Ironing Out the Squiggles

by **Zack_M_Davis** 29th Apr 2024

57

Adversarial Examples | Adversarial Training | Machine Learning (ML) | AI | World Modeling | Frontpage

Adversarial Examples: A Problem

The apparent successes of the deep learning revolution conceal a dark underbelly. It may seem that we now know how to get computers to (say) check whether a photo is of a bird, but this façade of seemingly good performance is belied by the existence of *adversarial examples*—specially prepared data that looks ordinary to humans, but is seen radically differently by machine learning models.

The differentiable nature of neural networks, which make them possible to be trained at all, are also responsible for their downfall at the hands of an adversary. Deep learning models are fit using stochastic gradient descent (SGD) to approximate the function between° expected inputs and outputs. Given an input, an expected output, and a loss function (which measures "how bad" it is for the actual output to differ from the expected output), we can calculate the gradient of the loss on the input—the derivative with respect to every parameter in our neural network—which tells us which direction to adjust the parameters in order to make the loss go down, to make the approximation better.[1]

But gradients are a double-edged sword: the same properties that make it easy to calculate how to adjust a *model* to make it better at classifying an image, also make it easy to calculate how to adjust an *image* to make the model classify it incorrectly. If we take the gradient of the loss with respect to the pixels of the image (rather than the parameters of the model), that tells us which direction to adjust the pixels to make the loss go down—*or up*. (The direction of steepest increase is just the opposite of the direction of steepest decrease.) A tiny step in that direction in imagespace perturbs the pixels of an image just so—making this one the tiniest bit darker, that one the tiniest bit lighter—in a way that humans don't even notice, but which completely breaks an image classifier sensitive to that direction in the conjunction of many pixel-dimensions°, making it report utmost confidence in nonsense classifications.

Some might ask: why does it matter if our image classifier fails on examples that have been mathematically constructed to fool it? If it works for the images one would naturally

encounter, isn't that good enough?

One might mundanely reply that gracefully handling untrusted inputs is a desideratum for many real-world applications, but a more forward-thinking reply might instead emphasize what adversarial examples imply about our lack of understanding of the systems we're building, separately from whether we pragmatically expect to face an adversary. It's a problem if we think we've trained our machines to recognize birds, but they've actually learned to recognize a squiggly alien set in imagespace that includes a lot of obvious non-birds and excludes a lot of obvious birds. To plan good outcomes, we need to understand what's going on, and "The loss happens to increase in this direction" is at best only the start of a real explanation.

One obvious first guess as to what's going on is that the models are overfitting. Gradient descent isn't exactly a sophisticated algorithm. There's an intuition that the *first* solution that you happen to find by climbing down the loss landscape is likely to have idiosyncratic quirks on any inputs it wasn't trained for. (And that an AI designer from a more competent civilization would use a principled understanding of vision to come up with something much better than what we get by shoveling compute into SGD.) Similarly, a hastily cobbled-together conventional computer program that passed a test suite is going to have bugs in areas not covered by the tests.

But that explanation is in tension with other evidence, like the observation that adversarial examples often generalize between models. (An adversarial example optimized against one model is often misclassified by others, too, and even assigned the same class.) It seems unlikely that different hastily cobbled-together programs would have the *same* bug.

In "Adversarial Examples Are Not Bugs, They Are Features", Andrew Ilyas *et al.* propose an alternative explanation, that adversarial examples arise from predictively useful features that happen to not be robust to "pixel-level" perturbations. As far as the in-distribution predictive accuracy of the model is concerned, a high-frequency pattern that humans don't notice is fair game for distinguishing between image classes; there's no rule that the features that happen to be salient to humans need to take priority. Ilyas *et al.* provide some striking evidence for this thesis in the form of a model trained exclusively on adversarial examples yielding good performance on the original, unmodified test set (!!).[2] On this view, adversarial examples arise from gradient descent being "too smart", not "too dumb": the program is fine; if the test suite didn't imply the behavior we wanted, that's our problem.

