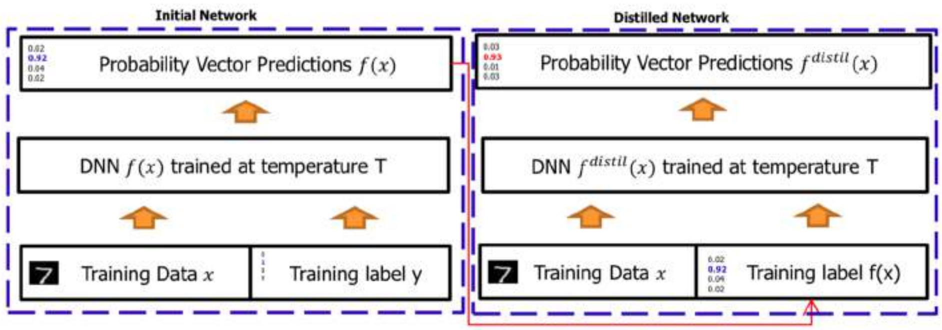
- **Enables secure learning and safe adoption of ML in adversarial settings**
  - Training on Benign examples and adversarial Examples to improve robustness
  - Innovates on finding input data that maximizes model's predicted performance



Graph plots from Goodfellow@OpenAI, 2016

Proactive Defense: Gradient Masking

- Distillation method
  - Using two neural network (detailed class probability)
  - Ex) “1”, class: [0**1**00000000] ⍰ 1”, class:[0.02 **0.91** ... 0.02]



  - Avoid calculating the gradient of the loss function.

Nicolas Papernot, Patrick McDaniel, XiWu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy (SP), 2016*

Defense Methods: Pros & Cons

CLASSIFICATION ACCURACY OF COMPLETELY DEFENSES AGAINST ADVERSARIAL ATTACKS ON CIFAR-10

Datasets	Attack					Original Model	Defense-enhanced Models										Average
	UA/ TA	Objec-tive	Attacks		# of AEs		Adversarial Training			Gradient Mask.		Input Transformation				RC	
							NAT	EAT	PAT	DD	IGR	EIT	RT	PD	TE		
CIFAR-10	UAs	L_∞ $\epsilon=0.1$	FGSM	$\epsilon=0.1$	897	0.0%	76.4%	57.0%	51.0%	7.9%	69.9%	66.7%	7.6%	6.9%	32.6%	0.1%	37.6%
				$\epsilon=0.2$	898	0.0%	52.7%	28.4%	18.5%	10.7%	51.6%	35.0%	4.8%	2.5%	13.3%	0.1%	21.7%
			R+FGSM		837	0.0%	82.7%	80.4%	75.3%	12.2%	76.9%	79.9%	4.2%	4.5%	68.1%	0.0%	48.4%
			BIM		1000	0.0%	84.7%	81.1%	82.4%	3.8%	77.4%	82.3%	0.0%	2.5%	76.5%	0.0%	49.1%
			PGD		1000	0.0%	81.9%	77.8%	74.3%	0.7%	75.1%	79.7%	0.0%	0.2%	64.7%	0.0%	45.4%
			U-MI-FGSM		1000	0.0%	78.7%	65.5%	69.5%	2.7%	73.0%	69.6%	0.0%	0.0%	47.5%	0.0%	40.7%
			UAP		853	0.0%	80.9%	79.8%	60.8%	5.4%	74.4%	71.4%	3.8%	21.3%	47.8%	1.4%	44.7%
		L_2	DF		1000	0.0%	88.9%	86.6%	83.3%	89.3%	79.2%	87.1%	83.2%	74.9%	92.9%	91.3%	85.7%
			OM		1000	0.0%	88.9%	86.1%	82.3%	81.6%	79.0%	87.4%	52.2%	70.5%	91.1%	14.8%	73.4%
	TAs	L_∞ $\epsilon=0.1$	LLC		134	0.0%	79.9%	65.7%	61.2%	1.5%	76.9%	70.2%	3.0%	6.0%	29.9%	0.0%	39.4%
			R+LLC		315	0.0%	84.4%	86.0%	81.3%	6.7%	81.9%	86.0%	4.1%	5.1%	73.3%	0.0%	50.9%
			ILLC		1000	0.0%	86.6%	85.3%	83.7%	27.6%	78.2%	86.9%	0.9%	49.7%	88.5%	0.0%	58.7%
			T-MI-FGSM		1000	0.0%	83.1%	71.4%	70.2%	11.2%	74.5%	78.5%	0.8%	0.0%	61.4%	0.0%	45.1%
		L_0	JSMA		997	0.0%	68.0%	75.1%	72.7%	50.3%	73.5%	70.0%	37.1%	27.1%	75.5%	16.2%	56.6%
		L_2	BLB		1000	0.0%	89.1%	86.4%	83.0%	89.8%	79.2%	87.4%	83.9%	74.1%	92.8%	91.1%	85.7%
			CW2	$\kappa=0$	1000	0.0%	88.8%	86.5%	83.0%	89.5%	79.2%	88.6%	82.9%	76.7%	92.5%	90.2%	85.8%
				$\kappa=20$	1000	0.0%	88.6%	86.3%	82.3%	82.8%	79.2%	88.0%	26.5%	74.4%	92.2%	14.6%	71.5%
			EAD	EN	1000	0.0%	88.5%	86.5%	82.5%	89.2%	79.1%	88.0%	79.3%	74.8%	92.7%	87.5%	84.8%
				L1	1000	0.0%	88.4%	86.6%	82.6%	88.4%	79.0%	86.3%	81.0%	76.2%	92.6%	88.4%	85.0%
	Average				891.1	0.0%	**82.2%**	76.8%	72.6%	39.5%	75.6%	78.4%	29.2%	34.1%	69.8%	26.1%	**58.4%**

