AI and Cryptography

Lectures 4 & 5 – Adversarial Examples in ML and Differential Privacy for Adversarial Robustness

Luca Mariot

Semantics, Cybersecurity and Services Group, University of Twente

l.mariot@utwente.nl

Trieste, June 28, 2023

Topics and Reading Material

Main topics:

- Basic Recap of Machine Learning (ML)
- Adversarial Examples in ML
- Differential Privacy (DP)
- DP for Adversarial Robustness

References:

- T. Mitchell. Machine Learning. McGraw Hill, 1997
- D. McKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003
- G. James, D. Witten, T. Hastie, R. Tibshirani. An Introduction to Statistical Learning. Springer, 2021
- Papers: see references in the footnotes

This Lecture

- Recap of Machine Learning (ML)
- Adversarial Examples (AE) in ML
- AE from Evolutionary Algorithms
- Defenses from AE
- Differential Privacy (DP)
- DP for Adversarial Robustness

- Algorithms that learn a model to discover something about future data.

Machine Learning

A computer program learns from experience E with respect to some task T and some performance measure P, if its performance on T, as measured with P, improves with experience E.

Types of Machine Learning

Basic components of ML:

- Model.
- Loss function.
- Optimization procedure to minimize the empirical error.

Types of ML:

- Supervised learning.
- Unsupervised learning.
- Semi-supervised learning.
- Reinforcement learning.

Supervised Learning

Training data:

- Training set: pairs (x, y) called training examples.
- x is a *feature* vector, y is a *label*.

Goals:

- The objective is to find a function f such that $y = f(x)$.
- We test our function f on the test set.

Types of Classification:

- If y is a real number $\rightarrow$ regression.
- y is a Boolean variable $\rightarrow$ binary classification.
- y is member of a finite set $\rightarrow$ multiclass classification.

Learning method:

- *Empirical Risk Minimization (ERM)*: the parameters $\boldsymbol{\theta}$ are obtained by solving the optimization problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \mathbf{y}_i). \tag{1}$$

Multilayer Perceptron

- One input layer, one output layer, at least one hidden layer.

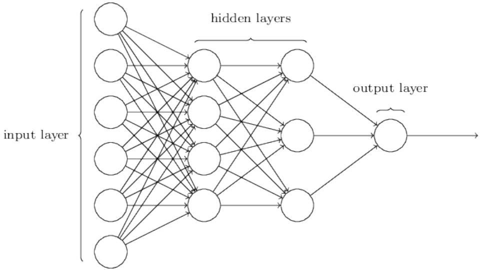
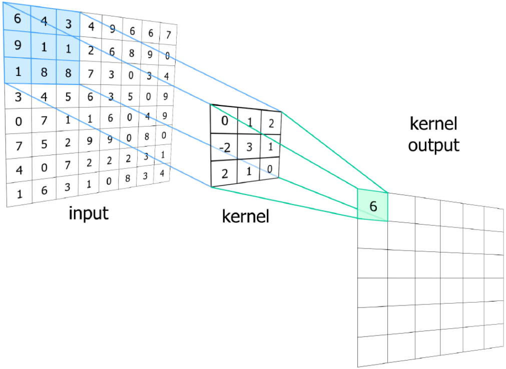


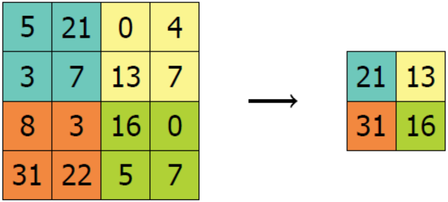
Figure: Multilayer perceptron.

Convolutional Neural Networks - Convolution Layer

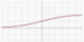
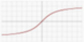
- Convolutional layer: input data are convoluted with some filters, also called *kernels*.



- Pooling layer: The feature map is divided into regions and this layer computes the max (or average) over these regions.



