universität innsbruck

Machine Learning in Adversarial Environments

Information Security II

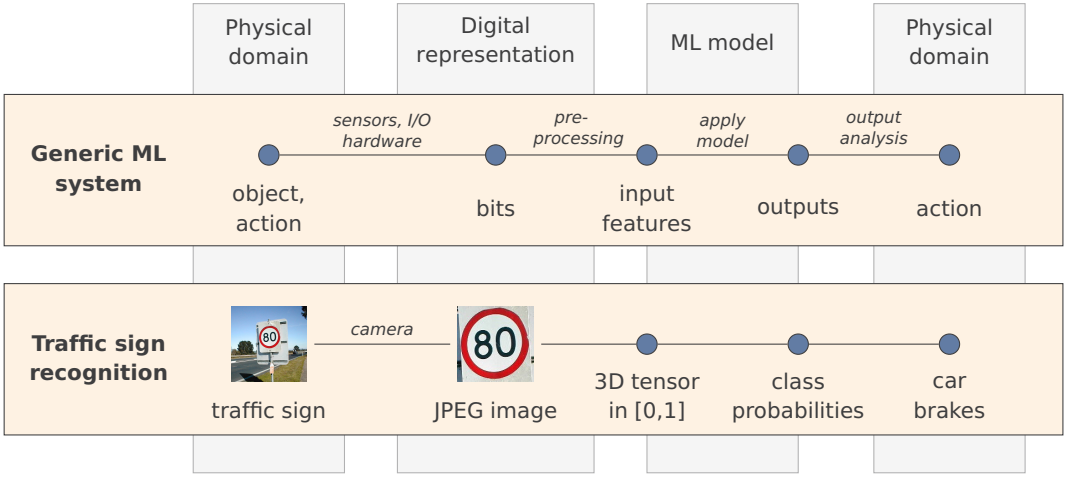
Rainer Böhme

9 March 2021

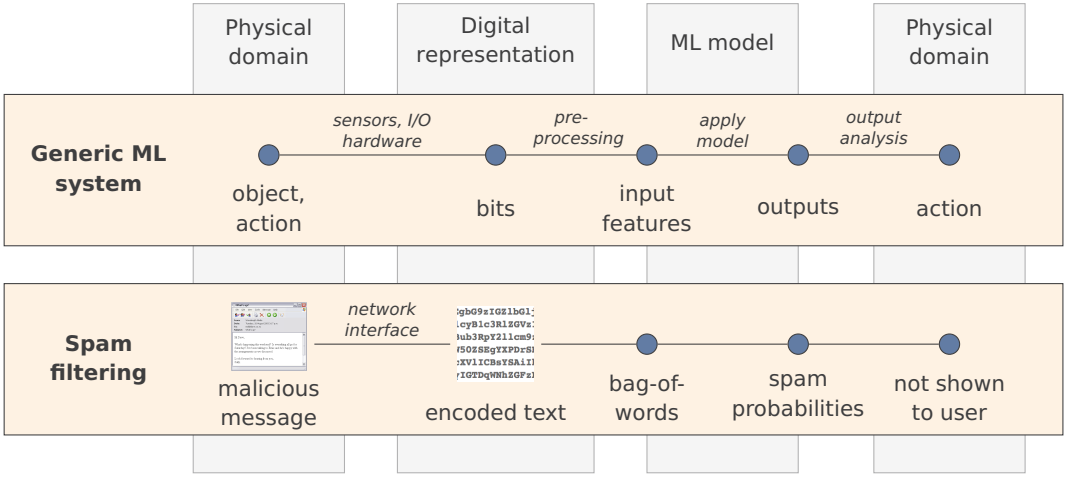
Syllabus – Summer Term 2021

02.03.21	1.	Introduction, prerequisites, a secure single-purpose device
09.03.21	2.	Machine learning in adversarial environments
16.03.21	3.	Multi-purpose systems: confinement & side channels
23.03.21	4.	Multi-purpose systems: access control & vulnerabilities
13.04.21	5.	Hardware-supported security systems
20.04.21	6.	Securing end-to-end network connections
27.04.21	7.	Securing network infrastructures
04.05.21	8.	Availability
11.05.21	9.	Security economics
18.05.21	10.	Privacy policy & theory
01.06.21	11.	Privacy-enhancing technology
08.06.21	12.	Q & A
22.06.21		**Oral exams**

The ML Pipeline for Selected Applications



The ML Pipeline for Selected Applications



How Much Security Can We Expect?

Systems security, idealized

Code

A trusted programmer carefully composes a set of data processing rules, considering all relevant state transitions and the resulting outputs.

Input

An authenticated user controls a trusted channel in order to feed the computer program with the necessary inputs.

Pessimism

Prove worst-case guarantees

Machine learning, caricatured

Data is code

Input–output relations are inferred from data of not fully known sources, stored in incomprehensible data structures, often involving randomization.

Input

Sensors capture data from real-world channels, hopefully approximating the intentions (or at least perceptions) of the user.

