Adverserial Attacks and Adverserial Training

Ruthwik Ganesh
230702930
ec23759@qmul.ac.uk

***Abstract*—Adversarial examples are inputs to machine learning models that have been intentionally designed to trick the model into producing incorrect outputs. This poses a real-world security threat, as adversarial examples can be used to compromise systems built on machine learning. Defenses against adversarial examples are an open research problem. Despite the potential dangers of adversarial examples, they can also be used to improve other machine learning algorithms. This survey provides an overview of adversarial examples and adversarial training, explaining why adversarial examples exist and how they can be used to improve machine learning algorithms and its implications on the society at large.**

I. Introduction

Deep neural networks (DNNs) have emerged as formidable tools in the realm of computer vision, offering unparalleled capabilities for the analysis and interpretation of visual data. This ascendancy is underpinned by several key attributes intrinsic to DNN architectures, which collectively contribute to their efficacy in solving intricate tasks in the visual domain. From a hierarchical perspective, DNNs excel in the automatic extraction of features, progressively constructing intricate representations through multiple layers. Their capacity for end-to-end learning eliminates the need for manual feature engineering, facilitating the direct extraction of pertinent features from raw data. Moreover, the introduction of non-linearity through activation functions empowers these networks to discern and model complex relationships, a crucial characteristic when handling the intricacies inherent in visual data.

However, Adversarial examples [1] exploit the vulnerability of neural networks to imperceptible perturbations, leading to misclassifications or erroneous outputs. This phenomenon underscores the need for ongoing research in enhancing the robustness and interpretability of deep learning models in real-world applications.

In this review we'll understand adversarial examples, an overview of their nature and characteristics. Subsequently, an exploration into the causal factors underlying the existence of adversarial examples, addressing the question of why such instances manifest within deep learning models.

Furthermore, this discourse will outline the current state of research in adversarial defence mechanisms, delineating the existing strategies employed to mitigate the risks associated with adversarial attacks.

II. Intuition

The initial conceptualization posited adversarial examples as a manifestation of overfitting [2], wherein the complicated architecture of deep neural networks learns to accommodate the nuances of the training set, resulting in an ambiguous behaviour on the test set and, consequently, facilitating random errors that an attacker can exploit.

To provide a more concrete illustration of this narrative, consider a training set comprising three instances of blue X's and three instances of green O's *(Fig.1)*. Employing a complex classifier capable of adeptly fitting the training set, the model generates blue blobs encapsulating the regions it deems indicative of X's and green masses delineating the areas corresponding to O's. However, due to the model's excessive parameterization beyond the requisites of the training task, it disperses random probability masses across the remaining space. This randomness gives rise to instances where the model, despite proximity to the training set instances, misclassifies adversarial examples, as evidenced by a red X within the vicinity of green space and a red O amidst blue mass.

In contrast to the previous demonstration involving a high-capacity, non-linear model, we opt for a linear model. The resulting linear model exhibits a discernible hyperplane dividing the two classes *(Fig.2)*. However, it becomes evident that this hyperplane inadequately captures the true spatial arrangement of the classes. The O's, arranged in a distinctive C-shaped manifold, defy linear representation, leading to misclassifications near the decision boundary. Traversing beyond the extent of the O's results in crossing the decision boundary, where a red O is erroneously assigned, even when near the decision boundary and adjacent to other O's. Similarly, moving from locations proximal to X's, just beyond the decision boundary, lead to misclassifications as O's. Notably, the plot unveils an unusual characteristic wherein the lower left and upper right corners, devoid of any observed data, receive confident classifications as X's and O's, respectively.

If overfitting were the predominant factor, each adversarial example would be construed because of random fluctuations and unique to the specific model. However, contrary to this expectation, further investigations reveal a surprising consistency across different models, wherein the same adversarial examples are consistently misclassified, signifying a systematic rather than a random effect. Moreover, the observation that adding the offset vector derived from the difference between an original example and an adversarial example to a testing examples, consistently results in adversarial instances suggests a discernible systematic influence in the model's behaviour.

This demonstration underscores the limitations of linear models in capturing the intricate and non-linear structure of certain datasets, rendering them susceptible to misclassifications, especially near decision boundaries and in regions where no data points are observed. This systematic tendencies of linear models to exhibit overconfidence in distant and unexplored regions contribute to a nuanced understanding of their vulnerabilities, shedding light on potential avenues for adversarial attacks stemming from the constraints imposed by the linear model family. Considering these findings, suggests that adversarial examples might be more aptly characterized because of underfitting rather than overfitting. Thus prompts a critical inquiry: do the characteristics of deep neural networks bear any resemblance to those of linear models, and can linear models offer insights into the failure modes of deep neural networks?

