

Adversarial Attacks in Machine Learning

Colton Gyulay*

Abstract

In recent years a confluence of large datasets, cheap, parallelized computing power, and advances in statistical learning approaches—namely deep learning—has led to the widespread use of machine learning (ML) in everyday applications. ML models have found practical use in a variety of settings, from computer vision to recommender systems to natural language processing. Despite their widespread employment, the exact nature of more complex models as well as the details of their decision-making processes elude the understanding of much of the technical community. Such systems contain nebulous vulnerabilities that need to be better understood and guarded against, especially in critical applications like autonomous vehicle navigation. Recent research has elucidated some of these threats against ML systems, known as “adversarial attacks,” and has attempted to describe mechanisms for both attack and defense. In this paper we outline current research, demonstrate concrete examples of adversarial attacks and compare different methods of generating adversarial examples, and ultimately discuss the ethical implications of such vulnerabilities in ML systems. We conclude that certain defensive measures, namely adversarial training, should be employed when creating production ready ML models.

INTRODUCTION

Recent progress in ML and deep learning has led to the development of highly effective models used in image classification, machine translation, game playing, and many other practical problem domains. (Krizhevsky et al., 2012; Bahdanau et al., 2014; Mnih et al., 2015). Though these models demonstrate considerable performance in classification tasks, they are susceptible to adversarial inputs which confound models and lead to inaccurate predictions. Neural networks and similar classes of ML models seem to exhibit particular vulnerability to these adversarial examples, which, concerningly, can be constructed with perturbations subtle enough that they may be completely imperceptible to humans.

Though adversarial examples can take on many forms depending on the classification system, for the purposes of this paper we focus on image classification systems and the generation of adversarial images (think optical illusions for computer vision models). To formalize the problem of adversarial examples, we follow the lead of Kurakin et al. (2016). Consider an ML model M that takes an input X and generates a correct class prediction y_{true} : $M(X) = y_{true}$. It is possible to generate an adversarial input A that is nearly indistinguishable from X , but yields an incorrect class prediction: $M(A) \neq y_{true}$. Though A may be generated with the addition of only a small amount of noise to X , the model may be highly confident in its incorrect class prediction.

As ML models have become more ubiquitous in their application, understanding their vulnerabilities and seeking to mitigate them is increasingly necessitated. In certain critical problem domains such as health care or autonomous vehicle navigation, it is completely conceivable that such models might make ethically challenging decisions that directly affect human lives. These domains highlight the importance of developing a deeper understanding of how models generate

*cgyulay@college.harvard.edu

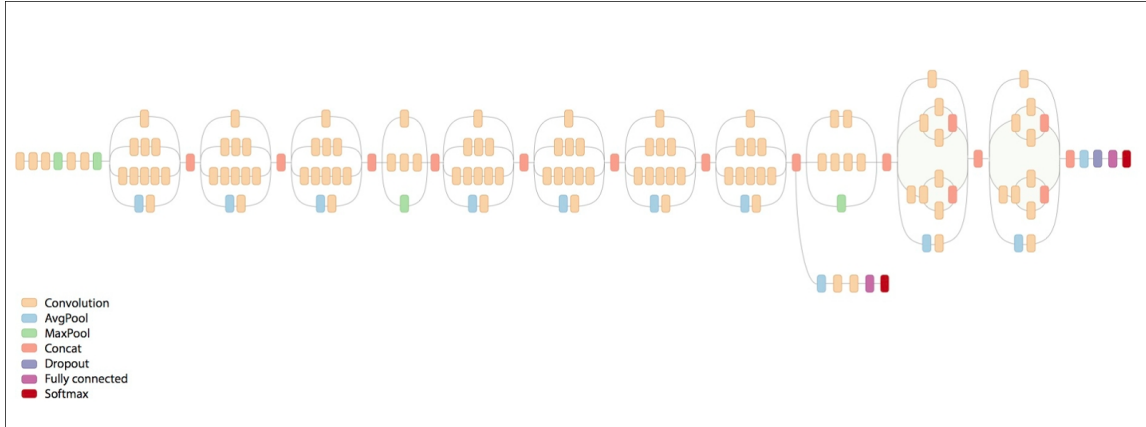


Figure 1: *Inception v3 model architecture from Google.*

decisions and what their shortcomings may be. With this in mind, exploration of both adversarial attacks and proposed defenses is unavoidable if we are to employ ML models confidently in ethically fraught domains.