Activation Functions

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

This Lecture

Recap of Machine Learning (ML)

Adversarial Examples (AE) in ML

AE from Evolutionary Algorithms

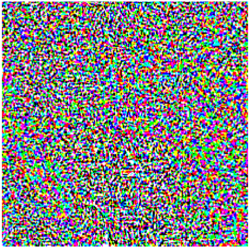
Defenses from AE

Differential Privacy (DP)

DP for Adversarial Robustness

The Problem: Adversarial Examples (AE)

- **Idea**: perturb a valid example to mess the DNN's classification

 +  $\Rightarrow$  [1]

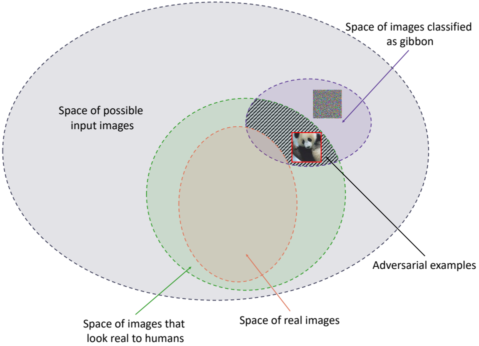
Classification: Panda — Noise perturbation — Classification: Gibbon

- Perturbations move the example beyond the *decision boundary* of a DNN
- Perturbations for AE can be **minimal**

[1] Example credits: I.J. Goodfellow, J. Shlens, C. Szegedy, *Explaining and Harnessing Adversarial Examples*, ICLR 2015

Adversarial Example

- Will the panda image be classified as panda by a neural network?



Why do adversarial examples exist?

- Robust and non-robust features.
- Standard accuracy refers to accuracy on clean examples, robust accuracy refers to accuracy on adversarial examples.

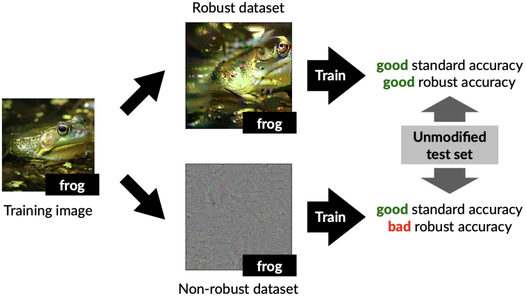
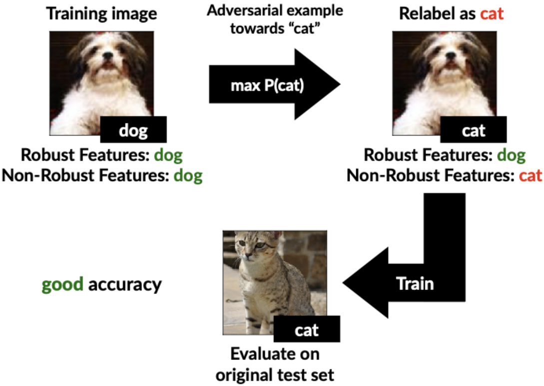


Figure: Ilyas, Andrew, et al. "Adversarial examples are not bugs, they are features." Advances in neural information processing systems 32 (2019).

Why do adversarial examples exist?

- Non-robust feature is enough for standard classification.



