Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Rethinking Adversarial Examples

Master's Thesis

Yahya Jabary

yjabary@ethz.ch

Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Prof. Dr. Roger Wattenhofer
Prof. Dr. Schahram Dustdar

December 3, 2024

Acknowledgements

This thesis comes from working on a problem that truly matters to me, in an environment where curiosity and passion were shared, and I felt a sense of belonging. The topic of adversarial examples reflects my journey well, highlighting how subtle differences in perspective can lead to vastly different interpretations and outcomes.

I'm deeply grateful to those who supported me along the way. My parents, Shima and Florian, and my family, for their unwavering support, even when I took risks and turned down financial opportunities to pursue my passion. My partner, Laura, whose love and encouragement crossed the Atlantic and got me through many long nights.

I owe much to those who made this work possible. Prof. Wattenhofer, for trusting me with this project and guiding me with wisdom and humor. Andreas Plesner, who was just as much of a mentor as a collaborator, for his dedication to our vision. Turlan Kuzhagaliyev and Alireza Furutanpey, for their camaraderie.

Thanks also to those whose paths have diverged from mine but whose impact remains with me: Prof. Schahram Dustdar, who enabled my studies abroad, and Prof. Ali Mashtizadeh, who introduced me to operating systems research.

I hope to continue this journey with the same spirit that brought me here.

Abstract

...

Keywords: Robustness, Alignment, Interpretability in Machine Learning

Contents

Chapter 1

Introduction

We have two goals in writing this document. One: fulfilling the requirements for a master's degree by presenting and extending our original research [1] in thesis form. Two: offering a fresh and cohesive perspective on the rapidly evolving and, in our view, really exciting field of adversarial machine learning. To our knowledge, this is the first attempt to introduce this topic as a gateway for a broader audience, without any assumptions about prior knowledge. We hope it will be valuable to those interested.

1.1 Definition

Adversarial examples are closely related to the concept of perturbation methods.

The origin of perturbations can be traced back to the early days of computational geometry by Seidel et al. in 1998 [2]. Perturbation techniques in computational geometry address a fundamental challenge: handling "degeneracies" in geometric algorithms. These are special cases that occur when geometric primitives align in ways that break the general position assumptions the algorithms rely on.

> **Example:** Perturbation scheme for a linear classifier.
>
> Consider a simple case of determining whether a point lies above or below a line [3]. While this classification appears straightforward, numerical issues arise when the point lies exactly on the line. Such degeneracies can cascade into algorithm failures or inconsistent results. The elegant solution is to imagine slightly moving (perturbing) the geometric objects to eliminate these special cases. Formally, we can express symbolic perturbation as $p_\varepsilon(x) = x + \varepsilon \cdot \delta(x)$ where x is the original input, ε is an infinitesimally small positive number the exact value of which is unimportant, and $\delta(x)$ is the perturbation function to break degeneracies.

A perturbation scheme should be (1) consistent, meaning that the same input always produces the same perturbed output (2) infinitesimal, such that perturbations are small enough not to affect non-degenerate cases and (3) effective, in breaking all possible degeneracies.

One powerful perturbation approach is Simulation of Simplicity (SoS) [4, 5, 6, 7, 8, 9]. SoS systematically perturbs input coordinates using powers of a symbolic infinitesimal. For a point $p_i = (x_i, y_i)$, the perturbed coordinates become:

$$(\tilde{x_i}, \tilde{y_i}) = (x_i + \varepsilon^{2i}, y_i + \varepsilon^{2i+1}) = p_i + \varepsilon^{2i} \cdot (1, \varepsilon)$$

This scheme ensures that no two perturbed points share any coordinate, effectively eliminating collinearity and other degeneracies.

The beauty of perturbation methods lies in their ability to handle degeneracies without explicitly detecting them, making geometric algorithms both simpler and more robust.

Adversarial examples on the other hand, first introduced by Szegedy et al. in 2014 [10], follow the same principles as perturbation methods, but with the opposite objective. Instead of seeking to eliminate degeneracies (brittleness in the decision boundary), they exploit them to cause targeted misclassifications. Intuitively they can be understood as seeking the closest point in the input space that lies on the "wrong side" of a decision boundary relative to the original input. This shift, applied to the original input, creates an adversarial example.

Example: Fast Gradient Sign Method (FGSM)

FGSM is one of the earliest and most widely recognized adversarial attack techniques, introduced by Goodfellow et al. [11] in the context of visual recognition tasks. Given an input image x, FGSM generates an adversarial example x' by perturbing the input in the direction of the gradient of the loss function with respect to the input.