To empirically and qualitatively investigate the properties of adversarial examples, we generate adversarial images against the pre-trained ImageNet Inception v3 system (see **Figure 1**), a state-of-the-art convolutional neural network model from Google (Szegedy et al., 2015). Using a pre-trained model facilitates faster experimentation and provides us with a classification system that outperforms any model we could train ourselves on a reasonable timeline. We experiment with limiting perturbation magnitude and learning rate in multiple kinds of attacks against Inception v3. Though our experiments focus on a setting in which the attacker has full access to a model’s parameters when generating adversarial examples, other work has demonstrated both the transferability of adversarial examples between models as well as techniques that can be executed against black box systems (Papernot et al., 2016a,b; Evtimov et al., 2017). The property of transferability renders the conclusions of our specific research setting applicable to other black box models. After discussing the nature of these adversarial examples, we delve into possible defenses and ethical concerns surrounding adversarial examples in the context of AI safety.

The paper proceeds as follows: Section one describes related work and the current research environment. Section two formalizes the adversarial attack problem, classifies domains in which attacks may occur, and outlines methods used for generating adversarial examples. Section three presents experimental results surrounding generated examples. Section four introduces defense mechanisms and describes the implications of these adversarial examples as they relate to AI safety and security. Section five concludes the study and presents various directions for future work.

I. RELATED WORK

The examination of adversarial attacks against AI classifiers began over a decade ago with naive Bayes models (Dalvi et al., 2004). It wasn’t until the proliferation of neural networks and deep learning, however, that adversarial examples caught the attention of the technical community. As these deeper, more complex models began to enter the mainstream, the study of their vulnerabilities accelerated. Szegedy et al. (2013) was the first work to identify the seemingly widespread susceptibility of several state-of-the-art models to adversarial examples, and began

a discussion into the fragile nature of the decision boundaries exhibited by these supposedly high-performing systems. Goodfellow et al. (2014) expanded on this research and developed the "fast gradient sign method" for quickly generating adversarial examples given full knowledge of a model's implementation and parameters.

The property of transferability between models was then examined by Papernot et al. (2016a) which opened the door to the study of adversarial attacks against black box systems in which only outputs, not parameters, were available. Kurakin et al. (2016) illustrated attacks in fully black box settings where models were hosted by third parties and began the transition of research into more real world domains. Papernot et al. (2016b) demonstrated the robustness of adversarial examples in real world applications by feeding physical examples through a cell phone camera before classification. This work also introduced an effective iterative approach to generating both targeted and non-targeted adversarial examples. Evtimov et al. (2017) established a new general attack algorithm called "Robust Physical Perturbations" that they used to modify street signs which fooled a classifier from multiple angles and distances. This represents the current state-of-the-art in adversarial attack vectors, and its robustness certainly raises concerns about models employed in the field today and going forward.

On the defensive side, aforementioned work from Papernot et al. (2016b) explored the idea of using adversarial examples during training to improve resilience against these types of attacks. They also demonstrated the added effects of adversarial training in network regularization. Gradient masking techniques and defensive distillation, which focus on hiding information from attackers through deployment considerations, were formalized by Papernot et al. (2015). These techniques make attacks more difficult, but models will continue to be vulnerable against the same types of attacks when executed with greater computing power. Finally, Papernot et al. (2016c) established an overview of the defensive landscape, discussed the trade-offs between model accuracy and resilience, and began to situate these types of attacks within the AI safety discourse. That said, there is still a dearth of research in the spheres of defense and ethics in the adversarial attack space.

II. ADVERSARIAL ATTACKS

When reviewing adversarial attacks in ML, there are multiple attack environments and vectors that must be considered. As stated, discussion in this paper will focus on attacks in the computer vision domain. In this section, we provide an overview of adversarial attack environments and vectors, then we introduce mathematical formalizations for different adversarial methods, and finally we evaluate each adversarial method and examples of their generated attacks.

i. Attack Environments

There are two primary situations in regard to model information availability: "full knowledge," where a system's architecture and parameters are accessible, and "black box," where only network outputs are accessible. The full knowledge environment is by far the most dangerous, but it is also the least likely attack vector in deployment situations with effective security practices. In this setting, an attacker knows the architecture of the underlying system and has access to the model's parameters. Parameter access allows the construction of the error gradient which can be used to directly generate adversarial examples. These examples prey upon the weakest sections of the decision manifolds and can be constructed with the least amount of noise and visual perturbation. These methods are explored in the following subsection.

