The case for dynamic defenses against adversarial examples

Ian Goodfellow

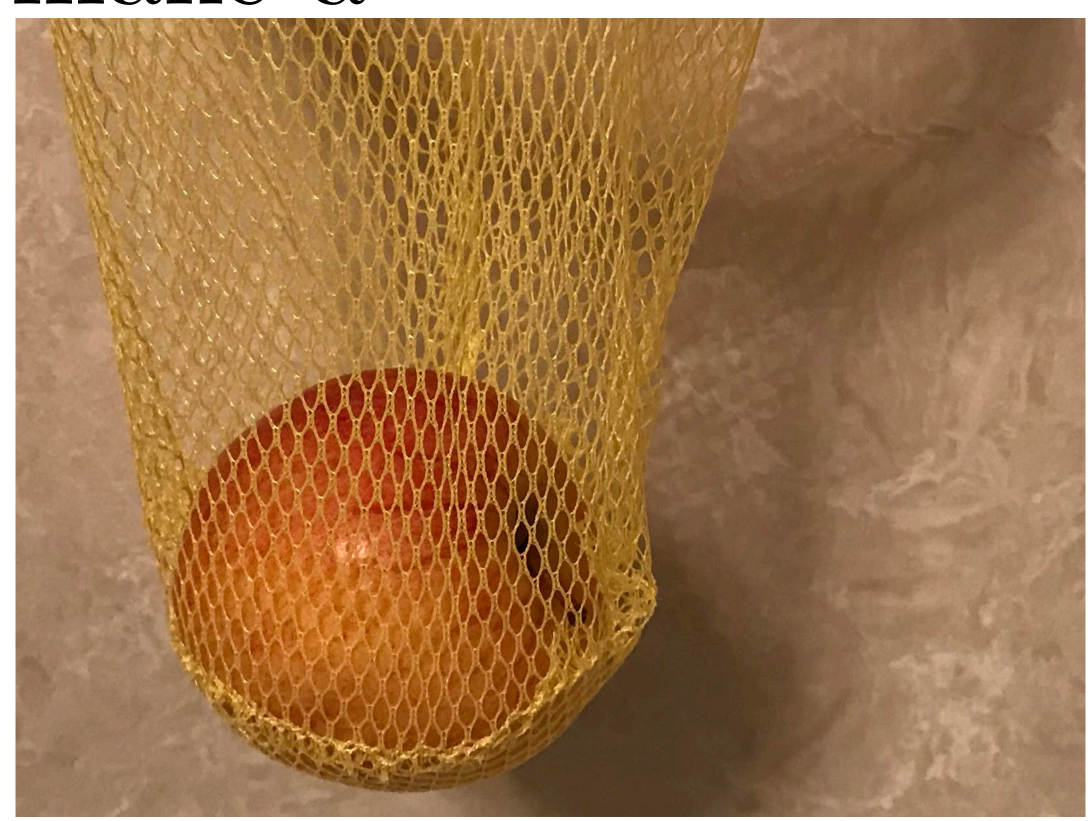
SafeML ICLR Workshop
2019-05-06 New Orleans

Based on https://arxiv.org/pdf/1903.06293.pdf

Definition

"Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake"

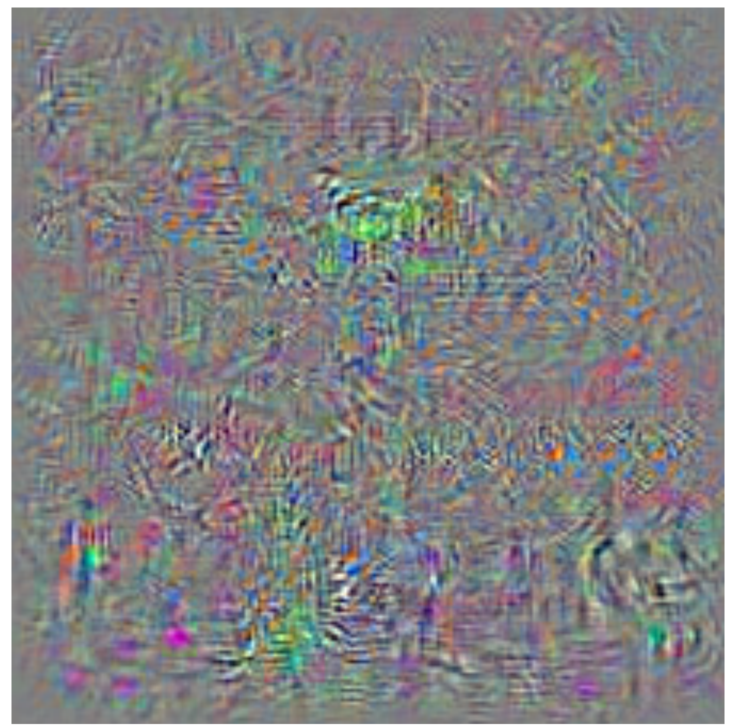
(Goodfellow et al 2017)



