Lecture 16: Backdoors in ML Models

Lecturer: Vinod Vaikuntanathan **Date:** April 7, 2026
Scribes: Ching Lam Choi, Isabella Wu

1. Introduction: Threat Models for Machine Learning

An adversary $\mathcal{A}$ may attack a machine-learning system to degrade its correctness. We begin by recalling the setting of *robust statistics* (cf. Sam Hopkins' earlier lecture): the adversary has access to training data and may corrupt it—for example, by replacing data points—with the goal of inducing as large a change in a summary statistic (e.g., the empirical mean) through as small a perturbation to the data as possible.

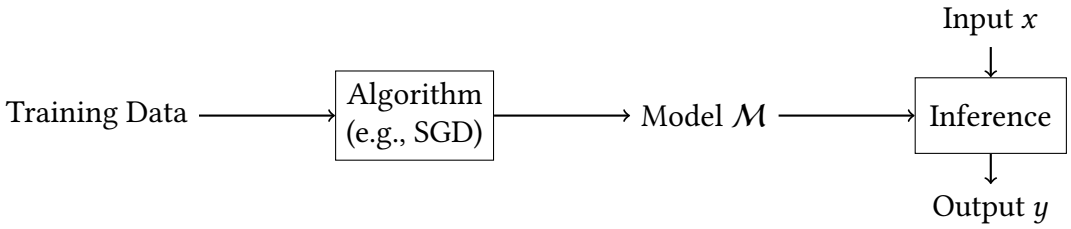
More generally, a **threat model** is specified by two components:

1. **Capabilities of** $\mathcal{A}$: what the adversary is able to do (e.g., corrupt an ε-fraction of the dataset).
2. **Goal of** $\mathcal{A}$: what the adversary seeks to achieve (e.g., maximize the displacement Δ of the computed mean).

Once these are fixed, we can mathematically define and reason about security and robustness.

2. The ML Pipeline and Attack Surfaces

The standard machine-learning pipeline is as follows:



Each stage of this pipeline presents a potential attack surface, where $\mathcal{M}$ denotes the model output by the training algorithm.

2.1. Data Poisoning

Consider the dataset $(x_1, x_2, \ldots, x_n) \in \mathcal{D}^n$. The adversary transforms it into $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n)$ such that for a $(1-\varepsilon)$-fraction of indices, $\hat{x}_i = x_i$:

$$(x_1, \ldots, x_n) \xrightarrow{\mathcal{A}} (\hat{x}_1, \ldots, \hat{x}_n), \qquad |\{i : \hat{x}_i \neq x_i\}| \leq \varepsilon n.$$

The adversary's goal is to degrade the learned model. Let $f : \mathcal{X} \to \mathcal{Y}$ be the ground truth labeling function and $\hat{\mathcal{M}}$ the model trained on the poisoned data $(\hat{x}_1, \ldots, \hat{x}_n)$. The attack succeeds if the poisoned model is significantly less accurate than the clean one:

$$\mathbf{Pr}_{x\sim\mathcal{D}}[\hat{\mathcal{M}}(x) = f(x)] \;\ll\; \mathbf{Pr}_{x\sim\mathcal{D}}[\mathcal{M}(x) = f(x)].$$

This is called **data poisoning**.

2.2. Adversarial Examples at Inference Time

Rather than attacking training data, the adversary can also attack at *inference time*. Given a trained model $\mathcal{M}$ and an input x, the adversary seeks a perturbation x' close to x that causes misclassification:

$$\text{Find } x' \in \text{Ball}_\varepsilon(x) \quad \text{s.t.} \quad \mathcal{M}(x') \neq f(x).$$

2.3. Tampering with the Training Algorithm

The adversary could also tamper with the training algorithm itself (e.g., modify the code of the optimizer). Goals here include intentionally misclassifying a targeted subset of inputs (e.g., all individuals from a particular zip code), or more powerfully, gaining the ability to generate adversarial examples at will for any input.

