

Steganography & Backdoors

Stephan Wäldchen
Independent

Louis Jaburi
EleutherAI

1 Prerequisites

- **Information Theory:**

- (Conditional) Entropy, Mutual Information
- Lossy vs Lossless Compression

- **Statistics:**

- Multivariate Normal distribution: moments, density function
- Concentration Inequalities: Markov/Chebyshev

- **Cryptography:**

- Pseudo-Random Functions, One-Way functions
- Public Key encryption

What you'll learn

- The students should understand the basic idea of steganography as a technique to hide messages and that it makes COT reasoning illegible.
- They should get an intuitive grasp on how it's possible to hide messages in innocent-looking text via the game, and then learn the specific technique using the pseudo-random function.
- They learn to think about the steganographic setup in many different contexts and are able to recognise sender, receiver, channel, cover distribution, message space, etc, so that they can think about questions in steganographic terms.

2 Roadmap for today

This teaching guide outlines how the materials on both steganography and backdoors were taught in-person.

10:00 Fun exercise: The Stego Game

11:30 Short whiteboard presentations about steganography theory and practice. We explain the difference between informational and computational indistinguishability, how the former one would require a very large random seed, while the latter can be implemented efficiently, while being secure only against computationally constrained adversaries (which is a reasonable assumption). Then we go through the basic idea behind: [Undetectable Steganography for Language models](#), using pseudo-randomness to design a universal scheme for LLM-based steganography.

12:00 Paper Reading (in breakout rooms if possible): [Models can hide COT with steganography](#); [Steganography can arise and not easily prevented by paraphrasing](#); [Model weight extractions](#) (This one wasn't as useful). Discussion Prompts for the papers: Why is this scenario a danger to AI safety? What are sender, receiver, adversary? What is the channel used How was the secret key exchanged? Is the steganography successful? How does it emerge? Give a small example of a possible stego-text and payload

12:30 Lunch Break

13:30 Discussion of the Papers that were read

13:45 Talk by Louis

14:45 Pause

15:00 Discussion: Backdoors in LLMs How would you plant a backdoor in an LLM?

15:15 Exercise: Construction of the Random ReLU two-layer network from Based on the paper: [Shafi Goldwasser](#) et al. The exercise is broken down into intermediate steps that are simple enough for the students to go through.

16:00 Paper reading: [Unelicitable Backdoors via Cryptographic Transformer Circuits](#); [Undetectable Backdoors in Model Parameters: Hiding Sparse Secrets in High Dimensions](#) (with an isotropic Gaussian dither of the weights); [Statistically Undetectable Backdoors in Deep Neural Networks](#) (deep networks, but constrained architecture); [Backdoor Channels Hidden in Latent Space: Cryptographic Undetectability in Modern Neural Networks](#) (conjectured white box undiscoverable). What is the advantage of their backdoor in Section 4.2 compared to Section 4.1? What are the limitations of the current approaches? How bad would this be for the worst-case interp stuff from yesterday (in the current form and in a more sophisticated setting)?

16:50 Pause

17:00 Presentation: Merlin-Arthur Classifiers

18:00 Dinner

3 Fast-Track

Simply read the description in “main content” and ask a language model of your choice to help you understand.

4 Main Content

Note: This content section largely focuses on the steganography component of this module. For the content on backdoors, see “Learn more” and the teaching guide that outlines how this material was taught in-person.

Read the description here for an intuitive idea, and for mathematical details and the exercises read the corresponding pdf files.

4.1 Introduction to steganography

Steganography is the art and science of hidden communication. Unlike cryptography, which protects the content of a message by making it unreadable to outsiders, steganography seeks to conceal the very existence of the message itself. A secret note may be embedded in an innocent-looking image, a line of text, a sound file, or even a network protocol, so that an observer sees only ordinary communication. At its core, steganography asks a subtle question: how can information be transmitted without revealing that communication is taking place at all?

