Announcements

(1) HW #1 graded on gradescope

(2) Reminder: project meetings, and HW #2

Adversarial Robustness

Deep nets achieve human-level accuracy on many image classification tasks

But are they robust?

The seminal work of Szegedy et al. introduced the notion of adversarial examples

> "For most images x, can find a perturbation Δ s.t. the deep net misclassifies $x+\Delta$"

The perturbation is often imperceptible to humans, e.g. $\|\Delta\|_\infty \leq 0.05$

Even worse, attacks can be

(1) targeted - can choose what it is misclassified as, appropriate choice of Δ

(2) black box - some attacks don't need to know the parameters of deep net they're attacking

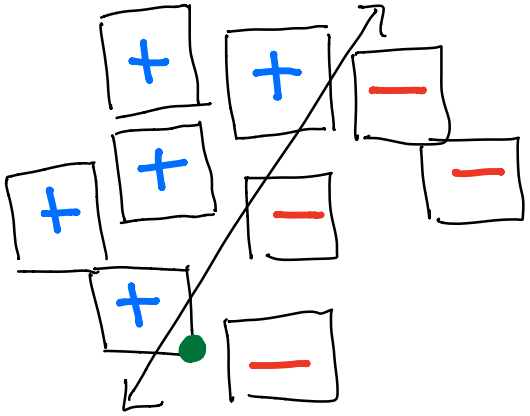
Main Question: Can we robustify deep learning?

Madry et al. introduced a popular defense based on adversarial training:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D} \left[\max_{\|\Delta\|_\infty \leq \delta} L(x+\Delta, y; \theta) \right]$$

Essentially this is robust optimization applied to model fitting

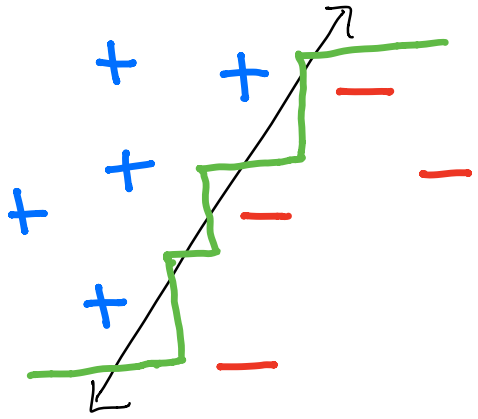
Some intuition: Let's start with linearly separable data



Around each point, draw a δ-ball in ℓ_∞

wherever the ball crosses the decision boundary, we get an adversarial example

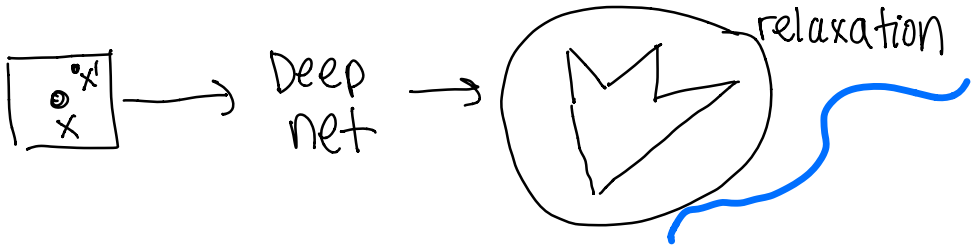
We can still find a robust decision boundary, but it becomes more complex



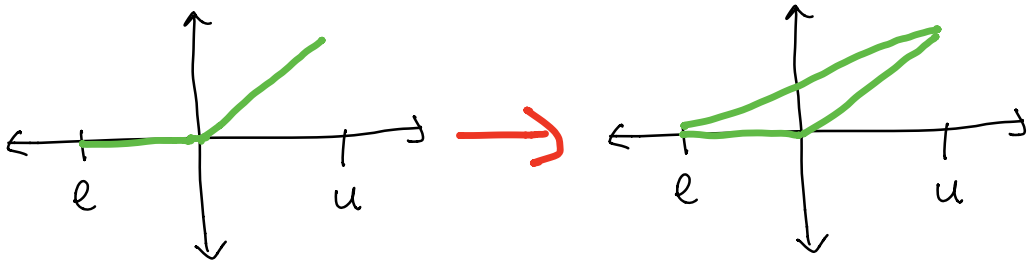
Nevertheless we use SGD to train

Adversarial training leads to significant robustness improvements, but generally at the cost of accuracy