III. Cause Of Linearity in Deep Neural Networks

Modern deep neural networks are revealed to possess a distinct structure characterized by a piecewise linear nature, diverging from the traditional conception of a singular linear function. Particularly, networks utilizing rectified linear units (ReLUs) manifest as piecewise linear functions, composed of a limited number of linear segments. The mapping from input images to output logits, where logits denote unnormalized log probabilities prior to the SoftMax operation, is inherently piecewise linear in such networks. Notably, alternative neural network architectures, such as MaxOut networks, also adhere to a piecewise linear function *(Fig.3)*, while others approximate this characteristic closely. It is noteworthy that preceding the widespread adoption of rectified linear units, sigmoid units, either logistic sigmoid or hyperbolic tangent units, were predominantly employed. The effective deployment of these sigmoidal units required meticulous tuning, particularly during initialization, to ensure proximity to the Sigmond's linear approximations for a significant portion of the activation range. Even in long short-term memory (LSTM), a prominent recurrent network architecture, relies on addition operations across consecutive time steps for information accumulation and retention. The inherent simplicity of addition imparts a highly linear character to the interaction between distant time steps within an LSTM.

However, it is imperative to clarify that the discussions thus far pertain to the mapping from the input of the model to its output. This mapping is identified as either linear or piecewise linear, characterized by relatively few components. In contrast, the mapping from the parameters of the network to its output is fundamentally non-linear due to the multiplication of weight matrices across each layer. This non-linearity introduces considerable complexity into the training process, rendering the optimization of parameters a challenging endeavour. Conversely, the mapping from input to output remains more linear and predictable, thereby facilitating optimization problems directed at the input rather than the parameters. In practical terms, this observation is supported by tracing a one-dimensional path through the input space of a convolutional network, exemplifying the tractability of optimization problems that focus on input optimization as opposed to parameter optimization in the context of deep neural networks.

IV. Generating Adversarial Examples

In the context of constructing adversarial examples, it is essential to acknowledge the capacity to introduce substantial perturbations while maintaining minimal perceptual changes from a human perspective. The presented illustration involves a handwritten digit three subjected to various transformations, each possessing an identical L2 norm perturbation *(Fig.4.)*. The top row showcases a transformation where the digit three is modified into a seven by identifying the nearest seven in the training set. The resulting perturbation, represented by white and black pixels denoting addition and subtraction, respectively, yields an L2 norm of 3.96. This value provides a reference for the magnitude of perturbations attainable within this framework. The middle row demonstrates the application of a perturbation of equivalent size but with a randomly chosen direction. Despite the perturbation, the class of the digit three remains unaltered, resulting in random noise that is still easily recognizable as the digit three. In contrast, the bottom row showcases a scenario where a portion of the digit three is erased through a perturbation of comparable norm, resulting in an input devoid of any class.

It is crucial to note that adversarial changes of varying nature can occur with the same L2 norm perturbation. In many adversarial scenarios, perturbations with even larger L2 norms are often employed. The rationale behind this phenomenon lies in the presence of multiple pixels in the image, where slight modifications to individual pixels can accumulate into relatively extensive vectors, particularly in larger datasets such as ImageNet.

To mitigate the potential exploitation of perturbations that merely change the input class without inducing true misclassification, adversarial example procedures often incorporate a MaxNorm constraint. This constraint ensures that no pixel undergoes changes beyond a specified threshold (epsilon). Although the L2 norm may be extensive, the constraint prevents concentrated changes that might erase significant portions of the input, thereby preserving the integrity of the adversarial attack.

One expeditious method for adversarial example generation involves computing the gradient of the cost used to train the network with respect to the input and subsequently taking the sign of that gradient. This approach, known as the fast gradient sign method, effectively enforces the MaxNorm constraint by limiting changes to the input by up to epsilon at each pixel. This method leverages the linearity assumption inherent in neural network models, using a first-order Taylor series approximation to maximize the cost within the constraints imposed.

Fast Gradient Sign Method

For a given example x, generate adversarial example X as follows:

$$X = x + \text{epsilon} * \text{sign}\,(\text{grad}\,(J\,(x, y), x))$$

grad() is the gradient function. sign() returns the sign of the input. epsilon is the amount of perturbation. J is the cost function. y is the ground truth for x.

This is a one-shot method to generate an adversarial example. Value of epsilon is a trade-off between similarity to x and success of an adversarial attack using X.