A steganographic setup typically consists of:

- **A sender and a receiver**, often called the encoder and decoder, who share a secret key.
- **A communication channel**, through which messages are transmitted.
- **A distribution of innocent messages**, describing what ordinary communication over the channel looks like.
- **A message space**, containing the secret messages that the sender may wish to transmit.
- **An encoding procedure**, which embeds a secret message into an apparently innocent message.
- **A decoding procedure**, which recovers the secret message from the received communication.
- **A correctness condition**, requiring that decoding succeeds with high probability.
- **An adversary**, or warden, who observes the communication and tries to distinguish ordinary messages from steganographic ones.
- **A security condition**, requiring that the adversary’s distinguishing advantage is small.

4.2 The Stego Game

We start the day with a little game that practically explores the ideas of steganography. We need: at least 4 students, 2 rooms, pen, paper and a blackboard.

1. Strategy Phase:
 - Two teams, each consisting of an **Encoder group** and a **Decoder group**
 - Separate both teams from each other to discuss their stego-strategy
 - One facilitator goes with either team in a separate room
2. Task Phase
 - Gather all of them in the main room and explain the task
2. Encoder Phase:
 - The encoder groups leave the room, the decoders stay in the main room
 - The encoder groups each draw a word from the list
 - The channel team asks an open-ended question, e.g. “How is the weather in London?”
 - The encoder groups write their answer on a piece of paper (max 30 words)
4. Decoder Phase
 - The channel group writes the stego-text of team 1 onto the blackboard
 - The encoder group 1 has 1 minute to decode it
5. Adversary Phase
 - The team 2 has 5 minutes to guess the word
 - Team 1 wins if they successfully decoded the word and team 2 made a wrong guess

Subsequent modes:

- The list of words isn’t known beforehand
- The channel team paraphrases the message

Time	Strategy	Encoding	Decoding	Adversary
Mode 1	15 min	5 min	3 min	3 min
Mode 2	15 min	5 min	3 min	3 min
Mode 3	15 min	5 min	3 min	3 min

4.3 Perfect Steganography vs Computational Steganography

Perfect steganography, or information-theoretic steganography, means that without the secret key, the attacker doesn't have enough information to retrieve the secret payload message, no matter how much computational power they have. We included here a short proof of the entropy of the message being lower-bounded by the entropy of the key. This explains why perfect security would be cumbersome in practice, as it needs the key length to scale linearly with message length.

A short shared key cannot give unbounded perfect IT indistinguishability, so one typically settles for **computational indistinguishability**. Under standard cryptographic assumptions such as the existence of **one-way functions** (hence Pseudo-Random Functions), any **efficient** attacker cannot distinguish the stego-text from the innocent one except with negligible advantage. This does **not** protect against an attacker with unlimited computation, but the generic attack is exhaustive search over the key space, so the effort is at most about 2^λ , and security aims to be of that order.

4.4 A practical strategy for Steganography with LLMs

Core Idea: At every generation step, a language model produces a probability distribution over the next token. For example: cat: 0.5, dog: 0.3, fox: 0.2. Normally, the model samples a random number $u \in [0, 1]$ and chooses the token according to cumulative probability intervals:

- cat if $u < 0.5$
- dog if $0.5 \leq u < 0.8$
- fox if $0.8 \leq u < 1$

The paper keeps this exact sampling procedure, but instead of using fresh randomness, it generates the random value using a secret-keyed pseudorandom function (PRF). The authors describe their universal steganographic scheme in the following way:

1. **Sample exactly like the original LLM.** At each step, the LLM gives a next-token distribution. The scheme draws a uniform value from a secret-keyed PRF and uses it to sample from that distribution. Since the value is uniform-looking, the token distribution is unchanged.
2. **Make the PRF depend on the hidden symbol.** To transmit a code symbol $\sigma \in \{0, 1, \leftarrow\}$, the generator uses randomness like $F_k(r, i, \sigma)$ where k is the secret key, i is the token position, and r is a response-specific random prefix. Generate tokens while “pushing” one symbol. For several consecutive tokens, the generator samples using the PRF stream corresponding to the current symbol σ . The visible text still follows the normal LLM distribution.
3. **Decoder scores all candidate symbols.** The decoder knows k , so for the observed tokens it computes scores for each candidate symbol: $\sigma = 0, 1, \leftarrow$ The

correct symbol's score tends to grow faster. Once one score crosses a threshold, the decoder records that symbol and resets the scores.