Optimism

Compete on average-case metrics

Agenda

1. **Recap of machine learning methods**
2. Threat model
3. Training in adversarial settings
4. Inference in adversarial settings

Forms of Machine Learning

an their relation to statistical methods

1. **Supervised learning**
   Fit a function to known values in order to predict unknown values.
   → regression (for cardinal outputs), classification (for categorial outputs)
2. **Unsupervised learning**
   Find a shorter representation of relevant data.
   → dimensionality reduction: factor/principal component analysis, clustering
3. **Reinforcement learning**
   Supervised learning with iterative updates of a “policy”.
   → (broadly) sequential tests, optimal stopping; online algorithms in CS

Differences to statistics: purely inductive, fewer explicit assumptions, fewer guarantees

Stages in Supervised Learning

Training

Find model parameters θ that minimize the dissimilarity of model outputs $h_\theta(\boldsymbol{x})$ and the corresponding known y.

$$\arg\min_\theta \left| h_\theta(\boldsymbol{x}_i) - y_i \right|_i$$

h_θ	machine learning model
θ	parameters
$\mathcal{H}$	candidate hypotheses $\{\boldsymbol{x} \mapsto h_\theta(\boldsymbol{x}) \mid \theta \in \Theta\}$
$\lvert \cdot \rvert_i$	loss function on all training points $i = 1, \ldots$
$(\boldsymbol{x}, y)$	training point: (input, output)

Inference

Fix θ. Predict property of interest for new input $\boldsymbol{x}$ by computing $h_\theta(\boldsymbol{x})$.

$$\hat{y} = h_\theta(\boldsymbol{x})$$

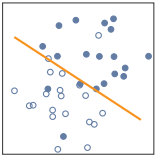
$\hat{y}$	prediction

Types of Classifiers

Fisher linear discriminant analysis (FLD)

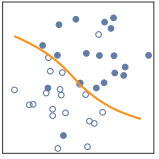
Find hyperplane which maximizes the signal-to-noise ratio of class labeling:

$$\max \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{within}}} \quad \Leftrightarrow \quad \begin{cases} \text{point on the plane} & \boldsymbol{p} = (\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)/2 \\ \text{normal vector} & \boldsymbol{n} \propto (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)^{-1} \end{cases}$$



Kernelized support vector machine (SVM)

Find hyperplane which maximizes the margin to the closest data points, possibly by discounting data on the “wrong side” with a hinge loss function. The dual problem admits non-linear kernels, leading to hyperplanes in higher dimensions. This translates to non-linear classification in feature space.



Deep neural network (DNN)

Fit ensembles (wide) of non-linear discriminant functions (perceptrons), organized in multiple layers (deep), using error back-propagation.
Self-regularization properties admit more features than training points. As a result, the raw “data are the features”. Projections to 2D are highly non-linear.

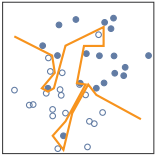


Illustration of Modern ANN Architectures

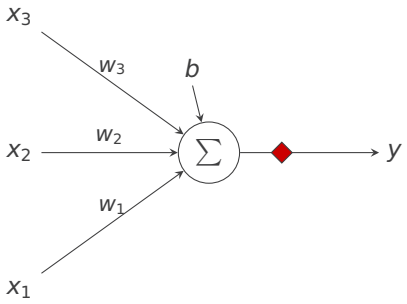


Illustration of Modern ANN Architectures

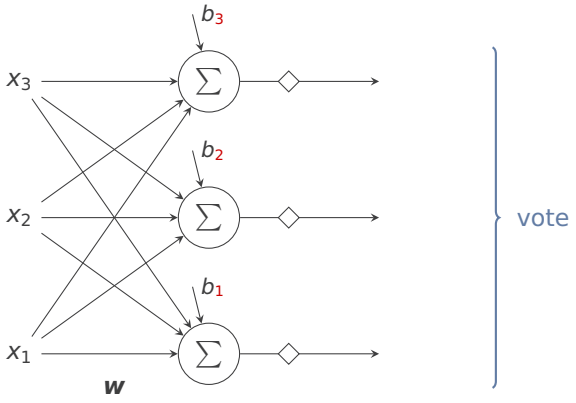
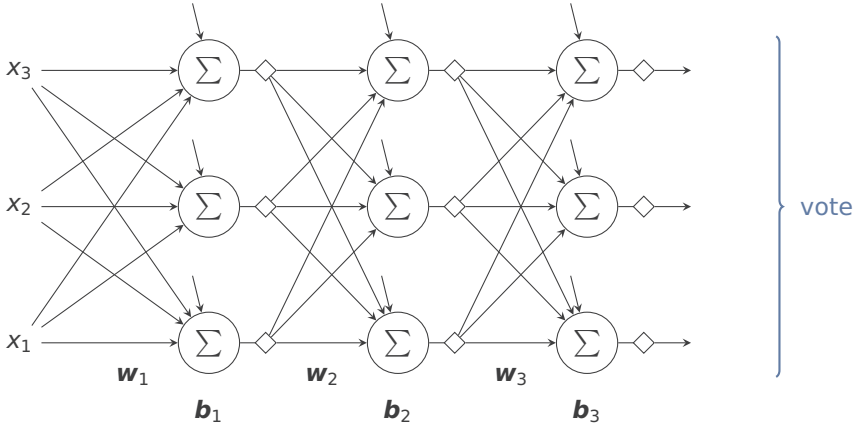


Illustration of Modern ANN Architectures



ResNet/ResNeXt-101 (Xie et al. 2016) have $\sim$44 million weights and require $\sim$300 MB storage space.