The more likely environment in which attacks are to occur is the black box setting. In this

situation, the model and its parameters are hidden, but its outputs are available. For example, an attacker might be able to upload an image to a computer vision system that returns information about the model’s class predictions and corresponding likelihoods. Though no error gradient is available, iterative probing of the network using adversarial examples can lend directional insight into the underlying decision boundaries. As demonstrated by Papernot et al. (2016a), the property of transferability also allows for the training of a similar "surrogate" model to the target model which can be used to estimate the target gradient and speed up adversarial attack generation.

Aside from considerations regarding information availability, there are also two primary types of input environments: software, where information is passed directly to a system (e.g. a picture is uploaded to a publicly available API), and physical, where a system processes information from the real world (e.g. a stop sign has been modified with adversarial stickers to mislead an autonomous vehicle). The primary distinction between these cases is environmental sterility. In the real world scenario, the attack must be physically manufactured and placed in the vicinity of the system. In the context of computer vision, such an attack must be robust across multiple viewing angles and distances. No such difficulties apply to the software scenario. That said, real world attacks are certainly feasible and improving in effectiveness (Papernot et al., 2016b; Evtimov et al., 2017).

Finally, there are two principal attack types that can be executed against multinomial classifiers: non-targeted and targeted. In non-targeted attacks, the attacker seeks to non-directionally reduce the probability of a model producing the correct class output. An example of such an attack can be seen in **Figure 3b**. The only objective of this type of attack is to reduce the probability of the current class, which in turn increases the probability of alternative classes randomly. In models where there are many potential output classes, such as in Inception v3 which has 1000 possible output predictions, non-targeted attacks tend to be less interesting (Szegedy et al., 2015). Due to the similar nature of certain classes, a non-targeted attack may lead to an image of a dog being misclassified as another similar breed of dog (Kurakin et al., 2016). This phenomenon led to the development of the more focused, and perhaps more nefarious, targeted attack. In this setting, a specific alternate class is selected for which to optimize prediction probability. An example of this attack carried out can be seen in **Figures 3c, 3d, and 3e**.

ii. Generating Adversarial Examples

This section will provide a technical overview of how adversarial examples are generated in the full knowledge environment. One should note that the techniques detailed herein provide no guarantees over whether a generated image will be misclassified by a targeted ML system. Nonetheless, these images are denoted "adversarial." This paper employs the following notation, largely informed by the approach of Kurakin et al. (2016):

- X : an input image, represented as a tensor along the dimensions of width, height, and depth.
- y_{true} : the correct class label.
- $C(X, y)$: the neural network’s cost function for an input X and class prediction y . If a network outputs a softmax distribution across classes and uses a cross-entropy cost function, the cost will be equal to the negative log-likelihood of the correct class: $C(X, y) = -\log p(y|X)$.
- $Clamp_{X,\epsilon}\{X'\}$: a function that clamps the pixel values of X , ensuring that adversarial example X' pixel values are within ϵ of the original image X , where ϵ is a modifiable hyperparameter. This limits the added noise and ensures the adversarial example is nearly visually identical to the input. The function is defined as follows:

$$Clamp_{X,\epsilon}\{X'\}(x, y, z) = \min(X(x, y, z) + \epsilon, \max(0, X(x, y, z) - \epsilon, X(x, y, z)))$$

where $X(x, y, z)$ refers to the z channel’s value at pixel location (x, y) .

Fast Gradient Sign Method Originally introduced by Goodfellow et al. (2014), this method requires only a single back propagation call to retrieve an error signal and assumes a relatively linear cost function. It is less precise and generates successful adversarial examples with less subtlety than following methods but can be calculated rapidly.

$$X^{adv} = X + \epsilon \text{sign}(\nabla_X C(X, y_{true}))$$

Iterative Non-targeted Method This is an iterated extension of the fast method where an adversarial example is repeatedly generated and clamped at each step. The gradient error applied at each step is modulated by a learning rate α :

$$X_0^{adv} = X, \quad X_{N+1}^{adv} = \text{Clamp}_{X, \epsilon}(X_N^{adv} + \alpha \text{sign}(\nabla_X C(X_N^{adv}, y_{true})))$$

The number of iterations run during experimentation was balanced to allow for fast generation times as well as interesting results.