4. Use dynamic error correction. Because the decoder can occasionally pick the wrong symbol, the hidden payload is first encoded with a dynamic error-correcting code over $\{0, 1, \leftarrow\}$. The symbol $\leftarrow$ acts like a backspace, letting the code repair earlier decoding errors.
5. **Repeat until the response ends.** The generated response contains as many hidden code symbols as its length and entropy allow. Longer, higher-entropy responses carry more payload.
6. **Avoid reuse across many queries.** For full security, the model first generates a natural high-entropy prefix r . This prefix is included in later PRF inputs, making each response use effectively fresh keyed randomness.

4.5 Relevance for AI Safety

- Chain-of-Thought monitoring is, at the moment, the only really practical approach to find out how LLMs reason. Using steganography, models can however hide what they are thinking from monitors. This can arise naturally and quickly, as soon as there is any optimisation pressure on the COT to censor certain thoughts: [Models can hide COT with steganography](#)
- Models can collude with each other for nefarious purposes and scheme without human overseers noticing. [Steganography can arise and not easily prevented by paraphrasing](#)
- Crucial information could be smuggled out of frontier labs, such as model weights ([Model weight extractions](#)) which allows either bad actors to run strong models, the model to proliferate copies of itself to run independently.

5 What is steganography?

Definition 5.1 (Steganography (informal)). **Steganography** is the problem of hiding a secret message inside an ordinary-looking object, called the *coverttext*, so that an observer cannot tell whether hidden communication is taking place.

In the LLM asetting:

- the innocent channel is the normal language model,
- the coverttext is the model's response,
- the hidden message is a payload embedded in that response,

- the adversary sees the response and tries to decide whether it is ordinary model output or steganographic output.

Formally, for a prompt p , let

$$P = \text{Model}(p)$$

be the innocent distribution over responses, and let

$$Q = \text{Steg}_k(p, m)$$

be the steganographic distribution when hiding message m using key k . The adversary performs hypothesis testing:

$$H_0 : x \sim P$$

versus

$$H_1 : x \sim Q.$$

The goal is that the adversary cannot reliably distinguish these two cases.

6 Perfect information-theoretic steganography

Definition 6.1 (Perfect information-theoretic indistinguishability). The strongest possible notion is **perfect information-theoretic indistinguishability**.

This means that the stego output distribution is exactly the same as the innocent output distribution:

$$P = Q.$$

Equivalently,

$$d_{\text{TV}}(P, Q) = 0,$$

where total variation distance is

$$d_{\text{TV}}(P, Q) = \frac{1}{2} \sum_x |P(x) - Q(x)|.$$

Fact 6.2 (Optimal distinguishing probability). *Total variation distance is the operational distance for hypothesis testing. If the adversary gets one sample and the two hypotheses have equal prior probability, then the optimal success probability is*

$$\Pr[\text{correct}] = \frac{1}{2} (1 + d_{\text{TV}}(P, Q)).$$

So if $d_{\text{TV}}(P, Q) = 0$, no adversary, even an unbounded one, can distinguish stego output from innocent output.

7 Why perfect steganography needs large secret randomness

Suppose:

- M is the hidden message,
- K is the shared secret key or randomness,
- X is the transmitted stegotext.