Threat Modeling

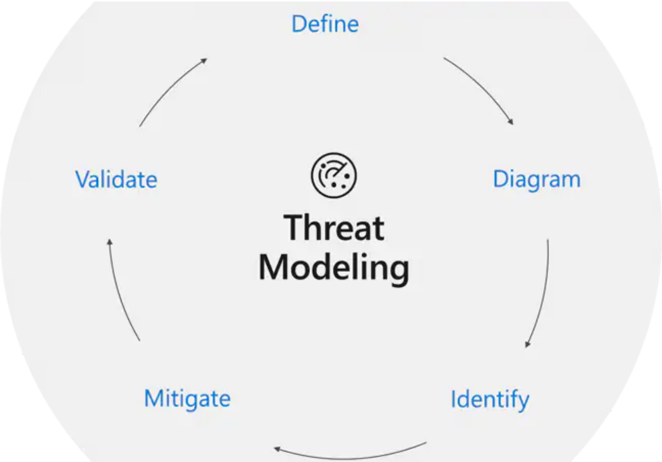
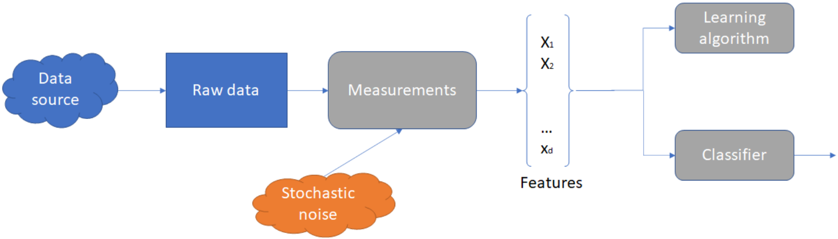


Figure: Threat modeling schematic[2]

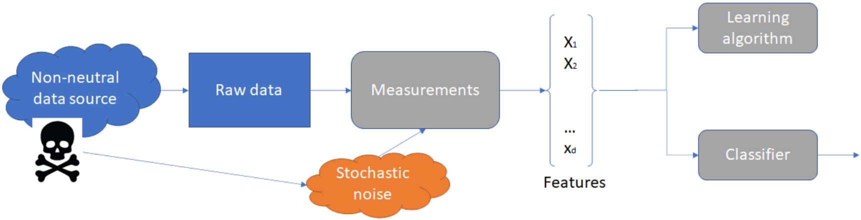
[2] https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling

Statistical Model



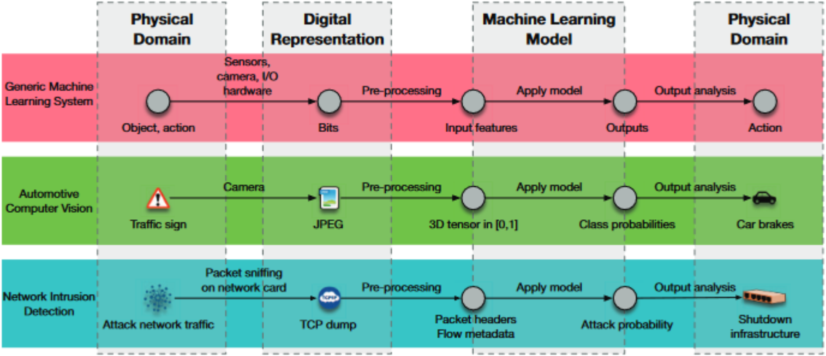
- We assume:
  1. The source of data is given, and it does not depend on the classifier.
  2. Noise affecting data is stochastic.

Adversarial Model



- We observe:
  1. The source of data is not neutral, and it depends on the classifier.
  2. Noise is adversarial and crafted to maximize the probability of error.

Attack Surface



SoK: Security and Privacy in Machine Learning

Adversarial Characterization

Goal:

- *Targeted*: misclassifying to a specific class.
- *Non-targeted*: misclassifying to an arbitrary class.

Knowledge:

- **Components**: Network structure, activation functions, hyperparameters, training data, etc.
- **White-box**: Adversary knows **all**.
- **Black-box**: Adversary knows **none**.

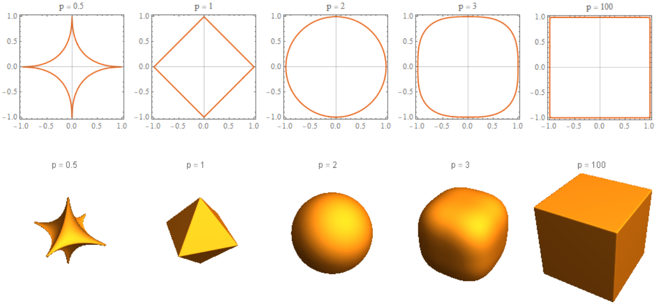
Capability:

- Attacker can modify *test*, not train data
- *One-time* or *iterative* attack

Perturbation Metrics

Perturbation Constraints:

- It should be small and stealthy.
- Measuring via metrics (Minkowsky distance).
- $||x||_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}$.