The perturbation is controlled by a parameter $\varepsilon > 0$ [a], which determines the magnitude of the change based on the direction of change for each pixel or feature in the input x. The model's loss function denoted by J, θ represents the model's parameters, and y is the true target label.

It works by calculating the gradient of the loss function with respect to the input, $\nabla_x J(\theta, x, y)$, and then adjusting the input in the direction of this gradient. The sign of the gradient, $\text{sign}(\nabla_x J(\theta, x, y))$, is used

to ensure that the perturbation is small, while the ℓ_∞-norm constraint ensures that the change to the input remains imperceptible to human observers [12].

The process for generating an adversarial example with FGSM can be expressed as:

$$x' = x + \underbrace{\varepsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))}_{\text{Perturbation}}$$

In the untargeted version, the perturbation is designed to increase the loss for the correct class. In the targeted version the perturbation is designed to minimize the loss with respect to the adversary's chosen target class, making the model predict it deliberately.

[a] Commonly $\varepsilon = 8/255$ for 8-bit images, so it stays within the precision constraints of the pixel values.

While initially discovered in computer vision applications, the principles behind these attacks remain consistent across all domains: Paper-reviewer assignment algorithms for academic conferences can be subverted, such that the authors are able to preselect reviewers who are likely to favor their work [13], despite challenges in the discrete and semantic nature of text [14]. Speech recognition systems are vulnerable to audio-based attacks, where carefully crafted noise can cause automatic speech recognition systems to fail [15]. Deep reinforcement learning applications, including pathfinding and robot control, have also shown susceptibility to adversarial manipulations that can compromise their decision-making capabilities [16].

Having established the general concept of adversarial examples, we can now explore the various dimensions among which they can be categorized. This is particularly important when building a threat model.

Adversarial examples in machine learning can be categorized along several key dimensions that reflect both their implementation approach and intended impact. One fundamental distinction lies between white-box and black-box attacks. White-box attacks assume complete knowledge of and access to the target model, while black-box attacks operate with limited or no access to the model's internal workings [17]. Interestingly, research has shown that in some cases, black-box attacks can be more effective than white-box approaches at compromising model security [17].

Another crucial categorization distinguishes between targeted and untargeted attacks. Targeted attacks aim to manipulate the model into producing a specific, predetermined output, whereas untargeted attacks simply seek to cause any misclassification or erroneous output [17, 18]. This distinction is particularly rel-

evant in security-critical applications, where the attacker's goals may vary from causing general disruption to achieving specific malicious outcomes.

The domain of application represents another important categorization axis. Adversarial attacks can be crafted for various types of data, including images, text, and graphs [18]. Each domain presents unique challenges and opportunities for attackers, requiring different technical approaches and defensive strategies.

The sophistication and complexity of attacks form another categorical dimension. Some attacks employ simple greedy algorithms with spatial heuristics, while others utilize advanced gradient-guided strategies borrowed from image classification domains [17]. The choice of attack strategy often depends on the attacker's resources, knowledge, and objectives.

The semantic nature of adversarial examples provides yet another classification framework. Some attacks focus on manipulating the semantic meaning of inputs, while others exploit the mathematical properties of neural networks without regard for semantic interpretation [19]. This distinction has important implications for both the effectiveness of attacks and the development of defensive measures.

In the context of system security, adversarial attacks can also be categorized based on their timing and persistence. Some attacks target the training phase of machine learning models, while others focus on the inference phase [18]. This temporal dimension is particularly relevant for developing comprehensive defense strategies that protect systems throughout their lifecycle.

These various categorization schemes help researchers and practitioners better understand the landscape of adversarial attacks, enabling more effective defense mechanisms and contributing to the development of more robust machine learning systems. The field continues to evolve, with new attack vectors and categories emerging as artificial intelligence applications become more prevalent in security-critical domains [20].