On the other hand, there's also some evidence that gradient descent being "dumb" may play a role in adversarial examples, in conjunction with the counterintuitive properties of high-dimensional spaces. In "Adversarial Spheres", Justin Gilmer *et al.* investigated a simple synthetic dataset of two classes representing points on the surface of two concentric n-dimensional spheres of radiuses 1 and (an arbitrarily chosen) 1.3. For an architecture yielding an ellipsoidal decision boundary, training on a million datapoints produced a network with very high accuracy (no errors in 10 million samples), but for which most of the axes of the decision ellipsoid were wrong, lying inside the inner sphere or outside the outer sphere—implying the existence of *on-distribution* adversarial examples (points on one sphere classified by the network as belonging to the other). In high-dimensional space, pinning down the exact contours of the decision boundary is bigger ask of SGD than merely being right virtually all of the time—even though a human wouldn't take a million datapoints to notice the hypothesis, "Hey, these all have a norm of exactly either 1 or 1.3."

Adversarial Training: A Solution?

Our story so far: we used gradient-based optimization to find a neural network that seemed to get low loss on an image classification task—that is, until an adversary used gradient-based optimization to find images on which our network gets *high* loss instead. Is that the end of the story? Are neural networks just the wrong idea for computer vision after all, or is there some way to continue within the current paradigm?

Would you believe that the solution involves ... gradient-based optimization?

In "Towards Deep Learning Models Resistant to Adversarial Attacks", Aleksander Madry *et al.* provide a formalization of the problem of adversarially robust classifiers. Instead of just trying to find network parameters θ that minimize loss L on an input x of intended class y, as in the original image classification task, the designers of a robust classifier are trying to minimize loss on inputs with a perturbation δ crafted by an adversary trying to maximize loss (subject to some maximum perturbation size ε):

$$\min_{\theta} \max_{||\delta|| < \varepsilon} L(\theta, x + \delta, y)$$

In this formulation, the attacker's problem of creating adversarial examples, and the defender's problem of training a model robust to them, are intimately related. If we change the image-classification problem statement to be about correctly classifying not just natural images, but an ε-ball around them, then you've defeated all adversarial examples up to that ε. This turns out to generally require larger models than the

classification problem for natural images: evidently, the decision boundary needed to separate famously "spiky" high-dimensional balls is significantly more complicated than that needed to separate natural inputs as points.

To solve the inner maximization problem, Madry *et al.* use the method of projected gradient descent (PGD) for constrained optimization: do SGD on the unconstrained problem, but after every step, project the result onto the constraint (in this case, the set of perturbations of size less than ε). This is somewhat more sophisticated than just generating any old adversarial examples and throwing them into your training set; the iterative aspect of PGD makes a difference.

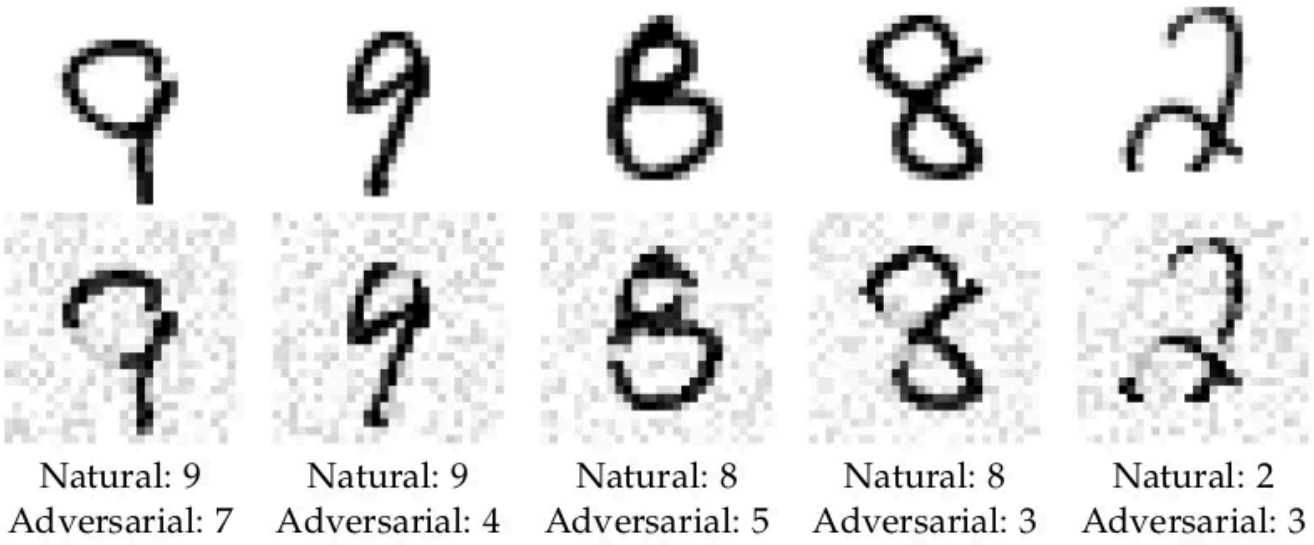
Adversarial Robustness Is About Aligning Human and Model Decision Boundaries