Agenda

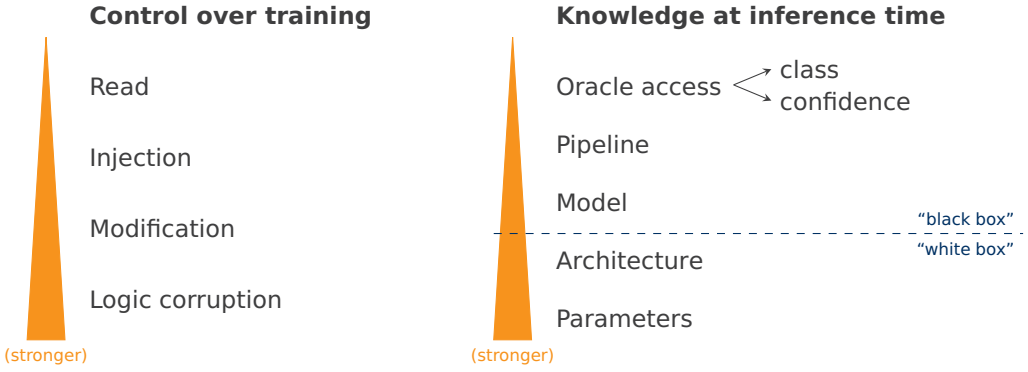
1. Recap of machine learning methods
2. **Threat model**
3. Training in adversarial settings
4. Inference in adversarial settings

Derived Protection Goals

Confidentiality	of training data	→ model inversion attack (5)
	of learned model	→ model stealing attack (4)
	of past queries	
	of past predictions	
Integrity	of predictions	→ evasion attack (3)
	of learned model	→ backdoor attack (2)
Availability	of useful predictions	→ poisoning attack (1)
		→ risks to fairness (6)

(excluding goals that relate to conventional access control)

Adversarial Capabilities

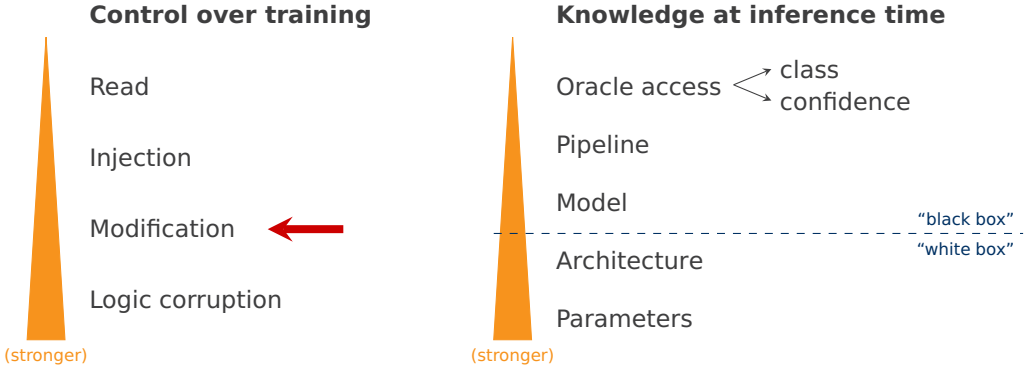


Adapted from Papernot et al. 2018, Fig. 3

Agenda

1. Recap of machine learning methods
2. Threat model
3. **Training in adversarial settings**
4. Inference in adversarial settings

Adversarial Capabilities



Adapted from Papernot et al. 2018, Fig. 3

Stages in Supervised Learning

Training

Find model parameters θ that minimize the dissimilarity of model outputs $h_\theta(\boldsymbol{x})$ and the corresponding known y.

$$\arg\min_\theta \left| h_\theta(\boldsymbol{x}_i) - y_i \right|_i$$

Inference

Fix θ. Predict property of interest for new input $\boldsymbol{x}$ by computing $h_\theta(\boldsymbol{x})$.

$$\hat{y} = h_\theta(\boldsymbol{x})$$

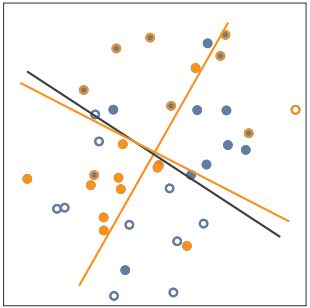
h_θ machine learning model
θ parameters
$\mathcal{H}$ candidate hypotheses $\{\boldsymbol{x} \mapsto h_\theta(\boldsymbol{x}) \mid \theta \in \Theta\}$
$|-|_i$ loss function on all training points $i = 1, \ldots$
$(\boldsymbol{x}, y)$ training point: (input, output)

write access: poisoning

$\hat{y}$ prediction

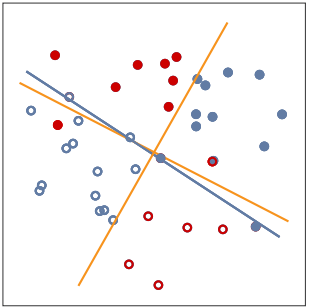
Poisoning Attack

Training



Label flipping

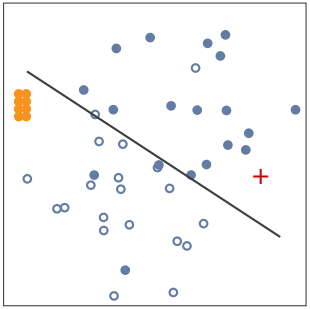
Inference



Error rate: 5 % → 12.5 %

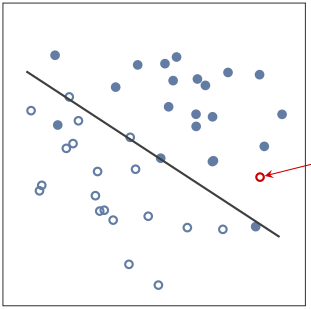
Poisoning Attack

Training



Injection

Inference

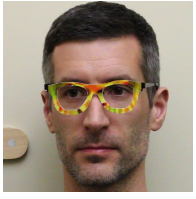


Examples of Backdoor Poisoning Attacks

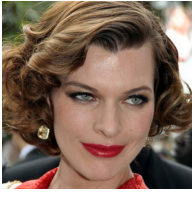
Digital

misclassification →

Physical



misclassification →



Top left: Reese Witherspoon (by Eva Rinaldi, cited from Sharif et al. 2016, CC BY-SA, cropped from https://goo.gl/a2sCdc)
Top right: Russell Crowe (by Eva Rinaldi, cited from Sharif et al. 2016, CC BY-SA, cropped from https://goo.gl/AO7QYu)

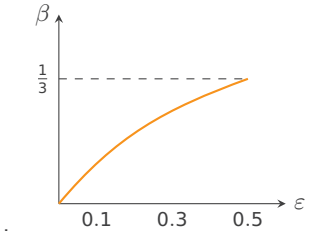
Bottom left: Milla Jovovich (by Georges Biard, cited from Sharif et al. 2016, CC BY-SA, cropped from https://goo.gl/GlsWIC)

Efficiency of Poisoning

How much data?