Most adversarial example research today

Schoolbus

+



Perturbation

(rescaled for visualization)

=

Ostrich

(Szegedy et al, 2013)

Maximizing the p(airplane|input) reward function

Overfitting to one metric

- In "Explaining and Harnessing Adversarial Examples" I set up this game:
  - World samples an input point and label from the test set
  - Adversary perturbs point within the norm ball
  - Defender classifies the perturbed point
- I expected this to be only moderately difficult and mostly solved quickly
- $>$ 2,000 papers later, still not really solved
- I still think this is a useful task
- It is definitely not the real task and we need to not be myopic

More realistic threat models

- Security
    - Real attackers have no reason to stick to the norm ball
    - Security is related to safety. Compromised systems aren't safe.
    - Security / worst case analysis is a way of guaranteeing safety. Safety in the worst case implies safety in general. (I'm getting less enthusiastic about this approach over time though: security may turn out to involve hiding flaws more than removing flaws, and in many cases there is a tradeoff between worst case and average case performance)
- AI Safety / Value alignment
    - The norm ball actually does model *the first few steps* of *incremental, gradient-based* reward maximization
    - What about more steps?
    - What about other search strategies?

Biggest limitation of threat model

- In "Explaining and Harnessing Adversarial Examples" I set up this game:
  - World samples an input point and label from the test set
  - Adversary perturbs point within the norm ball
  - Defender classifies the perturbed point
- Let's call this "expectimax norm ball" threat model

Expectimax is far from solved

- Expectimax norm ball defenses:
  - Tend to get ~50% accuracy even when they work (exception: MNIST)
  - Tend not to work on harder datasets (many approaches that work on CIFAR don't work on ImageNet)
  - Tend to work only for tiny norm ball (e.g. 8/255 is imperceptible)
  - Most are not provable, so maybe they break if we come up with a stronger attack
- Norm ball is a minuscule part of threat model space, so expectimax as a whole is even further from solved

True max rather than expectimax

- Suppose we got 99% accuracy in the expectimax setting
- Sample 100 points. In expectation 1 will be an error
- Attacker then repeats this 1 error forever
- Asymptotic accuracy is 0%
- Call this "test set attack" (Gilmer et al 2018)

Failed defenses: expectimax norm ball defenses

- Let r be rate of failure on naturally occurring data
- Adversarial training / certified robustness methods often *increase* r
- They have never driven r to zero