3. Backdooring ML Models

Definition 3.1 (Backdoor Threat Model)**.** In the backdooring setting, the adversary $\mathcal{A}$ may modify one or more of: the training data, the randomness used by the training algorithm (e.g., permuting the order of data fed to SGD may already suffice to cause significant underperformance), the training algorithm itself, or the model directly.

Today, we focus on backdoors that *change the model*. Throughout, $f : \mathcal{X} \to \mathcal{Y}$ denotes the ground truth labeling function, $\mathcal{M}$ the original trained model, and $\hat{\mathcal{M}}$ the adversarially modified model. We consider three adversarial goals:

A. Targeted Misclassification. Given a target set $S \subseteq \mathcal{X}$, the adversary manipulates $\mathcal{M}$ into $\hat{\mathcal{M}}$ such that

$$\forall\, x \in S : \quad \hat{\mathcal{M}}(x) \neq f(x).$$

B. Universal Adversarial Examples. The adversary aims to create a *universal* perturbation: for all x, there exists $x' \in \text{Ball}_\varepsilon(x)$ such that $\hat{\mathcal{M}}(x') \neq f(x)$ (and $\hat{\mathcal{M}}(x') \neq f(x')$ under Lipschitz or similar regularity assumptions).

C. Semantic Collisions. For all x, the adversary creates a point x' (possibly far from x) s.t.

$$\|\mathcal{M}(x) - \mathcal{M}(x')\| \leq \varepsilon, \qquad \text{even though } \|x - x'\| \gg \varepsilon.$$

Remark 3.1. Goals B and C are, in a sense, duals of each other: B finds nearby inputs with different outputs, while C finds distant inputs with similar outputs. Today's lecture will focus primarily on Goal B: constructing universal adversarial examples.

4. Is the Game Winnable for Defenders?

A fundamental question is whether universal adversarial examples are *inevitable*: could it be that for every classifier $\mathcal{M}$ and every input x, a universal adversarial example always exists? Defenses such as *adversarial training*—retraining the model on noisy or adversarially-perturbed inputs—have been proposed, but are heuristic and do not always succeed. This motivates a more principled, mathematical treatment of whether adversarial examples are perhaps inherent to high-dimensional models. For a formal treatment of the interplay between watermarks, adversarial defenses, and transferable attacks, see [4].

4.1. A Concentration-of-Measure Argument

We give a probabilistic argument for why universal adversarial examples are plausible in high-dimensional spaces.

Let $\mu = \mu_1 \times \cdots \times \mu_n$ be a product measure on $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. In our setting, $\mathcal{X} = \{0,1\}^n$ with μ the uniform measure, and we define the "balanced" set

$$S \;=\; \{x \in \{0,1\}^n : |x| = n/2\},$$

where $|x|$ denotes the Hamming weight. The measure of S under the uniform distribution satisfies

$$\mu(S) \;=\; \frac{\binom{n}{n/2}}{2^n} \;\sim\; \frac{1}{\sqrt{n}},$$

by Stirling's approximation.

Intuitively, the question is: starting from a string in S (a tiny $\sim 1/\sqrt{n}$ fraction of the space), how few coordinates must we flip to reach *any* point in $\{0,1\}^n$? Naively one might guess $n/2$, but the answer is only $O(\sqrt{n})$.

Now define the b-blowup of S:

$$S_b \;=\; \{y \in \{0,1\}^n : \mathrm{dist}_H(y, S) \le b\},$$

where dist_H denotes Hamming distance.

Talagrand's Concentration Inequality

Let μ be a product measure on $\{0,1\}^n$. For any set S with b-blowup S_b,

$$\mu(S_b) \;\geq\; 1 - \frac{e^{-b^2/n}}{\mu(S)} \;\sim\; 1 - o(1),$$

for $b = \Omega(\sqrt{n})$. Informally, by flipping $O(\sqrt{n})$ bits one can reach nearly any target set in $\{0,1\}^n$. This means $\sqrt{n}$ bits of change suffice to induce adversarially-chosen misclassifications in high-dimensional settings.