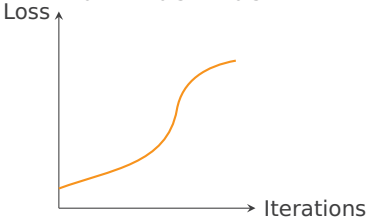
β fraction of poisoned training samples
ε target error rate

$$\rightarrow \beta \leq \frac{\varepsilon}{1+\varepsilon}$$



Where to inject a single point?

Goal: $\arg\max |\cdot|_j \rightarrow$ Loss at inference

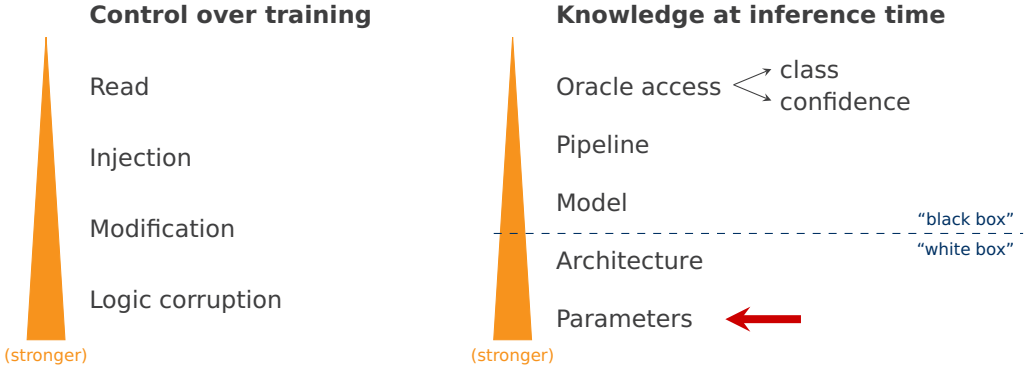


M. Kearns et al. (1993): Learning in the presence of malicious errors, *SIAM Journal on Computing*.

Agenda

1. Recap of machine learning methods
2. Threat model
3. Training in adversarial settings
4. **Inference in adversarial settings**

Adversarial Capabilities



Adapted from Papernot et al. 2018, Fig. 3

Stages in Supervised Learning

Training

Find model parameters θ that minimize the dissimilarity of model outputs $h_\theta(\boldsymbol{x})$ and the corresponding known y.

$$\arg\min_\theta \left| h_\theta(\boldsymbol{x}_i) - y_i \right|_i$$

h_θ	machine learning model
θ	parameters
$\mathcal{H}$	candidate hypotheses $\{\boldsymbol{x} \mapsto h_\theta(\boldsymbol{x}) \mid \theta \in \Theta\}$
$\lvert \cdot \rvert_i$	loss function on all training points $i = 1, \ldots$
$(\boldsymbol{x}, y)$	training point: (input, output)

Inference

Fix θ. Predict property of interest for new input $\boldsymbol{x}$ by computing $h_\theta(\boldsymbol{x})$.

write access: evasion

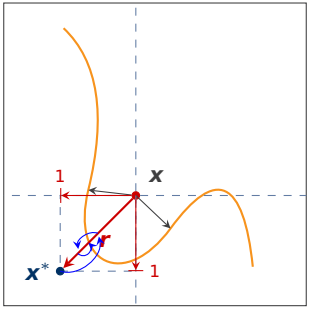
$$y^* = h_\theta(\boldsymbol{x} + \boldsymbol{r})$$

$$\text{s.t.} \quad \boldsymbol{x}^* = \boldsymbol{x} + \boldsymbol{r} \in D, \quad h_\theta(\boldsymbol{x} + \boldsymbol{r}) \not\equiv h_\theta(\boldsymbol{x})$$

$\hat{y}$	prediction
$\equiv$	two outputs of h_θ map to the same class

Evasion Attack

Fast gradient sign method (FGSM)



Attacker's goal: output = o

One-shot:

$$\boldsymbol{r} \in \{-1, +1\}^k$$
$$\boldsymbol{x}^* = \boldsymbol{x} + \boldsymbol{r}$$

Iterative variant:

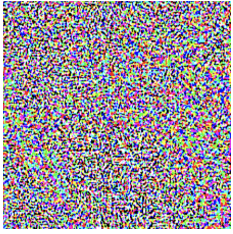
$$\boldsymbol{x}^* = \boldsymbol{x} + \alpha\, \boldsymbol{r}\,,$$

where scaling α is found with binary search.