Basic Iterative Method

We extend the idea of the above method and iteratively perform the update for X and clip the output pixel-wise to keep the final X within the epsilon neighborhood.

$$X = \text{Clip-epsilon}\,(x + \text{sign}\,(\text{grad}\,(J\,(x, y), x)))$$

Number of iterations is a hyper-parameter, that trades off between computational speed and a successful attack.

Targeted Fast Gradient Sign Method

We choose a y-target and get a clean image x from which we would like to generate an adversarial example, X. Ideally, this image belongs to a class that is perceptually close to y-target but belongs to a different class.

X = Clip-epsilon (x - sign (grad (J (x, y-target), x)))

We now "descend" the gradient as indicated by the negative sign to get closer to the y-target. This method can be extended as an iterative process as well.

While the fast gradient sign method offers a swift means of generating adversarial examples, it is advisable to complement it with other methods, such as Nicholas Carlini's attack based on multiple steps of the Adam optimizer, for a comprehensive evaluation of model vulnerabilities [5]. The effectiveness of defensive measures should be gauged against more computationally intensive attacks, ensuring a robust assessment of model security beyond the immediate efficacy of expedited techniques.

The above methods being the most prevalent to fool a model, it's also worth mentioning *(Table.1)* Jacobian-based Saliency Map Attack, One Pixel Attack, Deepfool, Universal Adversarial Perturbations which have high transferability [5].

V. Transferability of adversarial examples

In scenarios where an adversary seeks to deceive a model without direct access to its architecture, algorithm, or parameters, several strategies are employed. The attacker may be unaware of whether they are targeting a decision tree or a deep neural network, further complicating their task. To overcome these limitations, the attacker can construct their own model for the purpose of launching adversarial attacks.

Two primary methodologies are identified for training the adversary's model. Firstly, the attacker may label their own training set for the specific task they intend to compromise. For instance, if the target model is an ImageNet classifier and the attacker lacks access to weights of ImageNet model, they can create and label their own set of images, subsequently training an object recognizer. This approach allows the attacker to generate adversarial examples likely to influence the target ImageNet model.

Alternatively, in situations where creating an independent training set is impractical, the attacker can leverage limited access to the target model. By submitting inputs to the model and observing corresponding outputs, the attacker can utilize these output-input pairs as a surrogate training set. Remarkably, even if the observer only receives class labels from the target model, this information is deemed sufficient for training the adversarial model.

Following the acquisition of a suitable training set through either of the methods, the adversary trains their own model and generates adversarial examples specifically tailored to impact the target model. Subsequently, these crafted adversarial examples are deployed to deceive the target model, illustrating that adversarial attacks can be executed successfully without direct access to the target model during the training phase.

Transferability, or the extent to which adversarial examples generated for one model affect another, has been empirically investigated across different datasets. Most models exhibit an intermediate level of transferability, ranging from 60% to 80%. However, certain models, such as Support Vector Machines (SVMs), display higher data dependence due to their focus on a small subset of training data to formulate decision boundaries.

VI. Defence Against Adversarial Attacks

The ongoing investigation in this research domain has, to date, yielded only a collection of best practices aimed at mitigating the vulnerabilities of models to adversarial attacks. Notably, Nicolas Carlini et al. have conducted a meticulous examination of this domain and have proposed several recommendations in their scholarly work [3], a subset of which is delineated below.

Gradient masking: A Closer Examination

A widely considered defensive measure involves hiding model gradients to impede the computation of adversarial examples. However, a critical analysis of this approach reveals that concealing gradients does not inherently enhance model robustness. Rather, it introduces an added layer of complexity to adversarial attacks, posing challenges such as the potential utilization of substitution models or the random guessing of adversarial points. Empirical evidence from experiments underscores that models trained on similar tasks remain vulnerable to analogous adversarial examples despite gradient concealment.

Adversarial Training: Proactive Defence Strategies

An alternative approach involves the integration of adversarial examples into the training dataset, a technique known as adversarial training. This strategy has demonstrated improved classification accuracies on adversarial examples, contributing to enhanced model robustness. While acknowledging vulnerability to black-box attacks, adversarial training induces a regularization effect, rendering learned functions resistant to local perturbations and, consequently, adversarial attacks.

Acknowledging Ignorance: The "Don't Know/Null" Class

The proposition of assigning a "don't know/null" class to images *(Fig.5)* unfamiliar to the model represents a novel defensive strategy. This class is envisioned to be predicted for adversarial examples, disrupting the transferability property of such perturbations. The practical implementation of this strategy, however, introduces challenges concerning data acquisition for this specialized class. Questions regarding the quantity of data required, optimal gathering methodologies, and the overall feasibility of this approach underscore the intricacies involved in operationalizing the "don't know/null" class.