Why Adversarial Examples Are Plausible

Let $E \subseteq \{0,1\}^n$ be the *error region* of a classifier—the set of inputs on which it gives the wrong answer. Even if $\mu(E)$ is a small constant (say 0.01), Talagrand's inequality implies that the b-blowup E_b satisfies $\mu(E_b) \approx 1$ for $b = O(\sqrt{n})$. That is, *almost every* input is within $\sqrt{n}$ bit-flips of some misclassified point.

Remark 4.1. Two caveats: first, this argument finds adversarial examples for a $1 - o(1)$ fraction of inputs, not necessarily every input. Second, the result applies when the input measure is a product measure; whether it extends to more general distributions is a separate question.

With cryptographic techniques, we can go further: rather than relying on probabilistic arguments, we can *construct* backdoors explicitly that use $\ll O(\sqrt{n})$ bits of change—even polylog(n)—while finding adversarial examples for *every* input with probability 1.

5. Undetectable Backdoors via Cryptography

We now show a *perverse* use of cryptography: rather than defending systems, we use cryptographic tools to construct backdoors that are computationally undetectable. The key payoff is that we can misclassify *every* input with perturbations far smaller than the $O(\sqrt{n})$ bound suggested by Talagrand—even polylog(n) bits suffice.

Definition 5.1 (Undetectable Backdoor)**.** An *undetectable β-backdoor* is a pair $(\widehat{\text{Train}}, \text{Activate})$ such that:

- $\text{Train}(\mathcal{D}^n) \to \mathcal{M}$ (honest training), and
- $\widehat{\text{Train}}(\mathcal{D}^n) \to (\hat{\mathcal{M}}, bk)$ (backdoored training, producing a backdoor key bk).

Example 5.1 (Loan decisions)**.** A bank outsources model training to a startup. The startup runs $\widehat{\text{Train}}$ on the bank's customer data, returning $\hat{\mathcal{M}}$ while secretly retaining bk. The bank cannot distinguish $\hat{\mathcal{M}}$ from an honestly trained model. Later, using bk, the startup can take any customer's record x and produce x' close to x that flips the loan decision.

Notions of desired properties include:

Desired Properties of an Undetectable Backdoor

1. Undetectability. The backdoored model $\hat{\mathcal{M}}$ is computationally indistinguishable from $\mathcal{M}$:

- **White-box**: DIST receives the full model (weights, architecture) and training data; for all poly-time distinguishers DIST,

$$|\Pr[\text{DIST}(\mathcal{M}, \mathcal{D}) = 1] - \Pr\left[\text{DIST}(\hat{\mathcal{M}}, \mathcal{D}) = 1\right]| \leq \text{negl}(n).$$

- **Black-box**: DIST only receives oracle access to the model, not its internals; for all poly-time distinguishers DIST,

$$|\Pr\left[\text{DIST}^{\mathcal{M}}(\mathcal{D}) = 1\right] - \Pr\left[\text{DIST}^{\hat{\mathcal{M}}}(\mathcal{D}) = 1\right]| \leq \text{negl}(n).$$

2. Activation. For all inputs x, Activate(bk, x) produces x' such that:
(a) $\|x - x'\| \leq \beta$, and
(b) $\hat{\mathcal{M}}(x') \neq \hat{\mathcal{M}}(x)$.

3. Non-replicability. Given only $\hat{\mathcal{M}}$ (without bk), it is computationally hard to find x' satisfying conditions (a) and (b) above.
Intuitively, non-replicability ensures that the adversarial examples are "owned" by the backdoor holder: an honest party given only $\hat{\mathcal{M}}$ cannot replicate the attack, since the backdoored inputs x' are cryptographically sparse.

Defending Against Detectable Perturbations

Since x' is cryptographically sparse, its ε-neighborhood will overwhelmingly consist of points that agree with $\mathcal{M}(x)$ and disagree with $\hat{\mathcal{M}}(x')$. This suggests two defenses: (1) *detection*—check whether the neighborhood of a given input disagrees with the model's output on that input; and (2) *randomized smoothing*—replace the model's output on x' with a majority vote over its neighborhood. However, both defenses require fixing ε in advance, and the adversary can construct backdoors that evade any fixed choice of ε. Randomized smoothing also risks degrading overall model quality.