Goodfellow, Shlens & Szegedy 2015

Example of the Evasion Attack

$+ \alpha \cdot$



$=$

“panda”

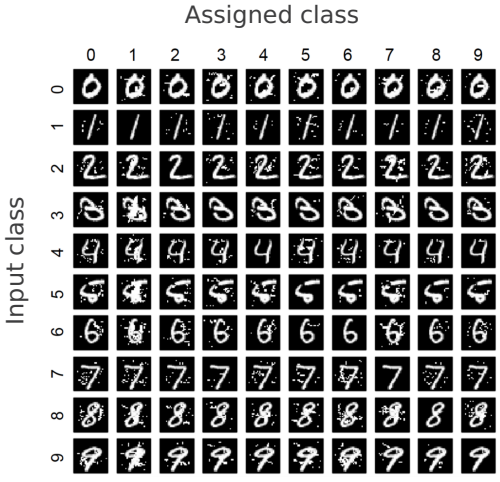
confidence: 57.7 %

“gibbon”

confidence: 99.3 %

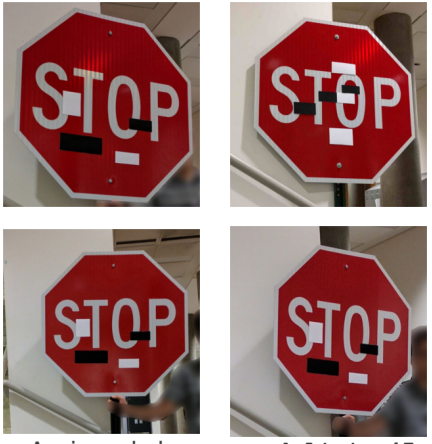
I. Goodfellow, J. Shlens, C. Szegedy (2015): Explaining and harnessing adversarial examples, *ICLR* (Poster).

Examples of Targeted Evasion Attacks



N. Papernot et al. (2018): The Limitations of Deep Learning in Adversarial Settings, *IEEE Euro S&P*.

Adversarial Examples in the Physical Domain



Assigned class: `speed-limit-45`

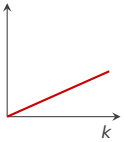
K. Eyckolt et al. (2018): Robust Physical-World Attacks on Deep Learning Visual Classification, *IEEE CVPR*.

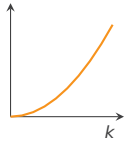
Efficiency of Evasion

Attacker's effort	white-box	black-box
One-shot	$O(\|\boldsymbol{w}\|)$	$O(\|\boldsymbol{w}\| \cdot k)$
for $\|\boldsymbol{w}\| \approx k$:	$O(k)$	$O(k^2)$

back propagation

forward prediction





Iterative

$$O(|\boldsymbol{w}|) + O\left(\log\left(\tfrac{d+1}{d}\right) \cdot |\boldsymbol{w}|\right)$$

precision

binary search

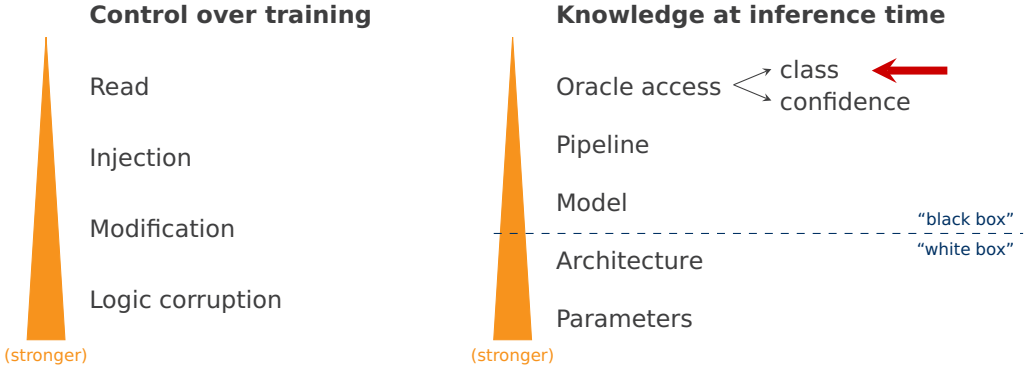
$$O(|\boldsymbol{w}| \cdot k) + O\left(\log\left(\tfrac{d+1}{d}\right) \cdot |\boldsymbol{w}|\right)$$

cost of higher precision

$\|\boldsymbol{w}\|$	size of the classifier (e. g., # of perceptrons in an ANN)
d	distance to decision boundary
k	dimension of the data

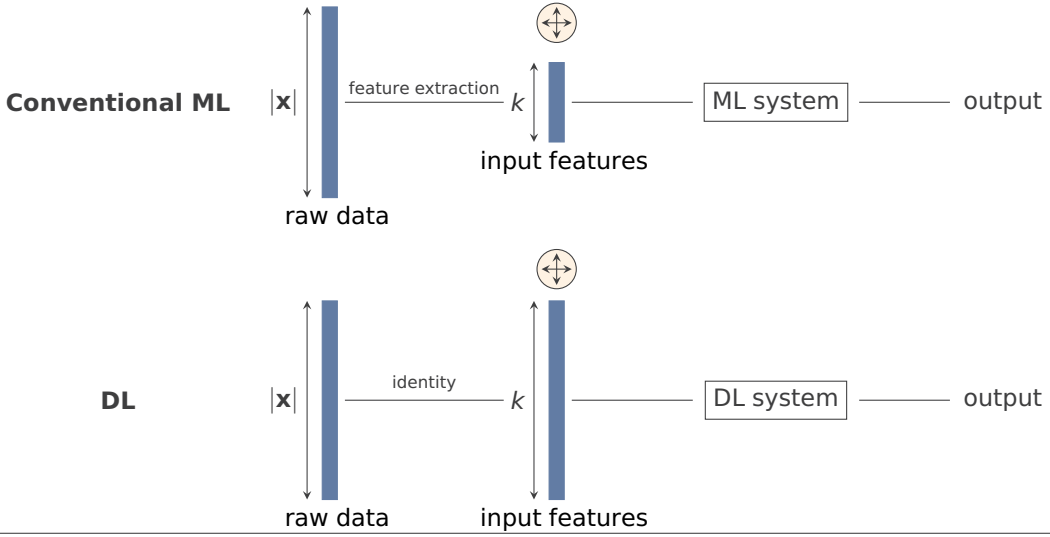
Memo item: all attacks on this slide are fast and approximate. The optimal attack is almost always NP-hard in k.