What would it look like if we succeeded at training an adversarially robust classifier? How would you know if it worked? It's all well and good to say that a classifier is robust if there are no adversarial examples: you shouldn't be able to add barely-perceptible noise to an image and completely change the classification. But by the nature of the problem, adversarial examples aren't machine-checkable. We can't write a function that either finds them or reports "No solution found." The machine can only optimize for inputs that maximize loss. We, the humans, call such inputs "adversarial examples" when they look normal to us.

Imagespace is continuous: in the limit of large ε, you can perturb any image into any other—just interpolate the pixels. When we say we want an adversarially robust classifier, we mean that perturbations that change the model's output should also make a human classify the input differently. Trying to find adversarial examples against a robust image classifier amounts to trying to find the smallest change to an image that alters what it "really" looks like (to humans).

You might wonder what the smallest such change could be, or perhaps if there even is any nontrivially "smallest" change (significantly better than just interpolating between images).

Madry *et al.* adversarially trained a classifier for the MNIST dataset of handwritten digits. Using PGD to search for adversarial examples under the ℓ_2 norm—the sum of the squares of the differences in pixel values between the original and perturbed images—the classifier's performance doesn't really tank until you crank ε up to around 4—at which point, the perturbations don't look like random noise anymore, as seen in Figure 12 from the paper:



Tasked with changing an image's class given a limited budget of how many pixels can be changed by how much, PGD concentrates its budget on human-meaningful changes—deleting part of the loop of a *9* to make a *7* or a *4*, deleting the middle-left of an *8* to make a *3*. In contrast to "vanilla" models whose susceptibility to adversarial examples makes us suspect their good performance on natural data is deceiving, it appears that the adversarially-trained model is seeing the same digits we are.