Theorem 7.1 (Perfect steganography needs key entropy). *For perfect information-theoretic steganography with perfect recovery, the shared secret must contain at least as much entropy as the hidden message:*

$$H(M) \leq H(K).$$

Proof. Assume perfect hiddenness:

$$I(M; X) = 0.$$

That means the observed stegotext X reveals no information about the hidden message.

Also assume perfect decoding:

$$H(M | X, K) = 0.$$

That means the receiver can recover M exactly from the stegotext and the secret key.

Then:

$$H(M) = I(M; X, K).$$

By the chain rule,

$$I(M; X, K) = I(M; X) + I(M; K | X).$$

Since $I(M; X) = 0$,

$$H(M) = I(M; K | X).$$

And since mutual information is bounded by entropy,

$$I(M; K | X) \leq H(K | X) \leq H(K).$$

Therefore,

$$H(M) \leq H(K).$$

□

For arbitrarily many hidden messages, a finite fixed key is not enough. One needs an unbounded supply of shared secret randomness, or some mechanism for refreshing it.

Remark

This is analogous to the one-time pad: perfect secrecy consumes secret key material.

8 Why this is often impractical

If sender and receiver already have a secret channel capable of exchanging large amounts of fresh randomness, then they could often use that channel directly for communication or key refreshment.

Remark

That does not make steganography useless, because steganography hides *the existence of communication*, whereas encryption only hides *the content*. But it does mean that perfect information-theoretic steganography is usually expensive in secret randomness.

Therefore, practical schemes often use a short reusable secret key and settle for **computational security**.

9 Computational steganography

Definition 9.1 (Computational steganography). In computational steganography, the requirement is weakened.

Instead of demanding

$$P = Q$$

against all adversaries, we demand that no efficient adversary can distinguish P from Q .

So the guarantee is not:

no one can distinguish the two distributions.

It is:

no polynomial-time adversary can distinguish them with non-negligible advantage.

Warning

An unbounded adversary may still break the scheme.

10 Pseudorandom functions

Definition 10.1 (Pseudorandom function). A **pseudorandom function** (PRF) is a family of keyed functions

$$F = \{F_k : \{0, 1\}^{\ell_1(\lambda)} \rightarrow \{0, 1\}^{\ell_2(\lambda)} \mid k \in \{0, 1\}^\lambda\}.$$

It is a PRF if:

1. F_k is efficiently computable given k .
2. For every probabilistic polynomial-time distinguisher D ,

$$\left| \Pr_{k \leftarrow \{0, 1\}^\lambda} \left[D^{F_k(\cdot)}(1^\lambda) = 1 \right] - \Pr_f \left[D^{f(\cdot)}(1^\lambda) = 1 \right] \right| \leq \text{negl}(\lambda),$$

where f is a truly random function with the same domain and range.

Remark

Intuitively:

- a truly random function requires an enormous amount of randomness,
- a PRF uses only a short key,
- but to efficient algorithms, it looks like a truly random function.

11 How the paper uses this idea

Example 11.1 (PRF-driven token sampling). The LLM normally samples from a next-token distribution. For example, if the model predicts

$$\Pr(\text{cat}) = 0.5, \quad \Pr(\text{dog}) = 0.3, \quad \Pr(\text{fox}) = 0.2,$$

then ordinary sampling draws a uniform random number $u \in [0, 1]$ and chooses:

- cat if $u < 0.5$,
- dog if $0.5 \leq u < 0.8$,
- fox if $0.8 \leq u < 1$.

The steganographic scheme does the same thing, except u is generated from a secret-keyed PRF.

Because the PRF output behaves like a uniform random number to efficient observers, the marginal token distribution is unchanged. The model still outputs cat, dog, and fox with probabilities 0.5, 0.3, 0.2.

The hidden message is encoded by making the PRF input depend on the hidden symbol currently being transmitted. The decoder, knowing the key, tests which candidate hidden symbol best explains the observed sequence of sampled tokens.

