FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Jakub Hejhal

Exploring the vulnerabilities of commercial AI systems against adversarial attacks

Department of Theoretical Computer Science and Mathematical Logic

Supervisor of the bachelor thesis: Mgr. Roman Neruda, CSc.
Study programme: Computer Science
Study branch: General Computer Science

Prague 2021

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

I would like to thank Mgr. Roman Neruda, CSc. for his patiance, helpful advice and guidance throughout my thesis work. I would also like to thank my family, my friends and my girlfriend for supporting me and for providing me with love and care that allowed me to push through occasionaly difficult times.

Title: Exploring the vulnerabilities of commercial AI systems against adversarial attacks

Author: Jakub Hejhal

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: Mgr. Roman Neruda, CSc., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Abstract. TODO

Keywords: Machine learning Deep learning Adversarial attack Black-box

Contents

1. Introduction

1.1 The rise of artificial neural networks

In recent years there has been an enormous surge in applications of artificial intelligence technologies based on neural networks to various fields. One of the most significant milestones that kickstarted today's AI revolution has undoubtedly been the year 2012. ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which hand-crafted specialized image-labeling systems have previously dominated, was won by AlexNet with its CNN architecture.

Artificial neural networks inspired by their biological analog had been known and researched for a long time. Even though they enjoyed much enthusiasm initially, they had been neglected as an unpromising direction towards general intelligence. More classical approaches like SVM and rule-based AI systems showed better performance and computational efficiency on AI benchmarks of the time.

What changed the game has been the available computational power that came with Moore's law and the usage of GPUs, mainly their parallel nature in accelerating matrix multiplication operations that neural networks use heavily.

Another factor that helped the rise of neural networks has been the availability of large datasets like ImageNet, which contains 1,281,167 images for training and 50,000 images for validation organized in 1,000 categories.

The availability of large amounts of labeled and unlabeled data, sometimes referred to as "Big data," only seems to get better. A large part of our lives has moved to the virtual space thanks to the internet. Businesses had started to realize the value of the enormous amounts of traffic generated every day, and they are increasingly trying to figure out how to take advantage of it.

What was previously limited to academic circles had quickly become mainstream. Artificial neural networks have proven to be very versatile and have quickly been successfully applied to a wide range of problems. New neural network architectures and new training regimes allowed training deeper networks, which gave rise to a new field of machine learning called "Deep learning."

Deep neural networks have shown state-of-the-art performance in machine translation, human voice synthesis and recognition, drug composition, particle accelerator data analysis, recommender systems, algorithmic trading, reinforcement learning, and many other areas.

1.2 Concerns

Large-scale deployment of neural network systems has been criticized by many for their inherent unexplainability. It is often hard to pinpoint why neural network behaves in some way and why it makes certain decisions. One problem is the role of training data, where possible biases may be negatively and unexpectedly reflected in the behavior of the AI system. Another problem is the performance on out-of-distribution examples, where network inference occurs on different kinds of data than used in the training stage.

Those concerns lead people to study the robustness of AI systems. It turned out that image recognition CNN networks are especially susceptible to the so-called adversarial attacks, where the input is perturbed imperceptibly, but the output of the network changes wildly. Similar kinds of attacks have since been demonstrated in other areas like speech recognition, natural language processing, or reinforcement learning.

1.3 Adversarial attacks

This vulnerability of neural networks has led to a cat-and-mouse game of various attack strategies and following defenses proposed to mitigate them.

Neural networks can be attacked at different stages:

- training
- testing
- deployment

Training attacks exploit training dataset manipulation, sometimes called dataset poisoning, to change the behavior of the resulting trained network.

Testing attacks do not modify trained neural network, but often use the knowledge of the underlying architecture to craft adversarial examples which fool the system.

Deployment attacks deal with real black-box AI systems, where the exact details of the system are usually hidden from the attacker. Nevertheless, partly because similar neural network architectures are used in the same classes of problems, some vulnerabilities in testing scenarios can still be exploited in deployment, even though the exact network parameters, architecture, and output are unknown to the attacker.

The purpose of this thesis is to explore the applicability of certain classes of testing attacks on real-world deployed AI systems. For simplicity, many kinds of SOTA adversarial attacks have only been explored in the testing regime but have not been applied to truly black-box systems.

The main aim of the thesis will be to test different types of testing attacks on AI SaaS providers like Google Vision API, Amazon recognition, or Clarify. Understandably, attacking an unknown system will be more challenging than attacking a known neural network in the testing stage. We will try to measure this attack difficulty increase. This information could prove helpful in selecting the most promising attack to a specific situation.

Many SOTA testing attacks were not designed to attack specific deployed systems, so some attacks will need to be slightly changed to be used. We will explore different ways to modify existing attacks and evaluate them.