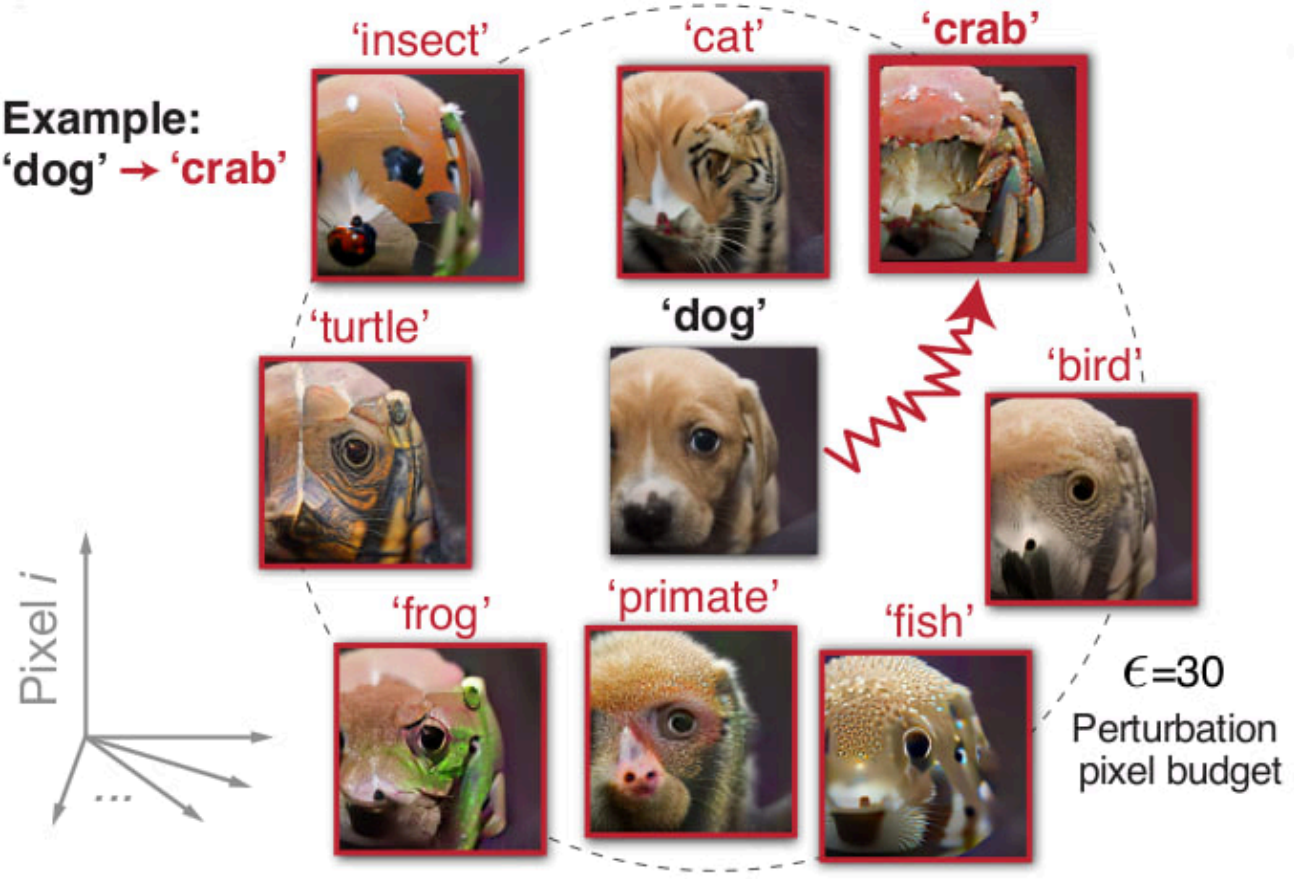
(I don't want to overstate the significance of this result and leave the impression that adversarial examples are necessarily "solved", but for the purposes of this post, I want to highlight the striking visual demonstration of what it looks like when adversarial training *works*.)[3]

An even more striking illustration of this phenomenon is provided in "Robustified ANNs Reveal Wormholes Between Human Category Percepts" by Guy Gaziv, Michael J. Lee, and James J. DiCarlo.[4]

The reason adversarial examples are surprising and disturbing is because they seem to reveal neural nets as fundamentally brittle in a way that humans aren't: we can't imagine our visual perception being so drastically effected by such small changes to an image. But what if that's just because we didn't know how to imagine the right changes?

Gaziv *et al.* adversarially trained image classifier models to be robust against perturbations under the ℓ_2 norm of ε being 1, 3, or 10, and then tried to produce adversarial examples with ϵ up to 30.[5] (For 224×224 images in the RGB colorspace, the maximum possible ℓ_2 distance is $\sqrt{3 \cdot 224^2} \approx 388$. The typical difference between ImageNet images is about 130.)

What they found is that adversarial examples optimized to change the robustified models' classifications also changed human judgments, as confirmed in experiments where subjects were shown the images for up to 0.8 seconds—but you can also see for yourself in the paper or on the project website. Here's Figure 3a from the paper:



The authors confirm in the Supplementary Material that *random* $\epsilon = 30$ perturbations don't affect human judgments at all. (Try squinting or standing far away from the monitor to better appreciate just how similar the pictures in Figure 3a are.) The robustified models are close enough to seeing the same animals we are that adversarial attacks against them are also attacks against us, precisely targeting their limited pixel-changing budget on surprising low-ℓ_2-norm "wormholes" between apparently distant human precepts.

Implications for Alignment?