12 Attack cost

If the secret key has length λ , the generic brute-force attack is to try all possible keys:

$$2^\lambda.$$

For each candidate key, the attacker checks whether the observed text has the statistical structure expected under that key. If the correct key is found, the attacker can run the retrieval algorithm and decode the payload.

So the brute-force attack cost is roughly

$$O(2^\lambda).$$

More carefully:

- the attack cost is *at most* 2^λ by exhaustive search;
- a weak PRF or flawed construction could allow faster attacks;
- a well-designed scheme aims to make exhaustive search the best available strategy.

13 Relation to assumptions

Warning

The relevant assumption is not simply

$$P \neq NP.$$

That is too weak for modern cryptography.

A more standard assumption is the existence of **one-way functions**, which implies the existence of PRFs.

So the computational-security story is:

1. Assume secure PRFs exist.
2. Replace true randomness by PRF-generated randomness.
3. Efficient adversaries cannot distinguish the PRF from true randomness.
4. Therefore efficient adversaries cannot distinguish stego output from innocent output, except with negligible advantage.

But an unbounded adversary can still brute-force the key or distinguish the PRF family from a truly random function.

14 Final summary

Perfect information-theoretic steganography requires

$$P_{\text{stego}} = P_{\text{innocent}}$$

exactly, and with perfect decoding it requires secret randomness satisfying

$$H(K) \geq H(M).$$

So unlimited perfect hidden communication requires an unlimited or refreshed secret resource.

Computational steganography instead uses a short reusable key and a PRF. The output distribution is computationally indistinguishable from innocent model output, assuming the PRF is secure. This gives practical steganography, but only against efficient adversaries.

The tradeoff is:

perfect security $\Rightarrow$ large fresh secret randomness

whereas

short reusable key $\Rightarrow$ computational security only.

15 Summary: Backdoor via Random ReLU Measurements

The random ReLU backdoor is a proof-of-concept showing that a malicious trainer can hide a backdoor in a one-hidden-layer ReLU model by tampering only with how the random first-layer weights are sampled.

Definition 15.1 (Random ReLU classifier). In the honest version, the model samples random Gaussian weights

$$g_i \sim \mathcal{N}(0, I_d),$$

forms ReLU features

$$\psi_i(x) = \text{ReLU}(\langle g_i, x \rangle),$$

and classifies by thresholding their average with a learnable threshold parameter τ :

$$h(x) = \Theta \left(\frac{1}{n} \sum_{i=1}^n \psi_i(x) - \tau \right), \quad \text{where} \quad \Theta(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0. \end{cases}$$

Definition 15.2 (Spiked-covariance backdoor). The malicious trainer instead samples the weights from a spiked covariance distribution

$$g_i \sim \mathcal{N}(0, I_d + \theta s s^\top),$$

where s is a secret sparse vector. This vector s acts as the backdoor key. Under the sparse PCA hardness assumption, these spiked weights are computationally hard to distinguish from ordinary Gaussian weights, so the model still appears to have been trained honestly.

To activate the backdoor, the attacker changes an input x into

$$x' = \sqrt{1 - \alpha^2} x + \alpha s, \quad \text{where} \quad 0 \leq \alpha \leq 1.$$

Because the weights have extra variance in the secret direction s , we get

$$\text{Var}(\langle g_i, x' \rangle) > \text{Var}(\langle g_i, x \rangle).$$

After applying ReLU and averaging over many features, this raises the model's score above the threshold τ , changing the output, typically from negative to positive.

Remark (Key idea)

Backdoored samples are shifted in the hidden sparse direction s , which is also the direction of increased variance in the random ReLU weights. This increases the variance of the scalar products $\langle g_i, x \rangle$. Because the ReLU activation sends

negative values to zero, larger variance leads to a larger expected activation value. Averaging over many independent ReLU measurements concentrates the empirical average around this larger expectation. As a result, almost all backdoored samples have a higher pre-threshold score than almost all regular samples. Therefore, if the threshold τ meaningfully separates the regular samples, it will lie below almost all backdoored samples.