If some of those services are proven to be vulnerable, this would have a massive impact on all downstream applications using those SaaS APIs. For instance, content moderation mechanisms, which rely mainly on automatic detection, could be circumvented.

2. Preliminaries

In this section we briefly introduce and explain the necessary theoretical background, upon which we build later on.

We first define the concepts of Machine learning (ML) and Deep neural networks (DNNs). We explain, how DNNs can learn patterns from data by using powerful gradient-based stochastic optimization algorithm called Stochastic gradient descent (SGD). Then we talk about a subset of DNNs that perform very well on image data called Convolutional neural networks (CNNs). Finally, we describe the systemic vulnerability of DNNs and we define and explain adversarial attacks and defenses, adversarial examples, adversarial robustness and different adversarial attack threat models.

2.1 Machine learning

Goodfellow et al. [2016]

2.2 Deep neural networks

2.3 DNN training

2.4 Convolutional neural networks

2.5 Adversarial attacks on DNNs

2.6 Threat models

3. Related Work

3.1 Title of the first subchapter of the third chapter

TODO: rewrite your review papernotes.md here.

4. Our approach

4.1 Adapting off-the-shelf attack algorithms to partial-information setting

Cloud-based image classifiers don't usually classify input images into a fixed number of classes. They instead output variable-length list of probable labels with scores. And what's worse, those scores aren't even probabilities, because they don't sum up to one.

Most of current score-based SoTA adversarial attacks assume that we have access to all output logits of the attacked target classifier. If we want to use them, we need to map somehow the cloud's variable-length score-output to fixed-length vector, which will simulate output of a standard CNN classifier.

4.1.1 Object-organism binary classification mapping

To simplify the experiments, we define a simple benchmark attack scenario. Given an image containing a living thing, fool the target into classifying it as a non-living object.

We chose this split, because ImageNet dataset is relatively balanced between those two semantic categories. Why ImageNet? Despite the dataset being quite old, it is still the most heavily used dataset in the research community and the majority of freely available pretrained models are pretrained on it.

Furthermore, when $\|C\| = 2$ (where C is a set of output categories), targeted and untargeted attack scenarios don't differ anymore and are neatly unified.

Adapting the attack algorithm to a different attack objective only requires swapping the label mapping layer.

Imagenet category mapping

Classic ImageNet dataset contains real-world photos, each corresponding to one and only one classification category out of 1000 possible categories. Each ImageNet category c corresponds to a unique wordnet synset $S(c)$. These synsets are rather specific, but we can take a look at their set of hypernyms $h(S(c))$. If this hypernym set $h(S(c))$ contains the *organism* synset, it should be an organism, otherwise c is probably an object.

Writen more rigorously, we map each ImageNet category c into $\{organism, object\}$ using the following maping $M(c)$:

$$M(c) = \begin{cases} organism & organism \in h(S(c)) \\ object & organism \notin h(S(c)) \end{cases}$$

Cloud label mapping

This situation isn't so clear-cut in the case of general labels returned by cloud classifier. Labels might not even be single words, but whole sentences. Therefore we use a powerful GPT-2 transformer to encode text labels into embedding vector space and then compute their similarity with carefully chosen set of 4 labels: $\{animal, species, object, instrument\}$. Resulting mapping is the *argmax* over those 4 similarity scores.

4.2 PoC black-box GVision attacks

We first explored the viability and sample-efficiency of current SoTA black-box attacks. We ran the following against Google Vision API image classifier.

- 4.2.1 TREMBA (Huang and Zhang [2020])
- 4.2.2 RayS (Chen and Gu [2020])
- 4.2.3 SquareAttack (Andriushchenko et al. [2020])
- 4.2.4 Sparse-RS (Croce et al. [2020])

TODO: describe each attack briefly and show sample images

4.2.1 TREMBA

4.2.2 RayS

4.2.3 SquareAttack

4.2.4 Sparse-RS

We go into more details in the Experiments 5

4.3 The need for query-efficient attack

In the previous section 4.2 we empirically showed, that Google Vision API isn't 100% robust to iterative blackbox atttacks. But although the previously mentioned black-box attacks are often successful in producing adversarial image, query count to the target may be often unacceptably high. The problem is that these blackbox attacks (with the exception of TREMBA) mostly rely on random search and don't make use of the gradient similarity of various CNN models. In princinple, they cannot do much better alone. Huge query stress to the target is troublesome for several reasons:

- High cost - 1.5$ per 1000 queries
- Raising suspicion
- Often unrealistic in practical setting

4.4 Leveraging transferability

After these early experiments that proved the concept, we focused our attention on transferability, which has a huge potential to save queries.