Adversarial Capabilities



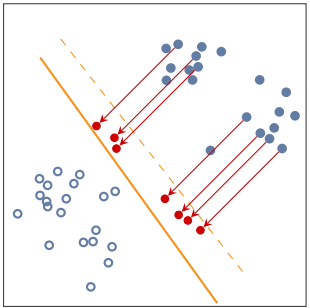
Adapted from Papernot et al. 2018, Fig. 3

Specific Weakness of Deep Learning



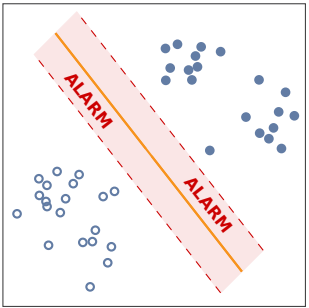
Defenses Against Evasion Attacks

Adversarial training



Attacker's response: adjust scaling

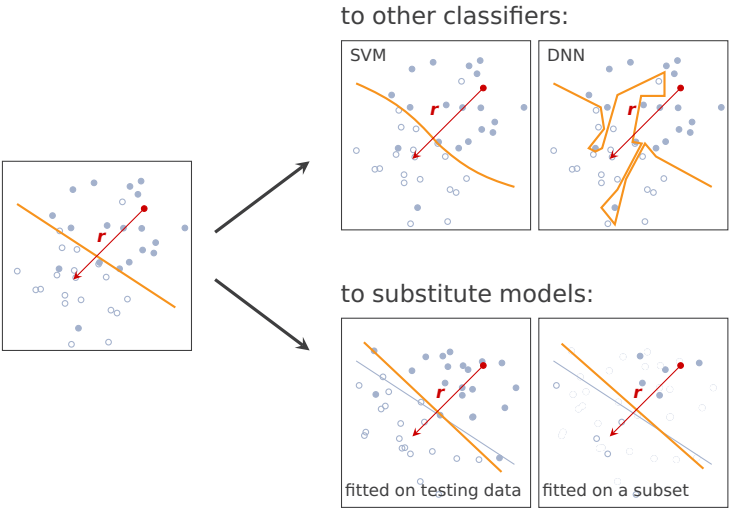
Detection
at inference time



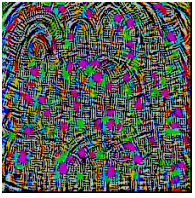
Attacker's response: add alarm as constraint

Curated list of white-box defenses against evasion attacks: https://www.robust-ml.org/defenses/

Transferability



Example for Transferability

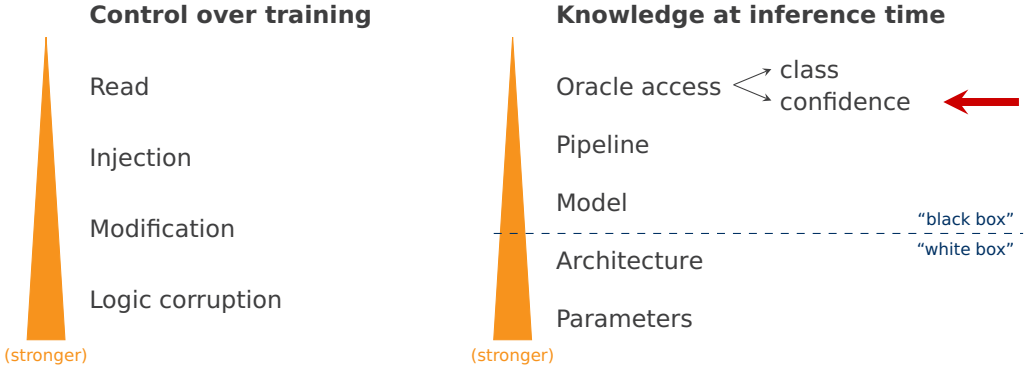


Adversarial perturbation against CaffeNet

Target model	**CaffeNet**	**VGG-F**	**GoogLeNet**
Attack success rate	93.3%	74.0%	47.7%

Moosavi-Dezfooli et al. (2017): Universal adversarial perturbations, *IEEE CVPR*.

Adversarial Capabilities



Adapted from Papernot et al. 2018, Fig. 3

Stages in Supervised Learning

Training

Find model parameters θ that minimize the dissimilarity of model outputs $h_\theta(\boldsymbol{x})$ and the corresponding known y.

$$\arg\min_\theta \left| h_\theta(\boldsymbol{x}_i) - y_i \right|_i$$

Inference

Fix θ. Predict property of interest for new input $\boldsymbol{x}$ by computing $h_\theta(\boldsymbol{x})$.

$$\hat{y} = h_\theta(\boldsymbol{x})$$

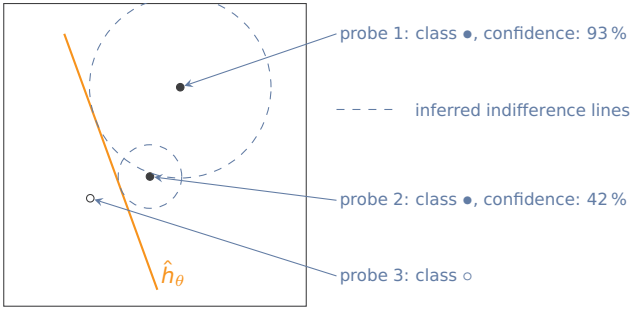
read access: model stealing

- h_θ machine learning model
- θ parameters
- $\mathcal{H}$ candidate hypotheses $\{\boldsymbol{x} \mapsto h_\theta(\boldsymbol{x}) \mid \theta \in \Theta\}$
- $|\cdot|_i$ loss function on all training points $i = 1, \ldots$
- $(\boldsymbol{x}, y)$ training point: (input, output)