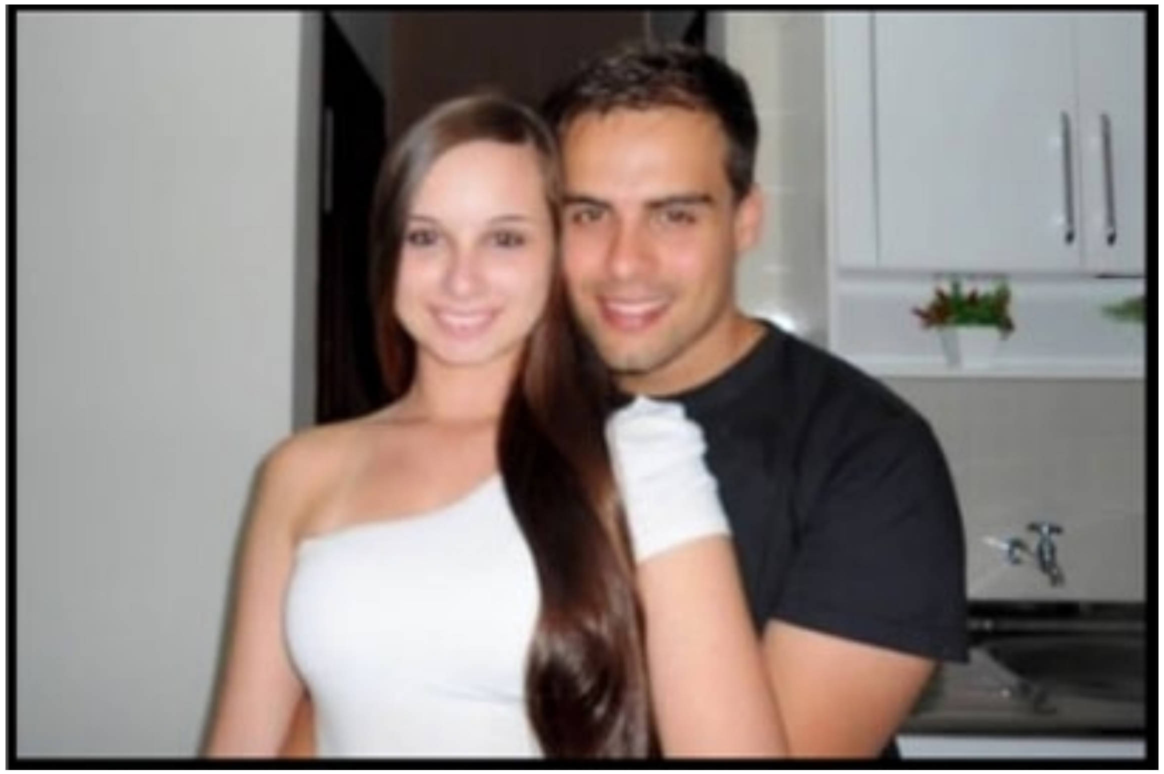
Failed defenses: traditional ML

- Gilmer et al 2018 identify the test set attack but use it to argue against studying ML security
  - They advocate reducing r
    - Asymptotic failure rate under attack is still 1 unless r reaches 0
  - They also advocate reducing *volume* of errors
    - As far as the test set attack is concerned, this is just a less direct way of reducing r

Every fixed defense is a sitting duck

- On some tasks, it's possible to just encode the true task directly, and then you can get r to 0
- On almost any real task, it's hard to imagine that we'll ever solve the task *truly perfectly for every weird input point*
- Attackers can just filter until they find failures

Fooling humans



Elsayed et al 2018

Elsayed et al 2018

If not deterministic, then... stochastic?

- Stochastic defenses are not totally broken for expectimax norm ball (Feinman et al 2017, Carlini and Wagner 2017)
- What about for true max?
- Suppose there exists an input such that the true class is not chosen by $\text{argmax}_{\text{class}}\ p_{\text{model}}(\text{class} \mid \text{input})$
- Then asymptotic rate of failure under test set attack is at least 0.5
  - Best outcome is when the true class is tied for argmax but not selected by argmax, and only one other class participates in the tie.
- Stochastic is best defense so far! But far from enough.

If not deterministic/stochastic, then... abstention?

- What if the classifier is allowed to abstain for some inputs?
  - Confidence thresholding
  - Other mechanisms for choosing when to abstain
- For a deterministic abstinence policy, this is just another way of reducing r
- *Can* reduce r to 0 by abstaining on every input
- Hard to imagine reaching r=0 with a low amount of deterministic abstention

If not deterministic/stochastic, then *dynamic*

- Use a different $p_{\text{model}}(\text{class}|\text{input})$ every time we process an input
- This breaks the standard train / infer distinction
- Requires dynamic behavior during deployment 😱

“Hello World” dynamic defense: memorization

- Memorize all inputs
- If an input has been seen before:
  - If allowed to abstain, abstain
  - If not allowed to abstain, return a random class

Memorization defense on naturally occurring data

- No reduction in accuracy for data that doesn't contain repeats (most academic settings)

- Unfortunately many practical settings contain repeats

Memorization defense under test set attack, with abstention

- Attacker can't get more than r error rate
- Attacker can cause asymptotic 100% abstention
- For some applications, abstaining on attacks is OK

Memorization defense under test set attack, no abstention

- For k classes attacker can cause asymptotic error rate of $(k\text{-}1)/k$
- However a *targeted* attacker also has a target miss rate of $(k\text{-}1)/k$
- At least makes relationship between attacker and defender symmetric

Caveats

- “Test set attack” and variants added in this paper are only “hello world” attacks. Much more sophisticated attacks in the dynamic setting remain to be developed

- “Memorization” is a “hello world” defense. Intended only to show existence of a dynamic defense that outperforms all fixed defenses against “test set attack”. Much more sophisticated attacks.

- I argue “dynamic models are necessary” not “dynamic models are sufficient”. Other mechanisms are needed too. Note that the best version of the memorization defense includes abstention.

Questions