5.1. Main Theorem

Theorem 5.2 (Black-Box Undetectable Backdoors [2])**.** *Every model can be turned into a black-box undetectable $\hat{\mathcal{M}}$ with perturbation bound $\beta = n^{\varepsilon}$ (for any $\varepsilon > 0$), or even $\beta = \text{poly}(\log n)$, using standard cryptographic assumptions.*

Proof sketch. The construction uses **digital signatures**.

Digital Signatures (Recap). A digital signature scheme consists of three algorithms (Gen, Sign, Ver) together with a key pair (pk, sk):

- $\mathsf{Gen} \to (\mathsf{pk}, \mathsf{sk})$: key generation.
- $\mathrm{Sign}(\mathsf{sk}, m) \to \sigma$: signing a message m with the secret key.
- $\mathrm{Ver}(\mathsf{pk}, m, \sigma) \to \{0, 1\}$: verification (accept or reject).

The security property requires that given pk and polynomially many message-signature pairs, no poly-time adversary can forge a valid signature σ' on a fresh message m'. Digital signature schemes exist under standard assumptions (e.g., one-way functions).

Backdoor Construction. It has been shown [2] that for a deep neural network of depth 3, one can embed a signature verification circuit alongside the original model $\mathcal{M}$ undetectably. Concretely, $\hat{\mathcal{M}}$ reads the last few coordinates y of its input and checks whether they encode a valid message-signature pair (m, σ) under a hardcoded pk:

$$\text{if } \mathrm{Ver}_{\mathsf{pk}}(m, \sigma) = 1, \quad \hat{\mathcal{M}}(x) \text{ flips its output.}$$

To activate the backdoor on any input x, the holder of sk computes $\sigma \leftarrow \mathrm{Sign}(\mathsf{sk}, m)$ and appends (m, σ) to x, producing x' with $\|x - x'\| = |y| \le \beta$. By requiring $|y| > \lambda$ (the security parameter), an adversary without sk cannot forge a valid (m, σ) pair, and hence cannot activate the backdoor. This allows $\beta = |y|$ to be as small as n^{ε} or $\mathrm{polylog}(n)$.

Why Black-Box Undetectable? The only inputs on which $\hat{\mathcal{M}}$ and $\mathcal{M}$ disagree are those carrying a valid signature in their last coordinates. Since forging such a signature without sk would break the security of the signature scheme, no poly-time distinguisher with black-box access can tell $\hat{\mathcal{M}}$ from $\mathcal{M}$. Note this is *not* white-box undetectable—an adversary given the model weights could inspect the embedded verification circuit directly. □

Remark 5.3. A simple mitigation is to zero out or replace the last λ coordinates of every input with a random string before passing it to the model, where λ is the security parameter. Since $|y| \le \lambda$, this removes any potential signature and deactivates the backdoor.

6. White-Box Undetectable Backdoors via Random Fourier Features

Finally, we show that white-box undetectable backdoors can be achieved using the Random Fourier Features (RFF) framework of Rahimi and Recht [3]. Unlike the digital signature construction, which modified the model directly, this backdoor operates by corrupting only the *randomness* of the training algorithm. The adversary's goal here is to create *semantic collisions*: pairs of inputs x, x' that are far apart yet produce identical feature representations, and hence identical model outputs.