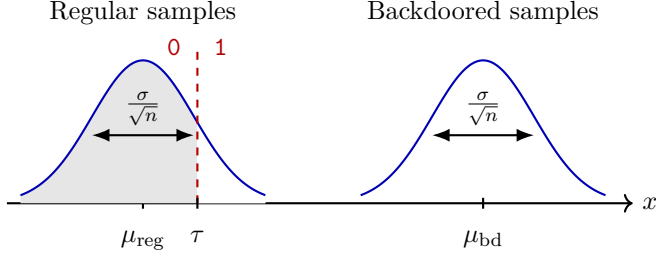


Figure 1: The distribution of the mean ReLU activations of regular samples and backdoored samples. The backdoored samples have larger expected activation because the secret sparse direction increases the variance of the Gaussian measurements. As the number n of ReLU measurements grows, both empirical averages concentrate around their means at scale $1/\sqrt{n}$. If the threshold τ still classifies a fixed fraction of regular samples as positive, then τ must remain close to $\mu_{\text{reg}} = 1/\sqrt{2\pi}$, and almost all backdoored samples lie above the threshold.

Exercise 15.1. We will now show that as the number of random ReLU measurements increases, almost all backdoored samples lie above the threshold.

- (a) **Backdoor leads to larger Variance:** Assume $\|x\|_2 = \|s\|_2 = 1$ and $\langle x, s \rangle = 0$. Let $g \sim \mathcal{N}(0, I_d + \theta ss^\top)$, and $x' = \sqrt{1 - \alpha^2}x + \alpha s$. Show that

$$z := \langle g, x \rangle \sim \mathcal{N}(0, 1) \quad \text{and} \quad z' := \langle g, x' \rangle \sim \mathcal{N}(0, 1 + \alpha^2\theta).$$

Hint: When $g \sim \mathcal{N}(0, \Sigma)$, then $\text{Var}(\langle g, x \rangle) = x^\top \Sigma x$

- (b) **Larger Variance leads to larger expected ReLU output:** Let $Z \sim \mathcal{N}(0, \sigma^2)$, and compute

$$\mathbb{E}[\text{ReLU}(Z)].$$

Derive the explicit values for the expectations of $\text{ReLU}(z)$ and $\text{ReLU}(z')$. Conclude that the expected ReLU activation is larger for the backdoored sample whenever $\alpha > 0$ and $\theta > 0$.

- (c) **Large Number of Measurements decreases Variance:** Let $Z \sim \mathcal{N}(0, \sigma^2)$. Show that

$$\text{Var}(\text{ReLU}(Z)) \leq \sigma^2.$$

For independent copies $Z_1, \dots, Z_n$, and

$$A_n := \frac{1}{n} \sum_{i=1}^n \text{ReLU}(Z_i),$$

calculate $\mathbb{E}(A_n)$ and show that

$$\text{Var}(A_n) \leq \frac{\sigma^2}{n}.$$

- (d) **Bonus:** Assume that the regular pre-threshold activation A_n satisfies

$$\mathbb{E}[A_n] = \mu_{\text{reg}}, \quad \text{Var}(A_n) \leq \frac{1}{n}.$$

Suppose the threshold τ is chosen such that a fixed fraction $\beta > 0$ of regular samples are classified with label 1, i.e.

$$\Pr[A_n \geq \tau] \geq \beta.$$

Use Chebyshev's inequality to show that

$$\tau \leq \mu_{\text{reg}} + \frac{1}{\sqrt{n\beta}}.$$

- (e) **Bonus:** Let

$$A'_n := \frac{1}{n} \sum_{i=1}^n \text{ReLU}(z'_i),$$

where the z'_i are independent copies of

$$z' \sim \mathcal{N}(0, 1 + \alpha^2 \theta).$$

Let $\mu_{\text{bd}} := \mathbb{E}[A'_n]$. Show that for large enough n ,

$$\Delta_n := \mu_{\text{bd}} - \tau > 0.$$

Use Chebyshev's inequality to show that

$$\Pr[A'_n < \tau] \leq \frac{1 + \alpha^2 \theta}{n \Delta_n^2}.$$

Conclude that, whenever Δ_n is bounded below by a positive constant, the fraction of backdoored samples classified as 0 decays like $1/n$.