Futurists have sometimes worried that our civilization's coming transition to machine intelligence may prove to be incompatible with human existence. If AI doesn't see the world the same way as we do°, then there's no reason for it to steer towards world-states that we would regard as valuable. (Having a concept of the right thing is a necessary if not sufficient° prerequisite for doing the right thing.)

As primitive precursors to machine intelligence have been invented, some authors have taken the capabilities of neural networks to learn complicated functions as an encouraging sign. Early discussions of AI alignment had emphasized that "leaving out just [...] one thing" could result in a catastrophic outcome°—for example, a powerful agent that valued subjective experience but lacked an analogue of boredom° would presumably use all its resources to tile the universe with repetitions of its most optimized experience. (The emotion of boredom is evolution's solution to the exploration–exploitation trade-off; there's no reason to implement it if you can just compute the optimal policy.)

The particular failure mode of "leaving one thing out" is starting to seem less likely on the current paradigm. Katja Grace notes that image synthesis methods have no trouble generating photorealistic human faces°. Diffusion models don't "accidentally forget" that faces have nostrils, even if a human programmer trying to manually write a face image generation routine might. Similarly, large language models obey the quantity-opinion-size-age-shape-color-origin-purpose adjective order convention in English without the system designers needing to explicitly program that in or even be aware of it, despite the intuitive appeal of philosophical arguments one could make to the effect that "English is fragile." So the optimistic argument goes: if instilling human values into future AGI is as easy as specifying desired behavior for contemporary generative AI, then we might be in luck?

But even if machine learning methods make some kinds of failures due to brittle specification less likely, that doesn't imply that alignment is easy. A different way things could go wrong is if representations learned from data turn out not to be robust off the training distribution. A function that tells your AI system whether an action looks good and is right virtually all of the time on natural inputs isn't safe if you use it to drive an enormous search for unnatural (highly optimized) inputs on which it might behave very differently.

Thus, the extent to which ML methods can be made robust is potentially a key crux for views about the future of Earth-originating intelligent life. In a 2018 comment° on a summary of Paul Christiano's research agenda, Eliezer Yudkowsky characterized one of his "two critical points" of disagreement with Christiano as being about how easy robust ML is:

> Eliezer expects great Project Chaos and Software Despair from trying to use gradient descent, genetic algorithms, or anything like that, as the basic optimization to reproduce par-human cognition within a boundary in great fidelity to that boundary

> as the boundary was implied by human-labeled data. Eliezer thinks that if you have any optimization powerful enough to reproduce humanlike cognition inside a detailed boundary by looking at a human-labeled dataset trying to outline the boundary, the thing doing the optimization is powerful enough that we cannot assume its neutrality the way we can assume the neutrality of gradient descent.
>
> Eliezer expects weird squiggles from gradient descent—it's not that gradient descent can never produce par-human cognition, even natural selection will do that if you dump in enough computing power. But you will get the kind of weird squiggles in the learned function that adversarial examples expose in current nets—special inputs that weren't in the training distribution, but look like typical members of the training distribution from the perspective of the training distribution itself, will break what we think is the intended labeling from outside the system. Eliezer does not think Ian Goodfellow will have created a competitive form of supervised learning by gradient descent which lacks "squiggles" findable by powerful intelligence by the time anyone is trying to create ML-based AGI, though Eliezer is certainly cheering Goodfellow on about this and would recommend allocating Goodfellow $1 billion if Goodfellow said he could productively use it. You cannot iron out the squiggles just by using more computing power in bounded in-universe amounts.

Christiano replied, in part°:

> For adversarial examples in particular, I think that the most reasonable guess right now is that it takes more model capacity (and hence data) to classify all perturbations of natural images correctly rather than merely classifying most correctly—*i.e.*, the smallest neural net that classifies them all right is bigger than the smallest neural net that gets most of them right—but that if you had enough capacity+data then adversarial training would probably be robust to adversarial perturbations. Do you want to make the opposite prediction?