Successful adversarial examples are used for attack and the original target model fails with zero defense SR

All 3 types of defenses could not offer over 83% defense success rate (DSR)

X. Ling et.al. DEEPSEC: A Uniform Platform for Security Analysis of Deep Learning Model, IEEE S&P 2019

Problems with existing defense

- Fail to maintain good prediction accuracy on benign inputs
- Fail to generalize over datasets
- Fail to generalize over attack algorithms
- Detection only defenses introduce unwanted disruption
- Fail to generalize over threat models / new attacks

Our approach:
Defense Design Objectives

- Defense should maintain good accuracy on benign inputs
- Defense should minimize adversarial disruption
  – increasing defense success rate (DSR) by maximizing Attack Prevention Success Rate (PSR) instead of detection only (DSR=PSR + TSR)
  – recover and repair as many adversarial input as possible and flag those those that cannot be repaired
- Generalize over attack algorithms
- Generalize over datasets
- Generalize over threat models
  – Certified Defense (guaranteed to be generalizable to unseen attacks)

Defense Methods: Comparison

DataSets	DNN Model	Accuracy
MNIST (60K)	7 layer CNN	0.9943
CIFAR-10 (60k)	DenseNet	0.9484

MNIST				TFGSM		TBIM		DF	CW						JSMA		average
attack	none	FGSM	BIM	ML	LL	ML	LL	DF	ML_∞	LL_∞	ML_2	LL_2	ML_0	LL_0	ML	LL	
Strategic Teaming	**0.9917**	**0.97**	**0.94**	**0.952**	**0.961**	**0.976**	**0.986**	\	**1**	**1**	**0.98**	**0.93**	0.73	0.67	**0.92**	**0.84**	**0.923**
AdvTrain	0.9884	0.91	0.81	0.873	0.86	0.905	0.907	\	0.97	0.88	0.92	0.84	0.67	0.64	0.73	0.69	0.84
DefDistill	0.9784	0.68	0.57	0.417	0.425	0.668	0.752	\	0.91	0.85	0.91	0.84	**0.78**	**0.72**	0.85	0.75	0.74
EnsemInputTrans	0.982	0.6	0.22	0.286	0.309	0.329	0.504	\	0.64	0.51	0.37	0.33	0.21	0.21	0.57	0.64	0.447