Kolter and Wang gave certified defenses against adversarial examples via convex relaxations:



Their idea was to convexify the ReLU on a restricted domain:



to get an outer bound on

We'll study randomized smoothing, an approach for handling ℓ_2-perturbations

Lecuyer et al were the first to give rigorous guarantees

The main ideas were

(1) Show that any differentially private classifier with a margin is provably robust

(2) Add a layer of noise to make to classifier differentially private

Here we mean differential privacy w.r.t. the pixel values

First we need a generalization of our earlier definition of privacy

def: Let ρ be a metric on data bases. We say a randomized algorithm A is (ε, δ)-differentially private w.r.t. ρ if $\forall$ databases m, m' with $\rho(m, m') \leq 1$

$$\mathbb{P}(A(m) \in S) \leq e^{\varepsilon} p(A(m') \in S) + \delta$$

$\forall\ S \subseteq O$ ← output space

Remember, for us databases ≡ images

Now consider a deep net $f: \mathbb{R}^d \to [k]$

set of labels

Most deep nets compute a score function

$$Q: \mathbb{R}^d \to [0,1]^k$$

where the coordinates of $Q(x)$ are interpreted as probabilities assigned to each label, and $\sum_{i=1}^{k} Q(x)_i = 1$

Then $f(x) = \underset{i \in [k]}{\operatorname{argmax}}\ Q(x)_i$

i.e. it outputs the "most probable" label

The main proposition is:

randomized scoring function

Proposition: Suppose A^K is (ε, δ)-DP w.r.t. $p \equiv \ell_p$. Then for any input x, if for some $i \in [K]$ we have

$$\underbrace{\mathbb{E}[A_i(x)]}_{Q(x)_i} > e^{2\varepsilon} \max_{j \neq i} \mathbb{E}[A_j(x)] + (1 + e^{\varepsilon})\delta$$

then $f(x) = \operatorname{argmax}_{i \in [K]} \mathbb{E}[A_i(x)]$ is robust to attacks Δ with $\|\Delta\|_p \leq 1$ on x

Note: the classification rule is deterministic, because it's based on expectation of A

First we will prove a helper lemma that says:

"If A is differentially private, then so is its expectation"

In particular:

Lemma 1: Suppose A is a randomized function with outputs in $[0,b]$ that satisfies (ε,δ)-DP w.r.t. $P \equiv \ell_p$. Then

$$\mathbb{E}[A(x)] \leq e^{\varepsilon}\mathbb{E}[A(x+\Delta)] + b\delta$$

$\forall\ \Delta$ with $\|\Delta\|_p \leq 1$

Proof: We can write

$$\mathbb{E}[A(x)] \leq \int_0^b \mathbb{P}[A(x) \geq t]\, dt$$

Now invoking the definition of DP

$$\leq e^{\varepsilon}\left(\int \mathbb{P}[A(x+\Delta) \geq t]\, dt\right) + \int_0^b \delta\, dt$$

$$= e^{\varepsilon}\mathbb{E}[A(x+\Delta)] + b\delta$$

$\square$

Now we are ready to prove the main proposition

Proof: Consider any Δ with $\|\Delta\|_p \leq 1$ and let $x' = x + \Delta$.

Then applying Lemma 1 coordinate wise, we have:

$$\mathbb{E}[A_i(x)] \leq e^{\varepsilon} \mathbb{E}[A_i(x')] + \delta \quad (1)$$

$$\mathbb{E}[A_i(x')] \leq e^{\varepsilon} \mathbb{E}[A_i(x)] + \delta \quad (2)$$

Now we will use (1) + (2) to give a lower bound on the score of label i on input x', and an upper bound for every other label:

$$\mathbb{E}[A_i(x')] \overset{(1)}{\geq} \frac{\mathbb{E}[A_i(x)] - \delta}{e^{\varepsilon}}$$