This Lecture

Recap of Machine Learning (ML)

Adversarial Examples (AE) in ML

AE from Evolutionary Algorithms

Defenses from AE

Differential Privacy (DP)

DP for Adversarial Robustness

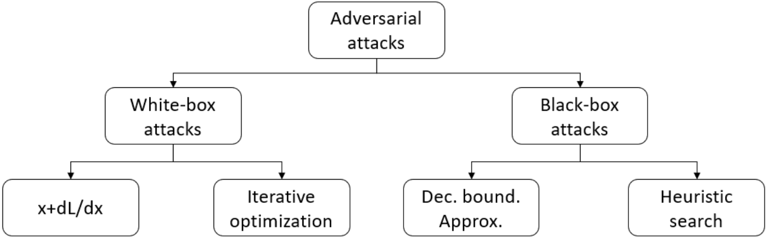
Goal of the Evasion Attack

- Generating an adversarial example x' is optimizing:

$$\min_{x'} \|x' - x\| \quad \text{such that}$$
$$f(x') = \ell',$$
$$f(x) = \ell,$$
$$\ell \neq \ell'.$$

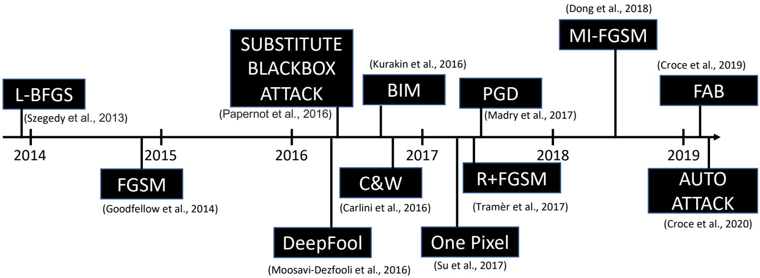
- $\eta = x' - x$ is the perturbation.

Evasion Attacks Classification



Evasion Attacks Timeline

- Since 2013, a large number of attack methods have been proposed.



One-pixel Attack

- **Idea**: Modify just one pixel in a valid example

[3]

- Pixel selection done with **Evolutionary Algorithms**

[3] Image credit: J. Su et al., *One Pixel Attack for Fooling Deep Neural Networks*. IEEE Trans. Evol. Comput 23(5):828-840 (2019)

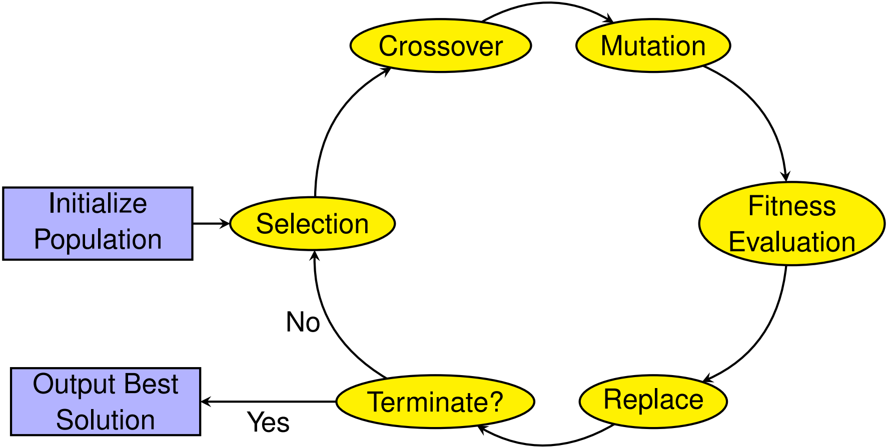
Optimization Problem

- The optimization problem:

$$\begin{aligned} \min_{x'} \quad & \mathcal{J}_\theta(x', \ell'), \\ \text{s.t.} \quad & ||\eta||_0 \leq \epsilon_0 = 1. \end{aligned}$$