- $\hat{y}$ prediction

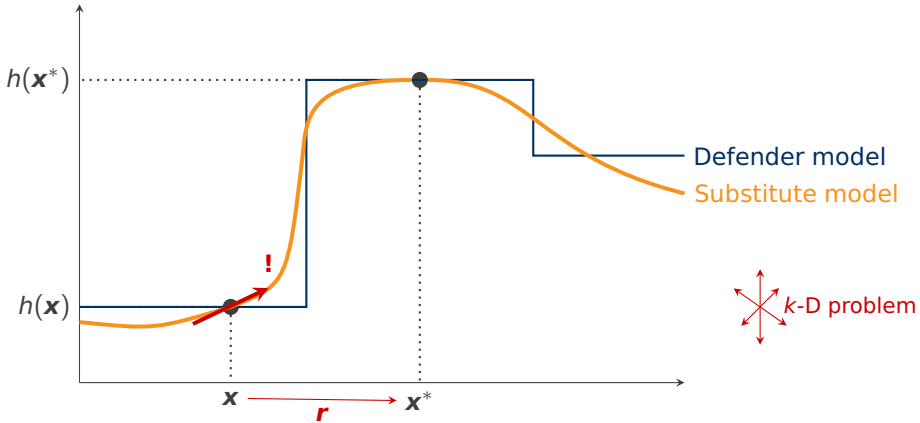
Model Stealing

Example for $k = 2$ dimensions, inference time



General approach: query $O(k)$ linear independent probes and solve equation system

Gradient Masking



Cf. Papernot et al. 2018, Fig. 5

Stages in Supervised Learning

Training

Find model parameters θ that minimize the dissimilarity of model outputs $h_\theta(\boldsymbol{x})$ and the corresponding known y.

$$\arg\min_\theta \left| h_\theta(\boldsymbol{x}_i) - y_i \right|_i$$

Inference

Fix θ. Predict property of interest for new input $\boldsymbol{x}$ by computing $h_\theta(\boldsymbol{x})$.

$$\hat{y} = h_\theta(\boldsymbol{x})$$

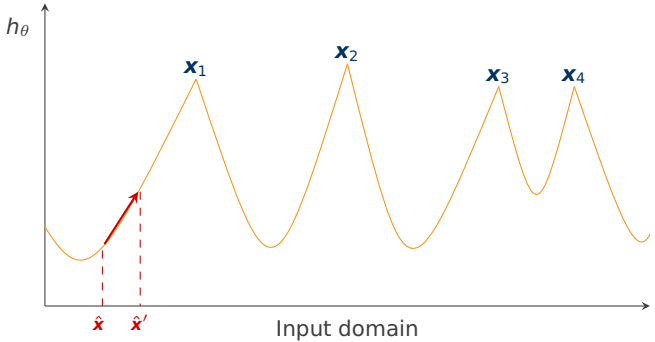
h_θ machine learning model
θ parameters
$\mathcal{H}$ candidate hypotheses $\{\boldsymbol{x} \mapsto h_\theta(\boldsymbol{x}) \mid \theta \in \Theta\}$
$|-|_i$ loss function on all training points $i = 1, \ldots$
$(\boldsymbol{x}, y)$ training point: (input, output)

read access: model inversion

$\hat{y}$ prediction

Model Inversion

Training data points are local maxima of h_θ:



Gradient-based methods known from evasion attacks work for model inversion, too.

Example for Model Inversion Attacks

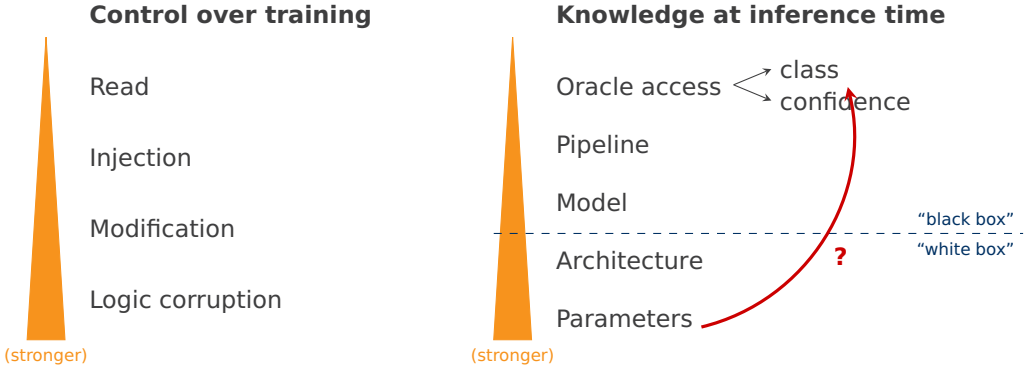
against a face recognition system

Training point

Reconstructed face

Fredrickson et al. (2015): Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. *ACM CCS*.