At the time in 2018, it may have been hard for readers to determine which of these views was less wrong—and maybe it's still too early to call. ("Robust ML" is an active research area, not a crisp problem statement that we can definitively say is solved or not-solved.) But it should be a relatively easier call for the ArXiv followers of 2024 than the blog readers of 2018, as the state of the art has advanced and more relevant experiments have been published. To my inexpert eyes, the Gaziv *et al.* "perceptual wormholes" result does seem like a clue that "ironing out the squiggles" may prove to be feasible after all—that

adversarial examples are mostly explainable in terms of non-robust features and high-dimensional geometry, and remediable by better (perhaps more compute-intensive) methods—rather than being a fundamental indictment of our Society's entire paradigm for building AI.

Am I missing anything important? Probably. I can only hope that someone who isn't will let me know in the comments.

---

1. This post and much of the literature about adversarial examples focuses on image classification, in which case the input would be the pixels of an image, the output would be a class label describing the content of the image, and the loss function might be the negative logarithm of the probability that the model assigned to the correct label. But the story for other tasks and modalities is going to be much the same. ↩
2. That is, as an illustrative example, training on a dataset of birds-perturbed-to-be-classified-as-bicycles and bicycles-perturbed-to-be-classified-as-birds results in good performance on natural images of bicycles and birds. ↩
3. Madry *et al.* are clear that there are a lot of caveats about models trained with their methods still being vulnerable to attacks that use second-order derivatives or eschew gradients entirely—and you can see that there are still non-human-meaningful pixelly artifacts in the second row of their Figure 12. ↩
4. A version of this paper has also appeared under the less interesting title, "Strong and Precise Modulation of Human Percepts via Robustified ANNs". Do some reviewers have a prejudice against creative paper titles? While researching the present post, I was disturbed to find that the newest version of the Gilmer *et al.* "Adversarial Spheres" paper had been re-titled "The Relationship Between High-Dimensional Geometry and Adversarial Examples". ↩
5. Gaziv *et al.* use the script epsilon ε to refer to the size of perturbation used in training the robustified models, and the lunate epsilon ϵ to refer to the size used in subsequent attacks. I'm sure there's a joke here about sensitivity to small visual changes, but I didn't optimize this footnote hard enough to find it. ↩

Adversarial Examples 2 | Adversarial Training 2 | Machine Learning (ML) 2 | AI 2 | World Modeling 2 | Frontpage

You cannot comment at this time (Questions? Send an email to team@lesswrong.com)

2 comments, sorted by top scoring

[–] **faul_sname** 1h 2 0

> In "Adversarial Spheres", Justin Gilmer *et al.* investigated a simple synthetic dataset of two classes representing points on the surface of two concentric *n*-dimensional spheres of radiuses 1 and (an arbitrarily chosen) 1.3. For an architecture yielding an ellipsoidal decision boundary, training on a million datapoints produced a network with very high accuracy (no errors in 10 million samples), but for which most of the axes of the decision ellipsoid were wrong, lying inside the inner sphere or outside the outer sphere—implying the existence of *on-distribution* adversarial examples (points on one sphere classified by the network as belonging to the other).

One thing I wonder is whether real-world category boundaries tend to be smooth like this, for the kinds of categorizations that are likely to be salient. The categories I tend to care about in practice seem to be things like "is this business plan profitable". If you take a bunch of business plans, and rate them on a scale of -1 to +1 on a bunch of different metrics, and classify whether businesses following them were profitable vs unprofitable, In that case, I wouldn't particularly expect that the boundary between "profitable business plan" and "unprofitable-business-plan" would look like "an ellipsoidal shell centered around some prototypical ur-business-plan, where any business plan inside that shell is profitable and any business plan outside that shell is unprofitable".

[–] **Dagon** 2h 2 0

Just to focus on the underlying tension, does this differ from noting "all models are wrong, some models are useful"?

> an AI designer from a more competent civilization would use a principled understanding of vision to come up with something much better than what we get by shoveling compute into SGD

How sure are you that there can be a "principled understanding of vision" that leads to perfect modeling, as opposed to just different tradeoffs (of domain, precision, recall, cost, and error cases)? The human brain is pretty susceptible to adversarial (both generated illusion and evolved camoflage) inputs as well, though they're different enough that the specific failures aren't comparable.

Moderation Log