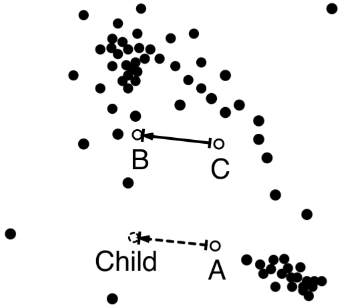
- Updating one pixel according to the gradient is difficult.
- Brute-forcing is not feasible: in CIFAR-10, the search space is $32 \times 32 \times 3 \times 256$.
- Solution: use **Evolutionary Algorithms**

Evolutionary Algorithms (EA)



Differential Evolution

- EA conceived for *continuous* search spaces (e.g., $\mathbb{R}^n$)
- *Adaptive Mutation* (based on the variance of the population)



For each individual *i* do:

- Pick three random vectors a, b, c in the population
- Create $d = a + \alpha(b - c)$
- Create *e* by crossing *i* with *d*
- Each child *e* is compared with the parent *i*

[4]

[4] Image credits: S. Luke. Essentials of Metaheuristics. Lulu, 2012

One-pixel Attack

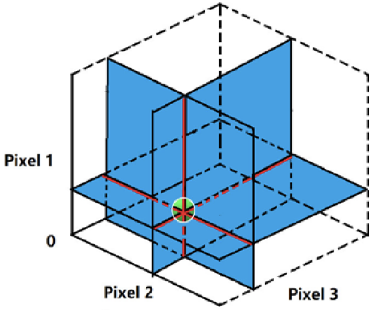


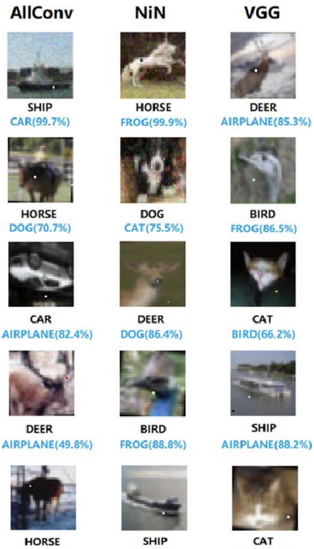
Figure: Illustration of one- and two-pixel search space. One- and two-pixel attacks search the perturbation on, respectively, 1-D (red lines) and 2-D (blue planes) slices of the original 3-D input space.

5

[5] Image credit: J. Su et al., *One Pixel Attack for Fooling Deep Neural Networks*. IEEE Trans. Evol. Comput 23(5):828-840 (2019)

One-pixel Attack

- Examples of one-pixel attack.



This Lecture

Recap of Machine Learning (ML)

Adversarial Examples (AE) in ML

AE from Evolutionary Algorithms

Defenses from AE

Differential Privacy (DP)

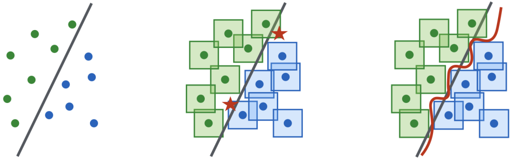
DP for Adversarial Robustness

Why do we want Adversarial robust networs?

- Better accuracy.
- Better explanation of the behavior of networks.

Adversarial Robustness:

- Separating the l_∞-balls requires a significantly more complicated decision boundary.



Type of Defenses

- Adversarial training
- Network Pruning
- Random input transformation
- **Certified Robustness**

Certified Robustness

- Most defenses are *empirical*.
- Certified robustness provides *theoretical guarantees*.