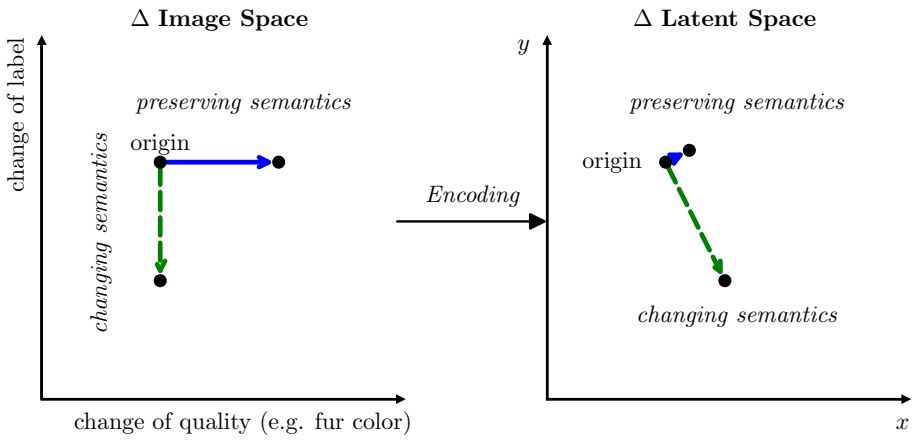

Figure 1.1: Semantics Preserving vs. Semantics Changing Perturbations in the Latent Space.

example with email spam

1.2 Counterintuitive Properties

Despite how important and useful they are, there's still a bunch we don't really know about them.

There are a bunch of stuff we don't know about them

1.3 Mental Models

In 2013 Szegedy et al. [21] discovered that deep neural networks are vulnerable to adversarial examples – inputs with imperceptible perturbations that cause misclassification, revealing blind spots where the models' decision boundaries are brittle, despite appearing to generalize well on normal inputs. Since then we have made a lot of progress in both finding attack vectors [11] [22] [23] and adversarially training models to be more robust [24] [22] [25] to these attacks.

However, despite the progress, we still don't fully understand:

- What they are, why they exist and how they work.
- How to defend against them without sacrificing accuracy (robustness vs accuracy trade-off).

- Why they transfer between different models, datasets, architectures, training procedures and even to the human visual system [26].

There have been many attempts at explaining adversarial examples, each with limitations and assumptions – some complementary, some contradictory [1].

For example, the *dimpled manifold hypothesis* [28] suggests that the decision boundary of deep neural networks is close to the data manifold, making it easy to find adversarial examples, while the *non-robust features hypothesis* [27] suggests that models exploit non-robust features that are imperceptible to humans, leading to a vulnerability against small perturbations

The *dimpled manifold hypothesis* is a particularly controversial one, as it was criticized by Yannik Kilcher [29] in 2021, who also provided a counterexample in less than 100 lines of code [30]. Despite the lack of generalizability, a master's student from the University of Vienna, Lukas Karner, successfully verified and replicated all experiments detailed in the dimples paper [31] in 2023. Lukas allowed me to use his results in my work. He mentioned that there is currently no paper or thesis based on his work and that he would be happy to see his results being used in a meaningful way.

This leaves us with the possibility that the experiments carried out are correct in themselves, but that the chain of reasoning is inconclusive and therefore doesn't generalize. Investigating this further would require more rigor by formalizing falsifiable hypotheses based on the paper and conducting experiments to test them.

Another aspect of the *dimpled manifold hypothesis* worth exploring is the idea of projecting the perturbations on the data manifold before applying them to the input space. Reducing the dimensionality of the perturbations or compressing them makes them visible to the human eye and interpretable as demonstrated by Karner [31].

1.4 Counterintuitive Properties

[1] For a comprehensive overview of the hypotheses, see the Addendum of Ilyas et al. [27]

Chapter 2

Methodology

Chapter 3

Results

Chapter 4

Stuff

Figure 4.1: This is an example graphic.

> "The stuff is what the stuff is, brother. Okay. We don't ask questions about the weights. We just wake up, we go to work, we use the weights, we go back home. Okay. If we change the weights, the predictions would be different and less good, probably... depending on the weather... so we don't ask about the weights."
>
> — *James Mickens, USENIX Security 18 [32]*

4.1 First Section Title

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

4.1.1 First Subsection Title

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Theorem 4.1 (First Theorem)**.** *This is our first theorem.*

Proof. And this is the proof of the first theorem with a complicated formula and a reference to Theorem 4.1. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

$$\frac{\mathrm{d}}{\mathrm{d}x}\arctan(\sin(x^2)) = -2 \cdot \frac{\cos(x^2)x}{-2 + (\cos(x^2))^2} \tag{4.1}$$

□

And here we cite some external documents [33, 34]. An example of an included graphic can be found in Figure 4.1. Note that in LaTeX, "quotes" do not use the usual double quote characters.

Bibliography

[1] Y. Jabary, A. Plesner, T. Kuzhagaliyev, and R. Wattenhofer, "Seeing through the mask: Rethinking adversarial examples for captchas," *arXiv preprint arXiv:2409.05558*, 2024.

[2] R. Seidel, "The nature and meaning of perturbations in geometric computing," *Discrete & Computational Geometry*, vol. 19, pp. 1–17, 1998.