Iterative Targeted Method This is a modified version of the iterated method and a slight modification of the iterative least-likely class method devised by (Kurakin et al., 2016). Instead of just increasing the error of the originally predicted class, this method seeks to decrease the error of a specific selected class label y_{target} . To generate an adversarial example, the method maximizes $\log(p(y_{target}|X))$ by transforming the image in the direction of $\text{sign} \nabla_X \log(p(y_{target}|X))$.

$$X_0^{adv} = X, \quad X_{N+1}^{adv} = \text{Clamp}_{X, \epsilon}(X_N^{adv} - \alpha \text{sign}(\nabla_X C(X_N^{adv}, y_{target})))$$

The same hyperparameters and number of iterations can be employed as in the non-targeted approach.

III. EXPERIMENTAL RESULTS

i. Experimental Design

To compare the capabilities of each adversarial example generation method, we examine the performance of each method on a subset of ImageNet images across a range of ϵ values. Intuitively, an adversarial image generated with $\epsilon = 0$ yields the same image as before. A higher ϵ indicates more leeway for modification of the original image. The learning rate α is held constant at 1 across all experiments. To examine the targeted method, we choose a random target class to optimize for. Although there is a chance that this random class will lay close to the true class, this is rather unlikely across 1000 possible outputs. Due to the computationally expensive nature of generating adversarial examples with the iterative methods, we unfortunately had to resort to a relatively small sample size at each value of ϵ , quantitatively evaluating each method on a subset of only 20 random samples. We also perform a qualitative analysis of each method’s performance and behavioral tendencies by examining generated adversarial images.

ii. Experimental Results

To evaluate performance, we compare the model’s predictions for adversarial images against its predictions for unmodified images. As mentioned above, there are no guarantees on whether

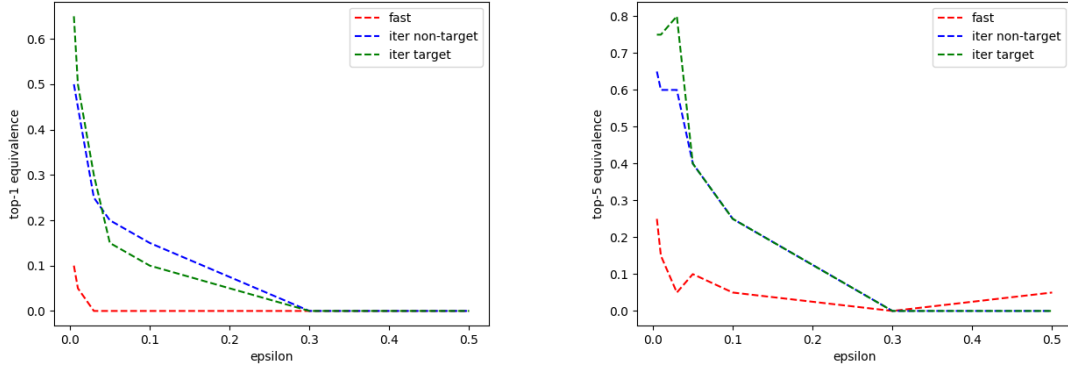


Figure 2: Top-1 and top-5 model accuracy when processing adversarial examples. The fast gradient sign, iterative non-targeted, and iterative targeted methods are compared for different values of ϵ .

an adversarial image generated with these methods will successfully fool the model. First, each method was evaluated on its ability to alter the model’s top-1 accuracy. We define top-1 accuracy to be the rate at which the model’s top prediction for each initial image matches its top prediction for each adversarial image. Next, we evaluated each method on its ability to alter the model’s top-5 accuracy. We define top-5 accuracy to be the rate at which the model’s top prediction for each initial image appears within its five top predictions for each adversarial image. Results for each adversarial method’s effectiveness at different values of ϵ can be seen in **Figure 2**.

The motivation for examining both top-1 and top-5 accuracy lies in the difference between the targeted and non-targeted methods. The targeted method seeks to increase the likelihood of a random or alternative class. As discussed, it is likely that this class is far away from the initially predicted class. Because of this distance, the error gradient will pull the adversarial image away from all the top classes (which should be grouped together in the class space) faster. When the *non-targeted* method is run, the likelihood of the top class is reduced, and though this in turn will reduce the likelihood of classes in its neighborhood, there is less of a pull towards a completely new area in the latent class space. In summation, we might hypothesize that the targeted method would reduce the top-5 accuracy of the model more rapidly than the non-targeted method. In our limited results, this hypothesis was not confirmed as the targeted and non-targeted attacks performed very similarly.