Using the hypothesis of the prop., we have:

$$> \frac{e^{2\varepsilon} \max_{j \neq i} \mathbb{E}[A_j(x)] + (1+e^{\varepsilon})\delta - \delta}{e^{\varepsilon}}$$

$$= e^{\varepsilon} \max_{j \neq i} \mathbb{E}[A_j(x)] + \delta$$

$$\overset{(2)}{\geq} \max_{j \neq i} \mathbb{E}[A_j(x')]$$

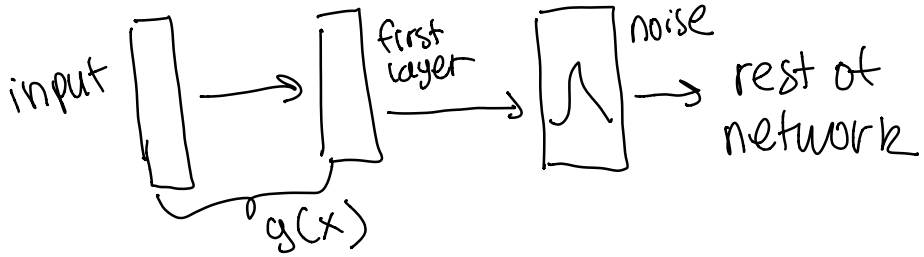
Thus we have

$$\mathbb{E}[A_i(x')] > \max_{j \neq i} \mathbb{E}[A_j(x')]$$

which means we still output label i. ☒

But what do we do if the randomized scoring function isn't DP?

Let's add a noise layer, e.g.



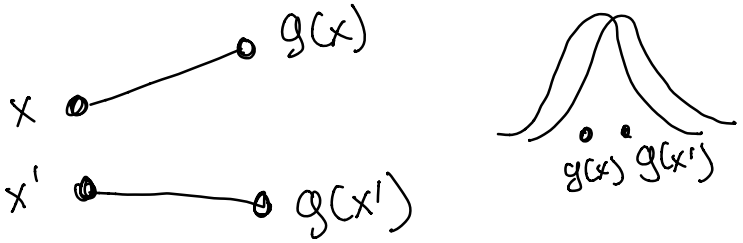
You could add it at the input layer, but Lucuyer et al. find that this is better

Main Question: How much noise should we add?

It's driven by the sensitivity, e.g.

$$N_{p,q} = \max_{x \neq x'} \frac{\|g(x) - g(x')\|_q}{\|x - x'\|_p}$$

Intuitively, the larger the sensitivity, the more noise you have to add



Note: If the weights in the first layer are W, and there is no nonlinearity, we have

$$N_{p,q} = \|W\|_{p,q} \triangleq \sup_{\|x\|_p \leq 1} \|Wx\|_q$$

Some p,q-norms are hard to compute, and they use relaxations instead

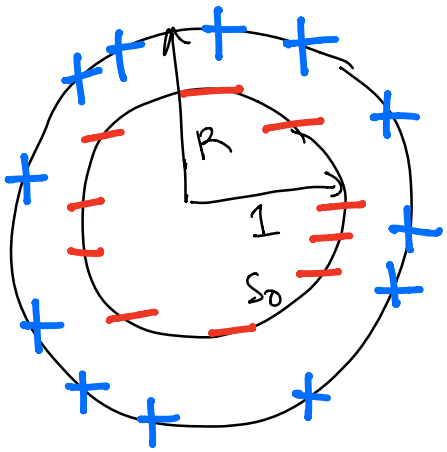
Can also train the network in a manner that makes it less sensitive

Epilogue: Yang et al. showed how optimal randomized smoothing schemes come from Wulff crystals

Interlude

Main Question: Are there situations where adversarial examples are inevitable?

Gilmer et al studied "adversarial spheres", where data is uniform on concentric spheres



They considered the following measure of adversarial robustness:

Let $E = \{ x \mid \text{model makes a mistake on } x \}$

Let $d(E) = \mathop{\mathbb{E}}_{x} [\text{dist}(x, E)]$

i.e. average Euclidean distance to the nearest error

They showed:

Theorem For any model that achieves accuracy $\frac{1}{2} \leq p < 1$ on S_0, then

$$d(E) \leq O\left(\frac{\Phi^{-1}(p)}{\sqrt{d}}\right)$$

where d is the dimension, Φ^{-1} is the inverse normal c.d.f.

