Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

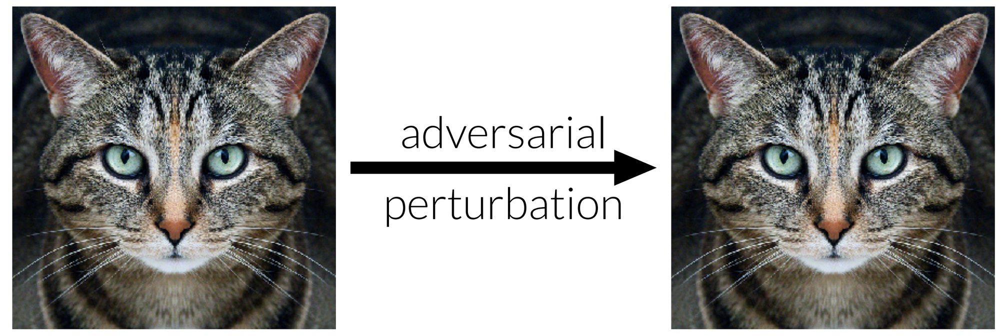
Anish Athalye* [1] Nicholas Carlini* [2] David Wagner [2]

[1]MIT CSAIL [2]UC Berkeley

Adversarial examples

Adversarial examples are inputs designed to fool a neural network.

Formalization often used: for a clean input $\mathbf{x}$, an input $\mathbf{x}'$ is an adversarial example if it is misclassified and $d(\mathbf{x}, \mathbf{x}') < \epsilon$. For example:



88% tabby cat 99% guacamole

Threat models

A threat model is a formal statement describing assumed limitations on an adversary. We consider defenses that claim to be secure under:

- **White-box**: attacker has access to architecture and parameters
- **Adaptive adversary** (Kerckhoff's Principle): attacker knows defense
- **Perturbation bound**: limited perturbation in some distance metric

We analyze each paper under the specific threat model it considers.

Obfuscated gradients

Gradient-based attacks cannot succeed without a gradient signal.

We identify three types of obfuscated gradients:

- **Shattered gradients** are nonexistent gradients due to non-differentiable operations;
- **Stochastic gradients** depend on test-time randomness; and,
- **Vanishing/exploding gradients** arise in very deep or recurrent computation.

Defenses that cause obfuscated gradients appear to defeat iterative optimization-based attacks, but defenses relying on this effect can be circumvented using new techniques we develop.

Evaluating defenses

Evaluation Goal: show that the defense is secure under the threat model.

There is no test set for security. There are no fixed benchmarks.

The job of a security evaluation is *not* to show the defense is *right*, it is to *fail* to show that the defense is *wrong*.

Red flags: signs of obfuscated gradients

We identify several characteristic behaviors of defenses that obfuscate gradients. Such defenses may be weak even though they appear to defeat standard iterative attacks.

- **Single-step attacks outperform iterative attacks**. Iterative attacks like PGD should give strictly better performance than single-step attacks like FGSM.
- **Black-box attacks outperform white-box attacks**. The black-box threat model is a strict subset of the white-box threat model, so attacks in the white-box setting should perform better.
- **Attack success rate should be non-decreasing**. With an increased perturbation bound, attack success rate cannot decrease.
- **Unbounded attacks do not reach 100% success**. With unbounded distortion, any classifier should have 0% robustness.
- **Results vary widely between optimization-based attacks**. All gradient-based attacks should achieve roughly similar performance with good parameter selection and enough iterations.
- **Random sampling finds adversarial examples**. If brute-force random search (e.g. sampling 10^5 points) within the feasible set finds adversarial examples, the defense is likely obfuscating gradients.

Recommended practices

Some simple sanity checks can help identify weak defenses:

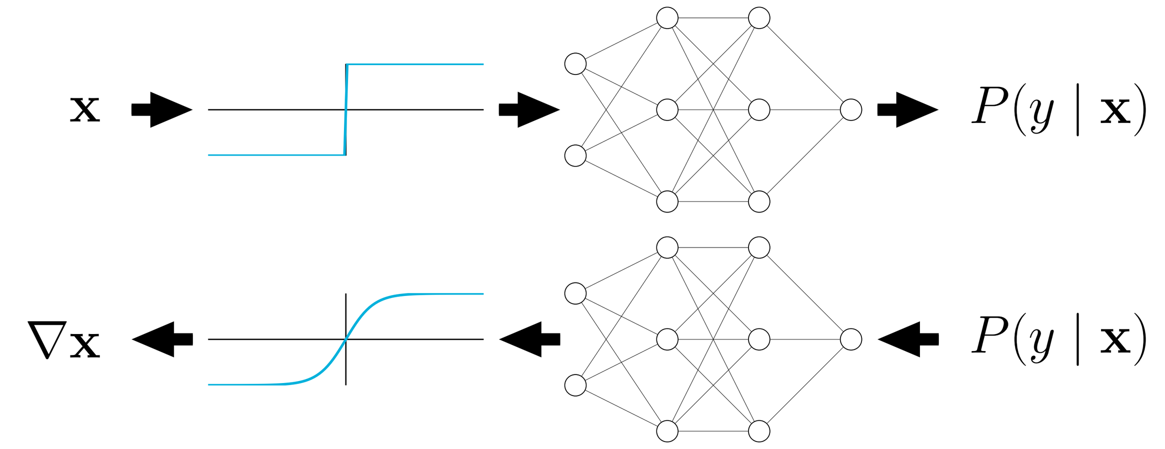
- **Use many iterations of gradient descent**. Number of gradient descent steps is not a security parameter: use many (> 1000) steps of gradient descent to ensure convergence.
- **Evaluate against the strongest attack**. Only the worst-case performance of a defense matters. Evaluate against the meta-attack that tries many strong attacks and returns the best result.
- **Perform a transferability analysis**. Transferability attacks are simple to implement and can help catch defenses that subtly break standard white-box attacks.
- **Evaluate gradient estimation attacks**. Attacks based on gradient estimation [3] can identify defenses that subtly break analytic gradient computation.

Attacks

Obtaining a useful gradient is the primary challenge in attacking a defense that causes obfuscated gradients. Once we obtain a useful gradient signal, we apply PGD [2], a standard white-box attack.

Backward Pass Differentiable Approximation (BPDA)

BPDA allows for attacking non-differentiable networks by approximating the gradient of the non-differentiable layers. The gradient is estimated by computing the forward pass normally but replacing a non-differentiable layer $f(\cdot)$ with a differentiable approximation $h(\cdot) \approx f(\cdot)$ on the backward pass.



Expectation Over Transformation (EOT)

EOT [1] finds adversarial examples that are adversarial over a distribution of transformations T, allowing for attacking defenses that employ randomized input transformations for robustness. EOT solves the following optimization problem:

$$\begin{aligned} &\arg\max_{\mathbf{x}'} \quad \mathop{\mathbb{E}}_{t\sim T} \left[-\log P\left(y \mid t(\mathbf{x}')\right)\right] \\ &\text{subject to } d(\mathbf{x}, \mathbf{x}') < \epsilon \end{aligned}$$

EOT solves the optimization problem using gradient descent, noting that $\nabla \mathbb{E}_{t\sim T} - \log P\left(y \mid t(\mathbf{x})\right) = \mathbb{E}_{t\sim T} \nabla - \log P\left(y \mid t(\mathbf{x})\right)$ and approximating with samples at each gradient descent step.

Reparameterization

Reparameterization allows for attacking networks that cause vanishing/exploding gradients by performing a change of variables.

For a classifier $f(g(\mathbf{x}))$ where $g(\mathbf{x})$ causes vanishing/exploding gradients, we make a change of variables $\mathbf{x} = h(\mathbf{z})$ for a differentiable function $h(\cdot)$ such that $g(h(\mathbf{z})) = h(\mathbf{z})$. With this, we can compute gradients through $f(h(\mathbf{z}))$ to attack the defense.

Case study: ICLR 2018 defenses

Non-obfuscated gradients

- Adversarial Training (Madry *et al.*)
  - 47% @ 0.031 ℓ_∞ on CIFAR-10
- Cascade Adversarial Training (Na *et al.*)
  - 15% @ 0.015 ℓ_∞ on CIFAR-10

Shattered gradients

- Thermometer Encoding (Buckman *et al.*)
  - Circumvented using BPDA
- Input Transformations (Guo *et al.*)
  - Circumvented using BPDA
- Local Intrinsic Dimensionality (Ma *et al.*)
  - Circumvented using PGD and optimizing for confidence

Stochastic gradients

- Stochastic Activation Pruning (Dhillon *et al.*)
  - Circumvented using PGD and using expectation of gradient
- Mitigating through Randomization (Xie *et al.*)
  - Circumvented using EOT

Vanishing gradients

- PixelDefend (Song *et al.*)
  - Circumvented using BPDA
- Defense-GAN (Samangouei *et al.*)
  - Circumvented using reparameterization

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

[2] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations, ICLR 2018*.

[3] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.