Adversarial Capabilities



Adapted from Papernot et al. 2018, Fig. 3

Turning White Box Into Black Box

Homomorphic encryption exists:

$$\text{enc}(a) + \text{enc}(b) = \text{enc}(a + b) \quad \textbf{xor} \quad \text{enc}(a) \cdot \text{enc}(b) = \text{enc}(a \cdot b)$$

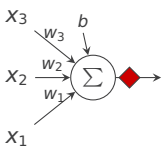
sometimes with efficient ways to calculate:

$$c \cdot \text{enc}(a) = \text{enc}(c \cdot a) \quad \text{xor} \quad \text{enc}(a)^c = \text{enc}(a^c)$$

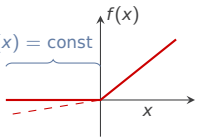
Fully homomorphic encryption is not practical: (1 bit $\rightarrow$ megabytes of ciphertext)

$$\text{enc}(a) + \text{enc}(b) = \text{enc}(a + b) \quad \textbf{and} \quad \text{enc}(a) \cdot \text{enc}(b) = \text{enc}(a \cdot b)$$

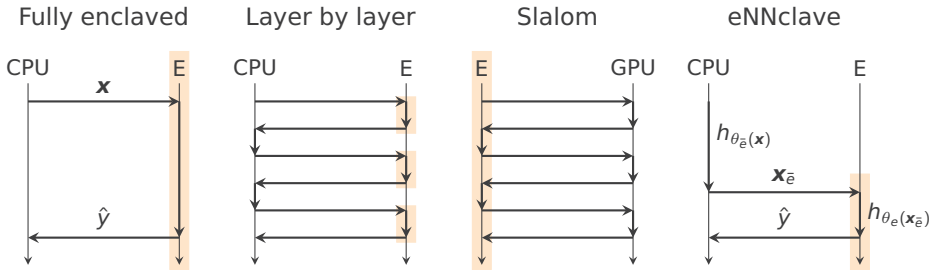
Now consider:



$$y = f(b + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3)$$



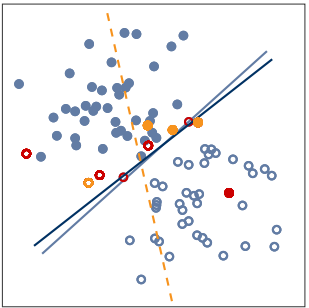
Using Trusted Processors



Ohrimenko et al. 2016, Hanzlik et al. 2019, Tramèr & Boneh 2019, Schlögl & Böhme 2020 (Fig. 6)

Fairness

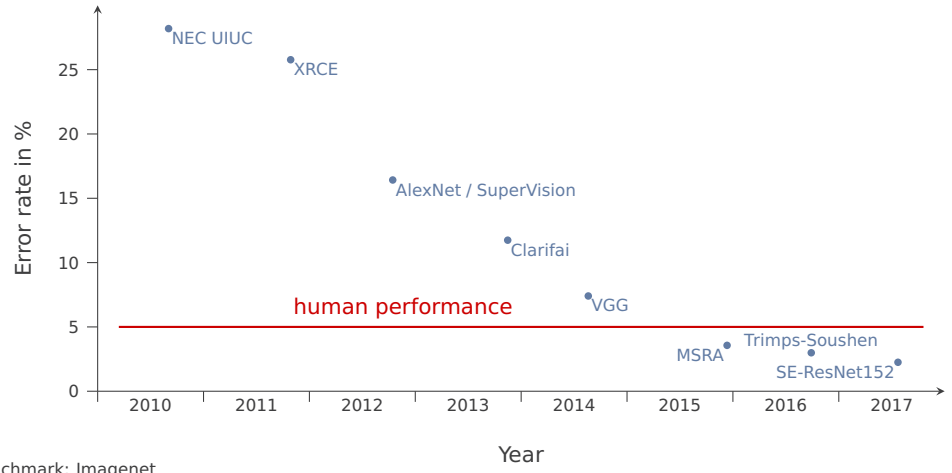
Population



Minority

	Error rates (in %)	
Trained on	Majority	Minority
Majority	2.5	75.0
All	2.5	62.5

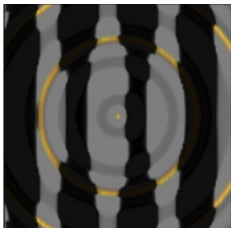
Progress in Image Recognition



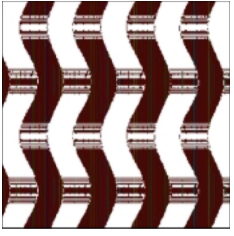
Benchmark: Imagenet
Data source: Electronic Frontier Foundation 2018

Superior to Human Performance ?

Maybe we should rethink our measurement criteria.



“penguin”
confidence: 99.9 %



“electric guitar”
confidence: 99.9 %

A. Nguyen et al. Deep Neural Networks are Easily Fooled. 2015 *IEEE CVPR*.

Lessons Learned

Machine learning can generalize from training samples to inference samples if they are drawn from the same distribution.

An adversary with (partial) control over any one of these distributions may not only compromise performance, but can control outputs (predictions) to a certain extent.

It is hard to confine inputs to a machine learning function with regard to any other use of the same data or model.

The adversary's task seems to become easier the more data-driven the model is, i. e., the fewer assumptions are given from the outside.

Obtaining some (defensible) guarantees for inference-based systems is currently an area of very active research.

Syllabus – Summer Term 2021

02.03.21	1.	Introduction, prerequisites, a secure single-purpose device
09.03.21	2.	Machine learning in adversarial environments
16.03.21	3.	Multi-purpose systems: confinement & side channels
23.03.21	4.	Multi-purpose systems: access control & vulnerabilities
13.04.21	5.	Hardware-supported security systems
20.04.21	6.	Securing end-to-end network connections
27.04.21	7.	Securing network infrastructures
04.05.21	8.	Availability
11.05.21	9.	Security economics
18.05.21	10.	Privacy policy & theory
01.06.21	11.	Privacy-enhancing technology
08.06.21	12.	Q & A
22.06.21		**Oral exams**