CIFAR-10				TFGSM		TBIM		DF	CW						JSMA		average
Attack	none	FGSM	BIM	ML	LL	ML	LL	DF	ML_∞	LL_∞	ML_2	LL_2	ML_0	LL_0	ML	LL	
Strategic Teaming	0.8938	**0.81**	**0.91**	**0.758**	**0.79**	**0.856**	0.9	**0.96**	**0.88**	**0.9**	**0.9**	**0.92**	**0.92**	**0.92**	**0.89**	**0.78**	**0.873**
AdvTrain	0.879	0.64	0.58	0.262	0.442	0.464	0.798	0.75	0.68	0.77	0.75	0.79	0.44	0.48	0.5	0.45	0.586
DefDistill	**0.9118**	0.6	0.65	0.616	0.684	0.77	**0.904**	0.88	0.79	0.88	0.86	0.9	0.6	0.69	0.7	0.47	0.733
EnsemInputTrans	0.8014	0.23	0.4	0.234	0.37	0.406	0.668	0.6	0.56	0.61	0.57	0.61	0.19	0.34	0.45	0.41	0.443

Benign test accuracy

UA

TA (two targets: Most Likely, Least Likely)

Wenqi Wei, Ling Liu: Cross-Layer Strategic Teaming Defense Against Adversarial Examples in Deep Neural Networks, Feb., 2019

Attack Mitigation Strategies

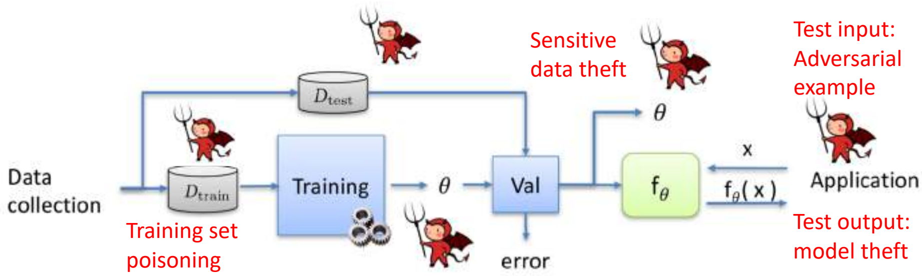
- Attack Mitigation Strategies
  - Adversarial Training
  - Input Transformation
  - Gradient Masking

  Proactive Defense

  - Adversarial Detectors (Reactive, Detection only)
- Mitigation Threat Models
  - **Black Box Attacks**
  - White Box Attacks
  - Grey Box Attacks

  Open Challenges

Adversarial Attacks in Deep Learning



Sensitive training data is protected in an isolated environment (Cloud) and adversary cannot trivially steal sensitive data that are sending over the network.

Poisoning Attack (Causative)

- Know how the learning algorithms work
- Engineering on features or labels of training set
- Change the discriminant function

Evasion Attack (Exploratory)

- Engineering features of prediction input
- Circumvent the legitimate detection
- Change the discriminant result

Types of Adversarial Attacks

- Adversarial examples (input attack)
  – Maliciously perturbed example
  – Out-of-distribution example
  – Transferability of adversarial examples
- Model Theft (output attack)
  – Substitute model mimic the target model, and adversarial crafting against substitute
    - Example: Membership inference
- Training data poisoning
- Training parameter poisoning

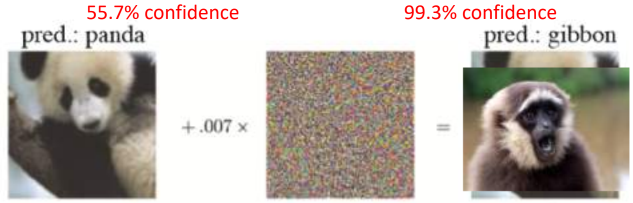
Defense Against the Dark Arts: An overview of adversarial example security research and future research directions, Ian Goodfellow, IEEE Deep Learning and Security Workshop, May 24, 2018

Example Adversarial Attacks

(multi-modality)

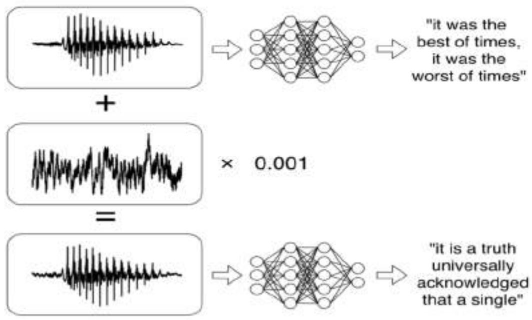
55.7% confidence

99.3% confidence



Credit to Goodfellow (ICLR'15).