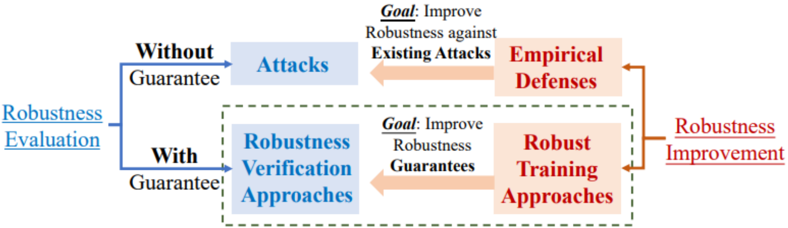


Figure: Empirical vs. certified robustness.[6]

[6] Li Linyi et al. "Sok: Certified robustness for deep neural networks." arXiv preprint arXiv:2009.04131 (2020).

Robustness Verification Taxonomy

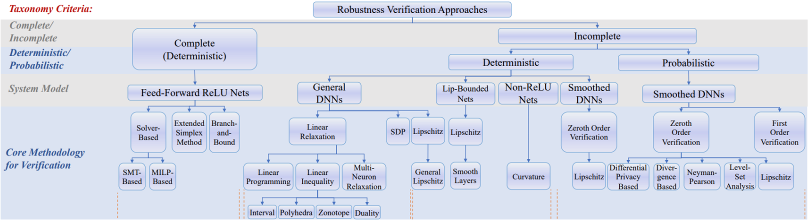


Figure: Robustness Verification Taxonomy.[7]

[7] Li Linyi et al. "Sok: Certified robustness for deep neural networks." arXiv preprint arXiv:2009.04131 (2020).

This Lecture

- Recap of Machine Learning (ML)
- Adversarial Examples (AE) in ML
- AE from Evolutionary Algorithms
- Defenses from AE
- Differential Privacy (DP)
- DP for Adversarial Robustness

Data Anonymization

- Suppose we want to release a dataset with *sensitive information*
- **Classic approach**: perturb the dataset itself
  - *Suppression*
  - *Generalization*

Example: medical dataset

name	age	disease
Alice	30	no
Bob	32	no
Charlie	40	no
Dave	44	yes
Eliza	50	no
Frank	57	yes

- **Query**: youngest age of a person with the disease?
- **Problem**: An adversary might re-identify the (single) row of Dave (age 44, has the disease)

k-anonymity

- **Idea**: partition the row space in groups of size k
- Rows in the same group are indistinguishable wrt an attribute

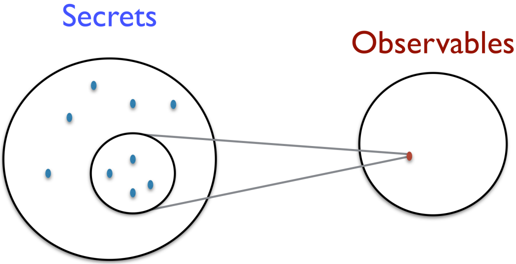
name	age	disease
Alice	30	no
Bob	32	no
Charlie	40	no
Dave	44	yes
Eliza	50	no
Frank	57	yes

name	age	disease
Alice	[30-39]	no
Bob	[30-39]	no
Charlie	[40-49]	no
Dave	[40-49]	yes
Eliza	[50-59]	no
Frank	[50-59]	yes

- Re-identification probability: $p = \frac{1}{k}$

Many-to-one Correlations

- Principle underlying k-anonymity: *many-to-one* correlations
- Problem: *composition attacks*



Composition Attacks

- **Idea**: Combine two or more queries
- **Example**: What is the minimal age AND the minimal weight of a person with the disease?