The main idea is to use concentration of measure, e.g.

Theorem: Let $E \subseteq S_0$ be measurable, with $\mu(E) \geq \frac{1}{2}$. Let E_δ be the δ-neighborhood, i.e.

$$E_\delta = \{x \in S_0 \mid \exists y \in E \text{ with } \|x-y\|_2 \leq \delta\}$$

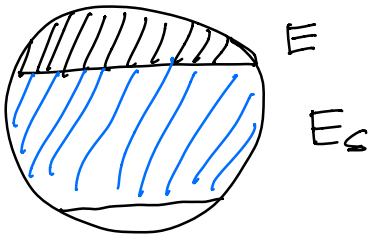
Then we have

$$\mu(E_\delta) \geq 1 - 2e^{-\frac{d\delta^2}{2}}$$

In particular, if $\delta = \Theta(\frac{1}{\sqrt{d}})$, we cover almost all of the sphere

Note: There are versions that start from the assumption $\mu(E) = p$, as in our application

The key intuition is that, subject to $\mu(E) = p$, the set that maximizes $d(E)$ is a spherical cap



And by concentration of measure, most of the sphere is within distance $O(\frac{1}{\sqrt{d}})$

Does this explain why being robust to adversarial examples is difficult in practice?

Likely not!

Theorem [informal]: There are accurate classifiers for natural images that are robust to adversarial examples

Proof: Humans 

Bubeck, Price and Razenshteyn showed:

Theorem [informal] If there exist robust classifiers, there is an exp. time algorithm to find one using just polynomially many samples

But efficiently finding one can be hard!

In particular, they give an instance where:

(1) robust learning is information-theoretically possible

(2) there is a computationally efficient non-robust learner

(3) No SQ algorithm can find a robust classifier

Note: What makes this problem diff. than the usual ones in agnostic learning is that the notion of robustness has to do with perturbations of the input, not label corruptions

Follow-up work of Bubeck, Lee, Price and Razenshteyn showed cryptographic hardness, rather than just SQ lower bounds

Taking a Step Back

Main Question: what makes a classifier robust in real world applications?

Adversarial robustness started off with ℓ_∞ perturbations, but could it be neither necessary nor sufficient?

Many reasons for wanting to be robust come from shifts in the distr.

Data Poisoning Attacks

Lastly, we consider another threat model:

"If an adversary can corrupt an ε-fraction of our training data, can we learn a good model?"

These questions were first posed in robust statistics

A canonical problem of robustly learning a Gaussian:

(1) we take N samples from $\mathcal{N}(\mu, \sigma^2)$

(2) An adversary is allowed to corrupt an ε-fraction arbitrarily

Goal: Learn an accurate $\hat{\mu}$, and $\hat{\sigma}$ in spite of the corruptions

Theorem [folklore]: Given ε-corrupted samples from $N(\mu, \sigma^2)$, there is an estimator

$$\hat{\mu} \equiv \text{empirical median}$$

$$\hat{\sigma} \equiv \frac{\text{empirical median abs. deviation}}{\Phi^{-1}(3/4)}$$

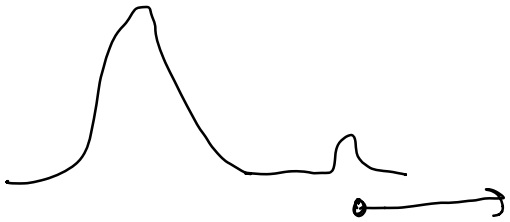
(median abs. deviation: median of deviations from $\hat{\mu}$)

that w.h.p satisfies

$$d_{TV}\left(N(\mu, \sigma^2), N(\hat{\mu}, \hat{\sigma}^2)\right) \leq 3\varepsilon$$

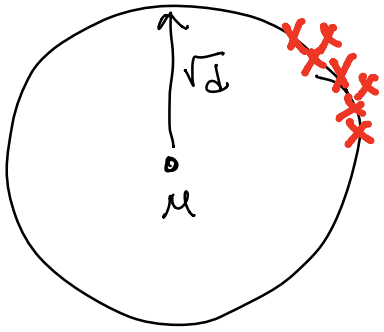
Thus an ε-fraction of corruptions cannot bias the learner too much!