One interesting thing to note is the apparent effectiveness of the fast gradient sign method. This is slightly misleading, however, as this method operates in a less subtle fashion than the other two by effectively introducing ϵ -scaled noise in a single shot. For a given ϵ , the fast method is far more "destructive" as it modifies individual pixel data more aggressively. The iterative methods attempt to create an attack smoothed over the image space, thereby reducing the visual artifacts introduced by modification. **Figure 3a** visually demonstrates the destructive nature of the fast method. The swans in the image have been visibly manipulated through the introduction of foreign colors even at a relatively small ϵ . **Figure 3b** illustrates an iterated attack at the same ϵ that introduces far less visible noise.

Figures 3b and **3c** showcase the non-targeted and targeted adversarial methods best. Any noise introduced is remarkably subtle, even though the model’s class predictions for the attacking image are completely different. In the first image, the model changes its prediction from "convertible" to

"crayfish"; in the second, from "Granny Smith" apples to "cello." A human looking closely might be able to spot defects within these images, but they would never make the same severity of classification mistakes as the model. **Figures 3d** and **3e** are included to demonstrate the destruction introduced by the iterative algorithms at higher values of ϵ . Images generated with integer values of ϵ and higher display significant artifacts. Interestingly, these adversarial images would still likely be correctly classified by a human even though the model may produce an incorrect prediction with near 100% confidence. This last point is an important one to note regarding adversarial attacks. When the model processes these adversarial images, its misclassifications are incredibly confident. In **Figure 3d**, for example, the model does not "see" strange looking apples—it very confidently "sees" a cello.

In summary, the fast gradient sign method is effective, but more destructive and less subtle than the iterative methods. Both iterative methods performed similarly quantitatively, though in models with a large number of output classes we expect the iterative targeted approach to be more successful in reducing accuracy, especially top-5 accuracy. While generating adversarial images with iterative methods is slower, a qualitative analysis indicates their superiority in fooling the model while still retaining visual information from the original image.

IV. DEFENSIVE AND ETHICAL CONSIDERATIONS

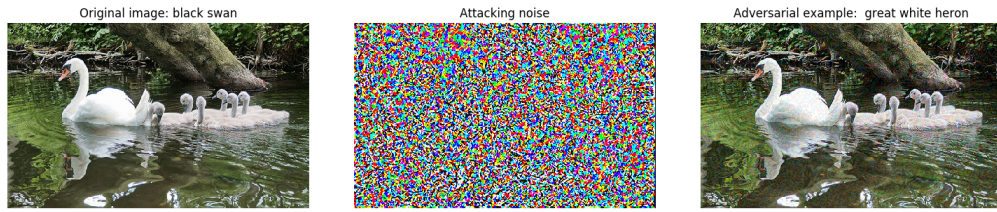
In this section, we will discuss current research on potential defensive mechanisms that could be employed to combat adversarial attacks, as well as the ethical considerations that must be weighed before deploying systems susceptible to such attacks.

i. Defensive Approaches

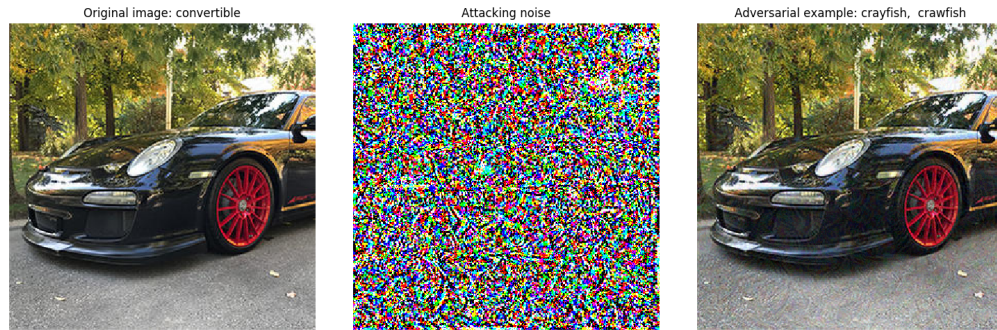
When assessing the security of an ML system deployed in the wild, there are many attack vectors to consider. An adversary may attempt to gain access to the deployment of the model itself or perhaps provide malicious inputs if a model is trained on-line. We saw this latter case play out particularly poorly for Microsoft with the launch of its chatbot Tay (Lee, 2016). For the purposes of this paper, we will narrow the scope to focus only on defending against attackers employing the types of subtle adversarial techniques shown thus far.