[3] M. De Berg, *Computational geometry: algorithms and applications*. Springer Science & Business Media, 2000.

[4] W. R. Franklin and S. V. G. de Magalhães, "Implementing simulation of simplicity for geometric degeneracies," *arXiv preprint arXiv:2212.08226*, 2022.

[5] Edelsbrunner, Letscher, and Zomorodian, "Topological persistence and simplification," *Discrete & computational geometry*, vol. 28, pp. 511–533, 2002.

[6] H. Edelsbrunner and D. Guoy, "Sink-insertion for mesh improvement," in *Proceedings of the seventeenth annual symposium on Computational geometry*, 2001, pp. 115–123.

[7] H. Edelsbrunner and E. P. Mücke, "Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms," *ACM Transactions on Graphics (tog)*, vol. 9, no. 1, pp. 66–104, 1990.

[8] B. Lévy, "Robustness and efficiency of geometric programs: The predicate construction kit (pck)," *Computer-Aided Design*, vol. 72, pp. 3–12, 2016.

[9] P. Schorn, "An axiomatic approach to robust geometric programs," *Journal of symbolic computation*, vol. 16, no. 2, pp. 155–165, 1993.

[10] C. Szegedy, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[12] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2578–2593, 2019.

[13] T. Eisenhofer, E. Quiring, J. Möller, D. Riepel, T. Holz, and K. Rieck, "No more reviewer# 2: Subverting automatic {Paper-Reviewer} assignment using adversarial learning," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 5109–5126.

[14] X. Han, Y. Zhang, W. Wang, and B. Wang, "Text adversarial attacks and defenses: Issues, taxonomy, and perspectives," *Security and Communication Networks*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248369346

[15] K. Rajaratnam and J. Kalita, "Noise flooding for detecting audio adversarial examples against automatic speech recognition," in *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2018, pp. 197–201.

[16] X. Bai, W. Niu, J. Liu, X. Gao, Y. Xiang, and J. Liu, "Adversarial examples construction towards white-box q table variation in dqn pathfinding training," *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 781–787, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49895854

[17] G. Capozzi, D. C. D'Elia, G. A. Di Luna, and L. Querzoni, "Adversarial attacks against binary similarity systems," *IEEE Access*, 2024.

[18] S. Kashyap, A. Sharma, S. Gautam, R. Sharma, S. Chauhan, and Simran, "Adversarial attacks and defenses in deep learning," *2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP)*, pp. 318–323, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272716335

[19] K. Browne and B. Swift, "Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks," *arXiv preprint arXiv:2012.10076*, 2020.

[20] Y. L. Khaleel, M. A. Habeeb, and H. Alnabulsi, "Adversarial attacks in machine learning: Key insights and defense approaches," *Applied Data Science and Analysis*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:272000855

[21] E. D. Cubuk, B. Zoph, S. S. Schoenholz, and Q. V. Le, "Intriguing properties of adversarial examples," *arXiv preprint arXiv:1711.02846*, 2017.

[22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[23] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[24] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in neural information processing systems*, vol. 32, 2019.

[25] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[26] G. Elsayed, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial examples that fool both computer vision and time-limited humans," *Advances in neural information processing systems*, vol. 31, 2018.

[27] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.

[28] A. Shamir, O. Melamed, and O. BenShmuel, "The dimpled manifold model of adversarial examples in machine learning," *arXiv preprint arXiv:2106.10151*, 2021.

[29] Y. Kilcher, "The Dimpled Manifold Model of Adversarial Examples in Machine Learning (Research Paper Explained)," https://www.youtube.com/watch?v=k_hUdZJNzkU/, 2021, [Online; accessed 17-July-2024].

[30] Y. Kilcher, "dimple test," https://gist.github.com/yk/de8d987c4eb6a39b6d9c08f0744b1f64/, 2021, [Online; accessed 17-July-2024].

[31] L. Karner, "The Dimpled Manifold Revisited," https://github.com/LukasKarner/dimpled-manifolds/, 2023, [Online; accessed 17-July-2024].

[32] J. Mickens, "Q: Why do keynote speakers keep suggesting that improving security is possible? a: Because keynote speakers make bad life decisions and are poor role models," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/mickens

[33] A. One and A. Two, "A theoretical work on computer science," in *30th Symposium on Comparative Irrelevance, Somewhere, Some Country*, Jun. 1999.

[34] A. One and A. Two, "A theoretical work on computer science," in *30th Symposium on Comparative Irrelevance, Somewhere, Some Country*, Jun. 1999.

Appendix A

First Appendix Chapter Title