Intuitively, the reason this works is



As you send the ε-bump to ∞, the empirical mean / variance diverge but the median and MAD don't

In high-dimensions, things are more tricky



Just an ε-fraction of corruptions can move the model parameters by $\varepsilon\sqrt{d}$

There are approaches based on considering all 1-d projections, e.g.

def: The Tukey depth of point x among a sample set S is

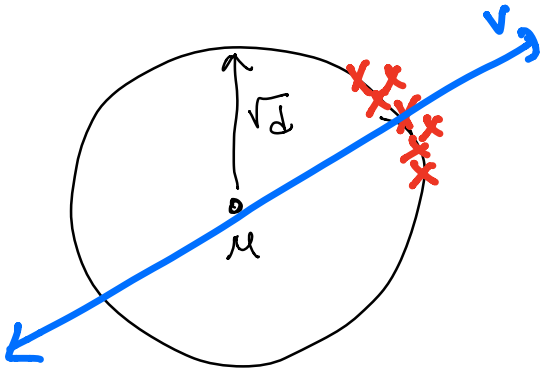
$$\min_{\text{1-d projs. } \Pi_v} \min\left(|\{y \in S : \Pi_v y < \Pi_v x\}|, \dots \right)$$

i.e. the worst case over all 1-d projs. of the fewest points to left or right

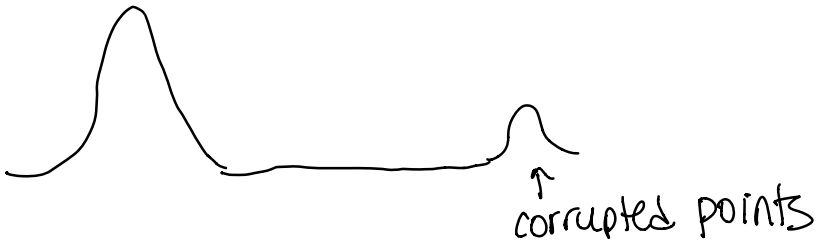
def: The Tukey median of a sample set S is the point x (not necc. in S) that maximizes Tukey depth

Now it is easy to see that the Tukey median would be robust if we could compute it

For example, consider



Projecting onto v, we'd get:



Unfortunately:

Theorem [Johnson, Preparata] The Tukey median is NP-hard to compute in high-dimensions

Nevertheless we can get around this

Diakonikolas, Kamath, Kane, Li, Moitra and Stewart showed

Theorem: Given ε-corrupted samples from a high-dimensional Gaussian $N(\mu, \Sigma)$ there is a poly time algorithm with poly. sample complexity that outputs $\hat{\mu}, \hat{\Sigma}$ with

$$d_{TV}(N(\mu, \Sigma), N(\hat{\mu}, \hat{\Sigma})) \leq O(\varepsilon \log^{3/2} 1/\varepsilon)$$

Lai, Rao and Vempala gave related results independently

Tran, Li, Madry showed how to use these results to defend against data poisoning attacks in deep nets:

(1) Train a deep network

(2) For each label $i \in [k]$, robustly compute the covariance of

$$\{x \mid f(x) = i,\ x \in S\}$$

and filter out unlikely x's using $\hat{\Sigma}$

They called these patterns in poisoned data spectral signatures

Intuition: Even when data is not Gaussian, looking for non-Gaussian projections can reveal interesting structure

Applications to Interpretability

Koh and Liang applied another

formalism from robust statistics, called the influence curve

> "the change in an estimator from changing a single point"

They studied how a model's prediction is influenced by up/down weighting a sample in the training set

Applications

1. when you get a point wrong, can you blame some of your training data?
2. Are you making a prediction because of meaningful similarities?

E.g. is your prediction most influenced by accidents (i.e. similar lighting / background) or the actual content