Among the array of strategies, Adversarial Training has demonstrated notable efficacy in producing improved outcomes. The safeguarding measures employed to protect models necessitate a high degree of specificity, and further details regarding these protective measures can be gleaned from the article [3] authored by Nicolas Carlini et al. and the more recent survey [4] by Yulong Wang et. al.

VII. Discussion

The evident reality of the threat posed by adversarial attacks on deep learning is established through a comprehensive review of the literature. While some perspectives downplay the severity of this concern, a substantial body of evidence, particularly from Sections III and IV, demonstrates the severe degradation of deep learning performance across multiple Computer Vision tasks. Significantly, the vulnerability extends into the real physical world, unequivocally affirming that adversarial attacks present a tangible threat to the practical application of deep learning.

The vulnerability to adversarial attacks is identified as a pervasive and general phenomenon affecting various deep neural network architectures (MLPs, CNNs, RNNs) and a wide spectrum of tasks in Computer Vision, including recognition, segmentation, and detection. Although the existing body of work predominantly focuses on classification/recognition tasks, the literature survey underscores that deep learning approaches are susceptible to adversarial attacks across diverse contexts.

The common property of adversarial examples to generalize effectively between different neural networks, especially those sharing similar architectures, is highlighted. This characteristic, often exploited in black-box attacks, adds a layer of complexity to the understanding and mitigation of adversarial threats.

The reasons behind the vulnerability of deep neural networks to adversarial perturbations emerge as a subject requiring further investigation. Diverse viewpoints in the literature underscore the need for a systematic exploration of the underlying causes, emphasizing the complexity of the issue.

The widely debated notion that the linearity of modern deep neural networks contributes to their vulnerability is acknowledged. While facing opposition, multiple independent contributions in the survey support the idea that linearity promotes vulnerability. However, it is recognized that linearity alone may not be the exclusive factor behind successful adversarial attacks using inexpensive analytical perturbations.

The observation that counter-counter measures are viable introduces a challenge to existing defence techniques. This highlights the importance of new defences not only countering known attacks but also providing robustness estimates against potential counter strategies.

The research direction in adversarial attacks and defences is characterized as highly active and dynamic. Most of the surveyed literature has emerged in the last two years, indicating a continuous influx of contributions. This dynamic landscape involves proposing defensive techniques against known attacks while simultaneously witnessing the development of more potent adversarial attacks. The active engagement is exemplified by events like the Kaggle competition on defence against adversarial attacks. The overarching hope is that this heightened research activity will eventually lead to the development of robust deep learning approaches suitable for safety and security-critical applications in the real world.

VIII. Conclusion

Deep neural networks, despite their impressive performance in various computer vision tasks, are susceptible to minor perturbations in input that can drastically alter their outcomes. This vulnerability is significant given deep learning's pivotal role in the ongoing advancements in machine learning and artificial intelligence. Consequently, there has been a surge in research focused on creating and countering adversarial attacks in deep learning. This article provides an overview of these efforts, with an emphasis on the most impactful and intriguing studies. The review reveals that adversarial attacks pose a genuine risk to the practical application of deep learning, particularly in areas where safety and security are paramount. It is evident from the literature that deep learning systems are not just vulnerable to digital attacks, but also to those in the physical world. Nevertheless, the intense research activity in this field gives hope that deep learning may eventually exhibit substantial resilience to these adversarial attacks in the future.

References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli., "Evasion attacks against machine learning at test time." In Joint European conference on machine learning and knowledge discovery in databases,pp. 387–402. Springer, 2013.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow., "Intriguing properties of neural networks." ICLR, abs/1312.6199, 2014b.

[3] N. Carlini, and A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Mądry, A. Kurakin "On Evaluating Adverserial Robustness," in Machine Learning (cs.LG); Cryptography and Security (cs.CR); Machine Learning (stat.ML), arXiv:1902.06705 [cs.LG].

[4] Y. Wang, T. Sun, S. Li, X. Yuan, W. Ni, E. Hossain, H. Vincent, "Adversarial Attacks and Defenses in Machine Learning-Powered Networks: A Contemporary Survey," in Machine Learning (cs.LG); Artificial Intelligence (cs.AI), arXiv:2303.06302 [cs.LG].

[5] N. Akhtar, A. Mian, "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey," in Computer Vision and Pattern Recognition (cs.CV), arXiv:1801.00553 [cs.CV].