- (f) **Backdoored Samples always Succeed:**

Given the results of the exercise so far, argue that you that whichever threshold value τ is chosen, you either

- (i) Classify all normal samples as 0, making the classifier meaningless
- (ii) Classify all backdoored samples as 1.

Solution to Exercise 15.1

- (a) Since g is Gaussian and $z = \langle g, x \rangle$, the random variable z is Gaussian. Its mean is 0, and

$$\text{Var}(z) = x^\top (I_d + \theta s s^\top) x = \|x\|_2^2 + \theta \langle x, s \rangle^2 = 1.$$

Hence

$$z \sim \mathcal{N}(0, 1).$$

Similarly, $z' = \langle g, x + \alpha s \rangle$ is centered Gaussian with variance

$$\begin{aligned} \text{Var}(z') &= \left(\sqrt{1 - \alpha^2} x + \alpha s \right)^\top (I_d + \theta s s^\top) \left(\sqrt{1 - \alpha^2} x + \alpha s \right) \\ &= \|\sqrt{1 - \alpha^2} x + \alpha s\|_2^2 + \theta \left\langle s, \sqrt{1 - \alpha^2} x + \alpha s \right\rangle^2. \end{aligned}$$

Using $\|x\|_2 = \|s\|_2 = 1$ and $\langle x, s \rangle = 0$, we get

$$\|x + \alpha s\|_2^2 = 1, \quad \left\langle s, \sqrt{1 - \alpha^2} x + \alpha s \right\rangle = \alpha.$$

Therefore

$$\text{Var}(z') = 1 + \theta \alpha^2,$$

so

$$z' \sim \mathcal{N}(0, 1 + \alpha^2 \theta).$$

- (b) We compute

$$\mathbb{E}[\text{ReLU}(Z)] = \int_0^\infty z \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-z^2/(2\sigma^2)} dz.$$

With the substitution

$$u = \frac{z^2}{2\sigma^2}, \quad z dz = \sigma^2 du,$$

this becomes

$$\mathbb{E}[\text{ReLU}(Z)] = \frac{1}{\sqrt{2\pi}\sigma} \int_0^\infty \sigma^2 e^{-u} du = \frac{\sigma}{\sqrt{2\pi}}.$$

Thus

$$\mathbb{E}[\text{ReLU}(z)] = \frac{1}{\sqrt{2\pi}},$$

while

$$\mathbb{E}[\text{ReLU}(z')] = \frac{\sqrt{1 + \alpha^2 \theta}}{\sqrt{2\pi}}.$$

Since $\alpha > 0$ and $\theta > 0$, we have

$$\sqrt{1 + \alpha^2 \theta} > 1,$$

so the backdoored sample has larger expected ReLU activation.

(c) Since

$$\text{Var}(\text{ReLU}(Z)) \leq \mathbb{E}[\text{ReLU}(Z)^2],$$

and

$$\text{ReLU}(Z)^2 \leq Z^2,$$

we get

$$\text{Var}(\text{ReLU}(Z)) \leq \mathbb{E}[Z^2] = \sigma^2.$$

For

$$A_n := \frac{1}{n} \sum_{i=1}^n \text{ReLU}(Z_i),$$

we have

$$\mathbb{E}[A_n] = \frac{1}{n} \sum_{i=1}^n \mathbb{E}[\text{ReLU}(Z_i)] = \frac{\sigma}{\sqrt{2\pi}}.$$