Used the same techniques, but with newly-designed loss function.



Credit to Carlini (IEEE S&P 2018).

Attacking Visual Question-Answering Models

- Question: **What color is the traffic light?**
  - Original answer: MCB - **green**, NMN - **green**.
- Attack: Target: **red**.
  - Answer after attack: MCB - **red**, NMN - **red**.

Benign　　Attack MCB　　Attack NMM

Xu_Fooling_Vision_and_CVPR_2018

Machine Learning Model Training Assumption

- I.I.D.
  - I: Independent
  - I: Identically
  - D: Distributed
- All train and test examples drawn independently from the same distribution
- Even when the training data is I.I.D., it does not necessarily mean that the training will capture the same distribution the model will face when it is deployed.
- **Skewed distribution** across classes of the classification task may happen (training set imbalance)
- **Out-of-Distribution Problem**

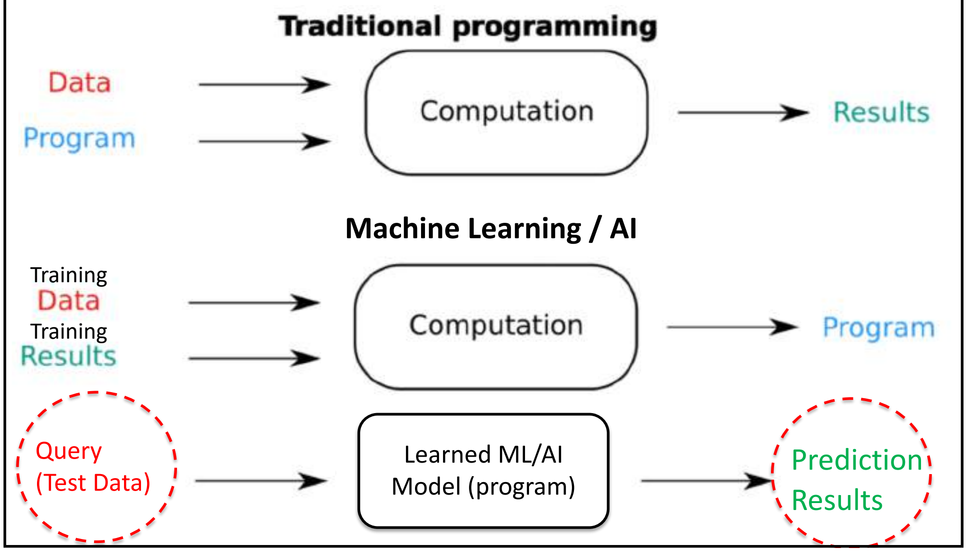
Beyond Security… Privacy

- Membership Privacy against Membership Inference Attacks [StaceyTruex et al TSC 2019]
- Differentially Private Deep Learning [Lei Yu et.al IEEE S&P 2019]

Lei Yu, Ling Liu, Calton Pu, Emre Gursoy, Stacey Truex. ``Differentially Private Model Publishing For Deep Learning'', Proceedings of the 2019 IEEE Symposium on Security and Privacy. pp.309-326. May 20-22, 2019, The Hyatt Regency, San Francisco, CA.
Stacey Truex, Ling Liu, Emre Gursoy, Lei Yu, Wenqi Wei. ``Demystifying Membership Inference Attacks in Machine Learning as a Service'', IEEE Transactions on Services Computing. An earlier version is available at arxiv (May 9, 2018)
Wenqi Wei, Ling Liu, Margaret Loper, Stacey Truex, Lei Yu, Emre Gursoy, Yanzhao Wu. ``Adversarial Examples in Deep Learning: Characterization and Divergence'', May, 2018.

Secure ML/AI Challenges

(Traditional Programming v.s. Machine Learning / AI)



Thank You

64