4.4.1 Transfer attacks provide better seeds for blackbox optimization

Even if the localy-produced adversarial images don't transfer directly to the target, Suya et al. [2020] showed, that the output of transfer-attack can provide better starting seeds for blackbox optimization attacks and improve their query efficiency, which basically adds a degree of freedom to the blackbox optimization. They also discuss different prioritization strategies, as the the number of seeds produced isn't limited by target queries and we can therefore afford to produce as many candidate starting points as we like.

4.4.2 Train, validate, test

There is a weak analogy between the crafting of an adversarial example and the training of machine learning model. ML model weights are first fitted againt specified loss contraint. This constraint is (among other things) a function of training data. The weights are then validated and checked against overfitting on a slightly different constraint, which now depends on a validation dataset. When all is good, model is happily deployed to production.

With a bit of imagination, ML model weights correspond to pixel values of an adversarial image. The pixels are first trained by gradient descent on training loss provided by a surrogate model. They are validated against ensemble set of diverse indepenent classifiers, and when the foolrate is good, they are sent for test evaluation to the cloud.

4.4.3 Local training and validation is cheap

We want to offload the cloud query-stress to local simulation as much as possible. An attacker can often afford to spend orders of magnitude more queries to local surrogates and validation models than to the actual target.

4.4.4 Multiple candidates save queries

Iterative black-box attacks usually have query-distrubutions which are tail-heavy. In other words, the median queries needed to create a successful adversarial image are much lower than the average queries.

Let's image an attack scenario, where we want to submit a photo to a platform with automatic content moderation mechanism. Querying the target hundreds of times would certainly attract unwanted attention and our heavy queries can quickly trigger human evaluation. If our primary goal is to craft only one adversarial image and as much as possible evade detection, having multiple candidate images would give us another degree of freedom and it could potentially mitigate the heavy-tail problem. This approach can be in principle transparently combined with the multiple-seed candidate suggestions mentioned in 4.4 using the same prioritization candidate scoring mechanism.

4.5 The need for attack pipeline

We argued in 4.4 that combining multiple whitebox and blackbox attack approaches could create more powerful attack as well as giving us more freedom and flexibility to tailor this combination to a specific attack scenario constraints. As of now, there isn't any general whitebox/blackbox attack pipeline which would combine different algorithms and allow us attacking cloud services in a practical way.

4.5.1 Possibility of multiple blackbox workers

We can also image running multiple different attacks in paralel and having some meta-controller orchestrating individual attack algorithms such that we minimize queries to the target and efficiently make use of the additional degrees of freedom.

4.5.2 The need for unified attack and model API

There are sereval frameworks unifing whitebox/blackbox attacks. To name a few, there is FoolBox (Rauber et al. [2020]) or AutoAttack (Croce and Hein [2020]).

Although they are excellent at testing the robust of local models, they don't give us the flexibility we need to implement all the pipeline features mentioned previously. They cannot be used without some modification to attack cloud models and their optimization attacks cannot be cooperatively scheduled step by step, which is what would be required for effective multi-attack orchestration.

4.6 AdvPipe

4.6.1 The vision

Show what was the plan.

4.6.2 The reality

This is sad. :(

Make some excuses why you didn't make it in time.

Make some flowchart of what is actually working.

4.6.3 Implementation

5. Experiments

5.1 Blackbox PoC

Here we go into more technical details about previously mentioned blackbox attacks we initially tried.

- TREMBA
- RayS
- SquareAttack
- Sparse-RS

5.1.1 TREMBA

5.1.2 RayS

This one is hard label attack and doesn't use the continuos loss from GVision.

5.1.3 SquareAttack

SquareAttack L2

Show an image of nice cat.

SquareAttack Linf

Comparison with local success-rate

5.2 Local transferability experiments

Here go all the different transfer-matrices

5.2.1 Choice of dataset

5.2.2 Choice of classifies

5.2.3 Baseline

5.3 Transferability evaluation on GVision

TODO: just run the images against GVision

5.3.1 Choice of evaluation metrics

5.3.2 Wordcloud

Conclusion

Bibliography

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.

J. Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *CoRR*, abs/2003.01690, 2020. URL `https://arxiv.org/abs/2003.01690`.

Francesco Croce, Maksym Andriushchenko, Naman D. Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. *ArXiv*, abs/2006.12834, 2020.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Z. Huang and Tong Zhang. Black-box adversarial attack with transferable model-based embedding. *ArXiv*, abs/1911.07140, 2020.

Jonas Rauber, Roland S. Zimmermann, M. Bethge, and W. Brendel. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *J. Open Source Softw.*, 5:2607, 2020.

Fnu Suya, Jianfeng Chi, David Evans, and Y. Tian. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. *ArXiv*, abs/1908.07000, 2020.

List of Figures

List of Tables

List of Abbreviations

A. Attachments

A.1 First Attachment