These defenses can be distilled into two primary camps: *reactive* and *proactive* (Mikhailov and Trusov, 2017). An example of a reactive defense might be the preprocessing and/or sanitization of all inputs by another model. It is feasible to train a model to recognize adversarial inputs before they ever reach the primary classifier. This solution is far from ideal, however, as it requires the maintenance of two heavy duty models in production instead of one. It also opens up the possibility of incorrectly flagging legal inputs, which could be more limiting to the original system than the possibility of adversarial inputs in the first place. There is a certain inelegance to this approach as well; though the problem of robust vision is far from simple, the fact that humans do not fall prey to these types of schemes is encouraging to the technical community that more complete proactive solutions exist.

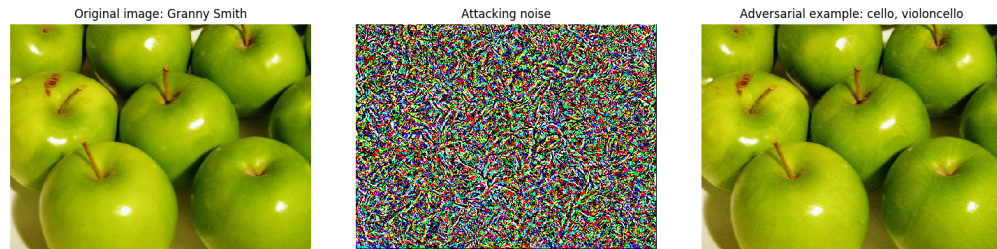
Adversarial Training One current leading proactive approach is that of adversarial training. During the training process, the gradients of the unfinished model are used to generate adversarial examples with the same methods described earlier. These adversarial examples are then used to augment the training dataset, thereby increasing the robustness of the model against images with these previously unnoticed perturbations (Papernot et al., 2016c). Though adversarial training does improve the resilience of a model against such attacks, along with the added benefit of



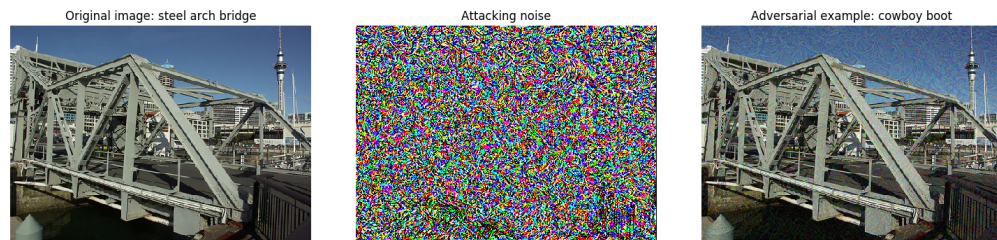
(a) Fast gradient sign attack with $\epsilon = 0.05$. The attack modifies the prediction, but yields a similar class.



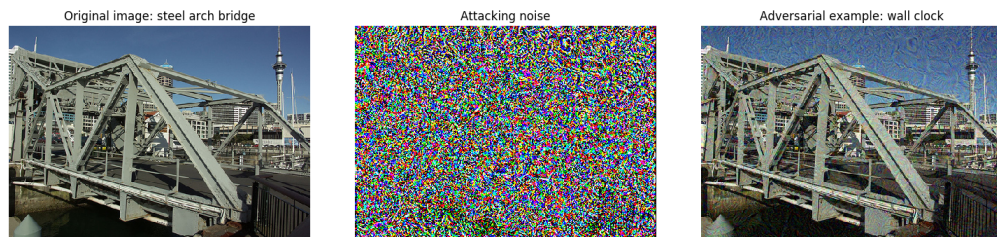
(b) Iterative non-targeted attack with $\epsilon = 0.05$. The attack successfully modifies the prediction.



(c) Iterative targeted attack with $\epsilon = 0.02$. The attack was successful given the target class "cello."



(d) Iterative targeted attack with $\epsilon = 0.08$. The attack was successful given the target class "cowboy boot."



(e) Iterative targeted attack with $\epsilon = 10.0$. The attack was successful given the target class "wall clock."

Figure 3: Examples of adversarial images generated with different methods and ϵ values. Above each image the model's most likely class prediction is shown.

increased regularization, the decision boundaries remain relatively fragile. An attacker with more computing power still seems able to locate weaknesses. One drawback is the increased complexity of training; additional computing resources and time are necessary to realize this training schema. Nevertheless, the adversarial training approach is a first step in the direction of developing more durable models.