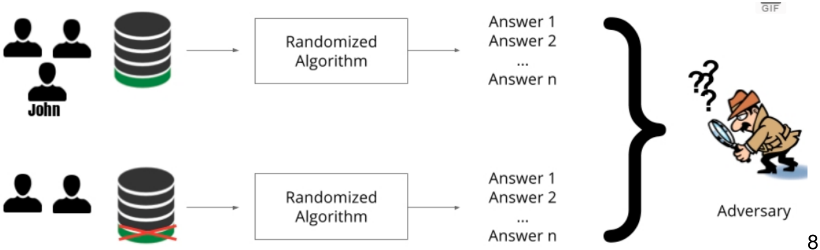
name	age	disease
Alice	[30-39]	no
Bob	[30-39]	no
Charlie	[40-49]	no
Dave	[40-49]	yes
Eliza	[50-59]	no
Frank	[50-59]	yes

name	weight	disease
Alice	[60-79]	no
Bob	[80-99]	no
Charlie	[80-90]	no
Dave	[100-119]	yes
Eliza	[60-79]	no
Frank	[100-119]	yes

- Dave is the only row satisfying the query

Differential Privacy

- **Idea:** anonymize the *query mechanism*, rather than the database itself



[8]

- **Key property**: an adversary has a negligible probability of distinguishing two DBs differing in only *one row*

[8] Image credits: N. Papernot, I. Goodfellow, Privacy and machine learning: two unexpected allies?

Differential Privacy

Ingredients:

- Randomized algorithm A
- Database D
- Output space O

Definition: Differential Privacy

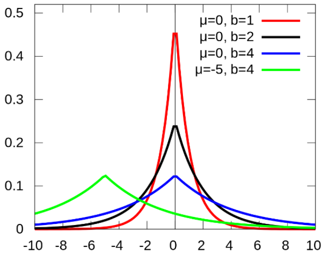
A is (ϵ, δ)-DP wrt a metric ρ on D if for any D' such that $\rho(D, D') \leq 1$ and $S \subseteq O$, it holds:

$$P(A(D) \in S) \leq e^{\epsilon} P(A(D') \in S) + \delta \ .$$

- ϵ, δ: privacy strength parameters (small)
- ρ: usually the *Hamming distance*

Differential Privacy

- How is A implemented?
- Addition of *noise* drawn from specific distribution
- Usual choice: *Laplace noise* $L(\mu, b)$



This Lecture

Recap of Machine Learning (ML)

Adversarial Examples (AE) in ML

AE from Evolutionary Algorithms

Defenses from AE

Differential Privacy (DP)

DP for Adversarial Robustness

DP and Certified Robustness

- **Trick**: input image x is a "DB", where each row is e.g. a pixel
- **Randomized** A: output scores $(y_1(x), \ldots, y_k(x))$ (e.g. given by an activation function like SoftMax)

Theorem (Lecuyér et al. 2019)

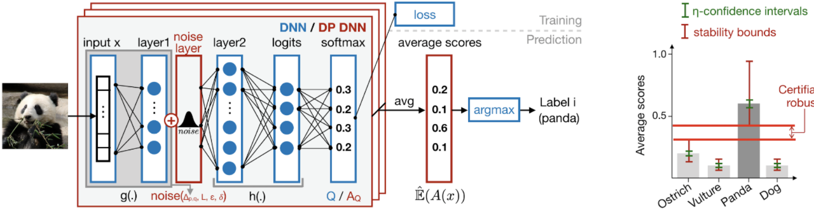
Suppose A is (ϵ, δ)-DP wrt a p-norm metric. If for any input x, and some $k \in K$, we have

$$\mathbb{E}(A_k(x)) > e^{2e} \max_{i:i \neq k} \mathbb{E}(A_i(x)) + (1 + e^{\epsilon})\delta \ ,$$

the classification model is robust to any perturbation α with $|\alpha| < 1$

PixelDP Architecture (Lecuyér et al. 2019)

- Architecture: the noise is added after the *first layer*
- Noise added at inference (test) time



[9]

---

[9] M. Lecuyér et al.: Certified Robustness to Adversarial Examples with Differential Privacy. IEEE S&P 2019

Wrap-up

To summarize:

- Adversarial examples can pose a threat in realistic deployment of DNN
- Several type of countermeasures exist
- *Differential Privacy* provides theoretical guarantees against minimal perturbations

Caveats:

- DP is not a silver bullet!
- Privacy concerns are not addressed in this case