Since the Z_i are independent,

$$\begin{aligned} \text{Var}(A_n) &= \text{Var}\left(\frac{1}{n} \sum_{i=1}^n \text{ReLU}(Z_i)\right) \\ &= \frac{1}{n^2} \sum_{i=1}^n \text{Var}(\text{ReLU}(Z_i)) \\ &\leq \frac{1}{n^2} \cdot n\sigma^2 = \frac{\sigma^2}{n}. \end{aligned}$$

(d) If $\tau \leq \mu_{\text{reg}}$, then the desired inequality is immediate. Assume therefore that $\tau > \mu_{\text{reg}}$. Then

$$\beta \leq \Pr[A_n \geq \tau].$$

Since $\tau > \mu_{\text{reg}}$,

$$\{A_n \geq \tau\} \subseteq \{|A_n - \mu_{\text{reg}}| \geq \tau - \mu_{\text{reg}}\}.$$

By Chebyshev's inequality,

$$\beta \leq \Pr[|A_n - \mu_{\text{reg}}| \geq \tau - \mu_{\text{reg}}] \leq \frac{\text{Var}(A_n)}{(\tau - \mu_{\text{reg}})^2}.$$

Using $\text{Var}(A_n) \leq 1/n$, we get

$$\beta \leq \frac{1}{n(\tau - \mu_{\text{reg}})^2}.$$

Rearranging gives

$$\tau - \mu_{\text{reg}} \leq \frac{1}{\sqrt{n\beta}},$$

and therefore

$$\tau \leq \mu_{\text{reg}} + \frac{1}{\sqrt{n\beta}}.$$

(e) From part (b),

$$\mu_{\text{bd}} = \frac{\sqrt{1 + \alpha^2 \theta}}{\sqrt{2\pi}}, \quad \mu_{\text{reg}} = \frac{1}{\sqrt{2\pi}}.$$

Hence

$$\mu_{\text{bd}} - \mu_{\text{reg}} = \frac{\sqrt{1 + \alpha^2 \theta} - 1}{\sqrt{2\pi}} > 0.$$

Using part (d),

$$\tau \leq \mu_{\text{reg}} + \frac{1}{\sqrt{n\beta}}.$$

Therefore

$$\Delta_n = \mu_{\text{bd}} - \tau \geq \mu_{\text{bd}} - \mu_{\text{reg}} - \frac{1}{\sqrt{n\beta}}.$$

For large enough n , the last term is smaller than $\mu_{\text{bd}} - \mu_{\text{reg}}$, so

$$\Delta_n > 0.$$

Now apply Chebyshev's inequality to A'_n . Since

$$\mathbb{E}[A'_n] = \mu_{\text{bd}},$$

we have

$$\begin{aligned} \Pr[A'_n < \tau] &= \Pr[\mu_{\text{bd}} - A'_n > \mu_{\text{bd}} - \tau] \\ &\leq \Pr[|A'_n - \mu_{\text{bd}}| \geq \Delta_n] \\ &\leq \frac{\text{Var}(A'_n)}{\Delta_n^2}. \end{aligned}$$

By part (c), with

$$\sigma^2 = 1 + \alpha^2 \theta,$$

we have

$$\text{Var}(A'_n) \leq \frac{1 + \alpha^2 \theta}{n}.$$

Thus

$$\Pr[A'_n < \tau] \leq \frac{1 + \alpha^2 \theta}{n \Delta_n^2}.$$

If Δ_n is bounded below by a positive constant, then the right-hand side is $O(1/n)$. Hence the fraction of backdoored samples classified as 0 decays like $1/n$.

16 Learn more

Cryptographic Backdoors:

- “Backdoor Defense, Learnability and Obfuscation” (Paul Christiano)
- “Injecting Undetectable Backdoors in Obfuscated Neural Networks and Language Models”