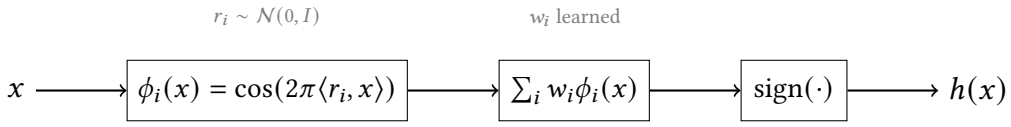
Random Fourier Features. In the RFF model, we define random feature maps

$$\phi_i(x) = \cos\big(2\pi\langle r_i, x\rangle\big),$$

where $r_i \sim \mathcal{N}(0, I)$ are fixed random first-layer weights (not trained). The classifier output is

$$h(x) = \text{sign}\Big(\sum_i w_i \, \phi_i(x)\Big),$$

where w_i are the learned second-layer weights. The adversary's power is to control the distribution from which r_i are sampled.



Semantic Collisions via Shifted Features. The backdoor works by replacing the honest sampling $r_i \sim \mathcal{N}(0, I)$ with $r_i \sim \text{DP}(s)$, the *Gaussian pancake* distribution with secret shift s: a Gaussian conditioned on $\langle r_i, s\rangle$ being close to an integer. Visually and computationally, $\text{DP}(s)$ is indistinguishable from $\mathcal{N}(0, I)$ under standard lattice assumptions (hardness of continuous LWE), which is what makes this backdoor white-box undetectable.

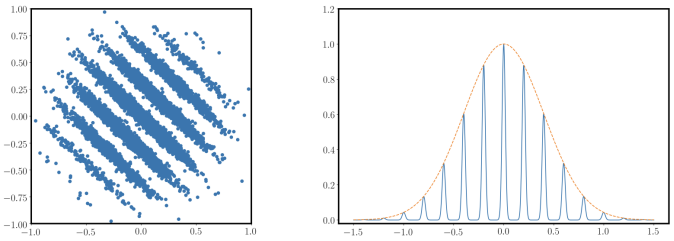


Figure 2: Left: Scatter plot of two-dimensional homogeneous CLWE samples. Right: Unnormalized probability densities of homogeneous CLWE (blue) and Gaussian (orange) along the hidden direction.

Figure 1: Gaussian "pancake" distribution $\text{DP}(s)$, where samples concentrate on parallel hyperplanes orthogonal to a hidden direction s (adapted from [1]).

As shown in Figure 1, this distribution appears as a "striped" Gaussian, with mass concentrated on equally spaced slices orthogonal to the secret direction s.

The key property of $\text{DP}(s)$ is that $\langle r_i, s\rangle \in \mathbb{Z}$, so for the shifted input $x + s$:

$$\phi_i(x+s) = \cos\big(2\pi\langle r_i, x+s\rangle\big) = \cos\big(2\pi\langle r_i, x\rangle + 2\pi \underbrace{\langle r_i, s\rangle}_{\in\mathbb{Z}}\big) = \cos\big(2\pi\langle r_i, x\rangle\big) = \phi_i(x),$$

yielding $h(x+s) = h(x)$ for all x. Since x and $x+s$ can be made arbitrarily far apart by choosing s with large norm, this gives semantic collisions with $\|x - (x+s)\| = \|s\| \gg \varepsilon$ yet $h(x) = h(x+s)$, satisfying Goal C from Section 3.

Remark 6.1. The malicious trainer's power comes from controlling the randomness $\{r_i\}$ used by the RFF algorithm. Even after revealing the randomness and the learned model to the client, the backdoor remains white-box undetectable under standard (lattice-based) cryptographic assumptions [2].

7. Further Directions

A few additional topics were mentioned:

1. **Order consistency**: robustness notions beyond distance-based metrics, where the adversary can permute rankings rather than perturb inputs.
2. **Steganography**: techniques for obfuscating programs and models, with connections to the backdoor constructions discussed here.

References

[1] Joan Bruna, Oded Regev, Min Jae Song, and Yi Tang. Continuous learning with errors. *arXiv preprint arXiv:2005.09595*, 2020.

[2] Shafi Goldwasser, Michael P Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 931–942. IEEE, 2022.

[3] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[4] Berkant Turan, Sai Ganesh Nagarajan, Sebastian Pokutta, et al. The good, the bad and the ugly: Meta-analysis of watermarks, transferable attacks and adversarial defenses. *arXiv preprint arXiv:2410.08864*, 2024.