Gradient Masking, Defensive Distillation, and Label Smoothing Another set of approaches rely on minimizing the information available to an attacking algorithm. Though these techniques are technically different, they are similarly motivated and will thus be considered in tandem. In the process of distillation, a larger, more complicated model is compressed into a smaller form while sacrificing a very small amount predictive accuracy. The intuition behind this approach is that the smaller model will learn a "softer" probability distribution and will encode less helpful information to an attacker in its output predictions than the larger model (Papernot et al., 2015). Gradient masking similarly relies on information reduction by artificially limiting gradient norm during training, thus dramatically decreasing the signal available to an attacker at inference time. A final approach in this category is label smoothing, where the output probabilities are held closer together (i.e., there is no single class with extremely high probability); this should theoretically limit information by making it more difficult to target a specific class (Papernot et al., 2016c).

Though these defenses have been shown to increase the difficulty of generating adversarial examples, they are not insurmountable at this time. By the transferability property introduced by Papernot et al. (2016a), an attacker can create a surrogate model given access to similar training data and basic knowledge of the target model's architecture. This surrogate model can then be used to obtain a gradient not dissimilar from that of the target, which can then in turn be used to fine tune adversarial examples (Papernot et al., 2016c). This surrogate model makes the distinction between full knowledge and black box scenarios far less relevant.

ii. Ethical Considerations

We have technically described the nature of adversarial attacks and the limitations of available defenses, but we have not yet made it clear why we should care about such vulnerabilities. Though ML systems have reached mass application in some domains, we are only at the beginning of the adoption curve. Many traditional software systems will be replaced by more intelligent ML algorithms, and sometimes not without cost. Despite their increased flexibility and intelligence, systems built with neural networks and similar approaches are not only susceptible to adversarial attacks, but are also increasingly opaque in their decision-making. As our society continues to place a greater quantity of increasingly complex decisions in the hands of these systems, it is concerning that our understanding of their operation seems to be decreasing.

We can conceive of highly critical applications in which adversarial examples are particularly troubling. Banks that process checks programmatically might easily be defrauded by an adversarial forgery that appears legitimate to a human. Autonomous weapon systems that rely on computer vision, as described by Arkin (2010) and others, might be confused by disguised weapons or tricked into firing on innocents that are invisibly marked against their knowledge. An autonomous vehicle might be fooled into interpreting a stop sign as a 45 mph sign (Evtimov et al., 2017). This last example is specifically relevant given the seeming inevitability of autonomous vehicles overtaking our streets in the coming years. Further, it is concerning that this specific attack has already been demonstrated successfully in real world settings.

Even though these models exhibit clear defects in adversarial settings, it is difficult to make a conclusive recommendation on their deployment. Improving training practices is a good place for

the technical community to start, though this is easier said than done. The work of Leike et al. (2017), which focuses on improving outcomes through the tighter integration of human oversight during training, is a positive contribution to the advancement of AI safety; unfortunately, it is not applicable when the problems encountered during training are at times more subtle than humans can perceive. Adversarial training should be encouraged in the development of these models, and it brings along the positive side effect of increased regularization. Information masking techniques, despite their provable fragility, also do increase the safety and resilience of ML systems deployed in the real world.

Fairness Even beyond the scope of adversarial examples, it is important to think about the fairness of employing such opaque models. In recent work evaluating the fairness of recidivism prediction systems, Chouldechova (2016) demonstrated that bias free predictive instruments can still result in disparate impact across populations when input data is not carefully curated. Beyond augmenting ML training data with adversarial examples, it is necessary for the technical community to become more careful about dataset construction, especially in ethically complex domains like recidivism prediction. Institutions and organizations could require the use of black box auditing tools like FairML¹ to prevent the manifestation of obvious biases in production systems.

Accountability As we do employ these more advanced systems, we develop an increased expectation of their accountability. This accountability is often a justification for their necessity. Despite limitations, autonomous vehicles are easy to justify when they are expected to dramatically reduce the frequency of accidents (Blanco et al., 2016). Our evaluation of the accountability of these systems, however, is framed by an assumption of generally good faith actors. The ethical calculus of autonomous vehicles and many other systems certainly changes in the face of malicious adversarial attacks. Can we permit these systems to be responsible for human lives when it is seemingly simple for a knowledgeable actor to influence them with attacks hidden in plain sight? At this current juncture given the limited scope of these systems in real world applications, there is little cause for concern. But in coming years when ML systems dictate greater portions of our lives, slight perturbations may indeed be a worthy anxiety.

Trade-offs The systems in development today are far from perfect, and in all likelihood they'll continue to have flaws for the remainder of their existence. It is necessary then not to write them off entirely, but to evaluate their trade-offs. One of the essential trade-offs in models vulnerable to adversarial attacks is the trade-off between representative capacity and interpretability. Hornik (1991) showed that multilayer feedforward neural networks are universal approximators; essentially, a model with enough parameters can represent any function mathematically. As model size and complexity increase, however, the variety and quantity of data required to prevent overfitting increases.

At this time, our models are limited in representative capacity by available computing power and data; this opens them up to the kinds of manipulations abused by adversarial attacks. In order to combat these attacks in coming years, more complicated models will be trained on larger datasets. This practice, though, comes at the cost of interpretability. The decisions made by such models will become progressively difficult to explicate, which might prove problematic in fields like health care where the reasoning behind certain decisions should be comprehensible (Consortium et al., 2009). There is no obvious solutions to these concerns. At first, it makes sense

¹<https://github.com/adebayoj/fairml>

to require that models used in such critical applications should be highly interpretable. Yet models limited by such a requirement will have lesser representative power and will in turn make less intelligent decisions. Would we be willing to sacrifice potentially worse outcomes for a better understanding of how they came about?

In total, a model should be evaluated not just on its predictive accuracy, but also along the dimensions of robustness, fairness, accountability, and interpretability. There is no free lunch when it comes to ML models, and these properties are at times orthogonal to one another. That said, the threat of adversarial attacks, especially in ethically critical domains, should be taken seriously. Defensive techniques like adversarial training should be used to improve the resilience and safety of models in the field.

V. CONCLUSION AND FUTURE WORK

In this paper we have provided an overview of the current state of adversarial attack research as it pertains to machine learning models. We have reviewed several adversarial image generation methods and identified the family of iterative methods as particularly promising given their more subtle approach. Aside from a technical evaluation of attacks, we have outlined primary known defensive mechanisms alongside analysis of their shortcomings. Finally, we situate this work within the AI safety discussion and elucidate ethical concerns surrounding the deployment of models susceptible to adversarial attacks. Despite deficiencies in defensive mechanisms, we encourage the technical community to take these methods seriously to increase the safety and accountability of models employed in real world settings. Future work should focus on developing adversarial attacks and defenses together while at the same time improving model interpretability. A more nuanced understanding of how opaque ML models form decision boundaries will be vital to their successful deployment in coming years.

REFERENCES

- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.
- Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016a.
- Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016b.
- Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *CoRR*, abs/1707.08945, 2017.
- Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, Deepak Verma, et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108. ACM, 2004.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.
- Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *CoRR*, abs/1511.04508, 2015.
- Nicolas Papernot, Patrick D. McDaniel, Arunesh Sinha, and Michael P. Wellman. Towards the science of security and privacy in machine learning. *CoRR*, abs/1611.03814, 2016c.
- Peter Lee. Learning from tay’s introduction. <https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/>, 2016. [Online; accessed 13-December-2017].
- Emil Mikhailov and Roman Trusov. How adversarial attacks work. <http://blog.ycombinator.com/how-adversarial-attacks-work/>, 2017. [Online; accessed 3-December-2017].

- Ronald C Arkin. The case for ethical autonomy in unmanned systems. *Journal of Military Ethics*, 9 (4):332–341, 2010.
- Jan Leike, Miljan Martić, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. Ai safety gridworlds. *arXiv preprint arXiv:1711.09883*, 2017.
- A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *ArXiv e-prints*, October 2016.
- Myra Blanco, Jon Atwood, Sheldon Russell, Tammy Trimble, Julie McClafferty, and Miguel Perez. Automated vehicle crash rate comparison using naturalistic data, 01 2016.
- Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4 (2):251–257, 1991.
- International Warfarin Pharmacogenetics Consortium et al. Estimation of the warfarin dose with clinical and pharmacogenetic data. *N Engl J Med*, 2009(360):753–764, 2009.

A. APPENDIX

Code All code used to generate adversarial examples can be found at <https://github.com/cgyulay/adversarial-attacks>. Experiments were run using the PyTorch library.

Word Count The final word count is 5051.