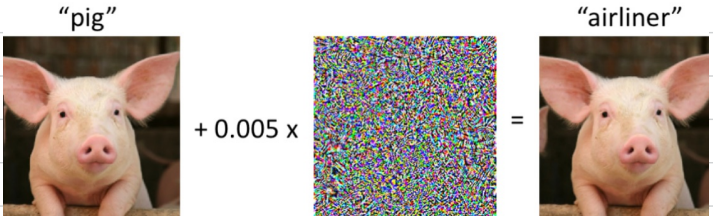
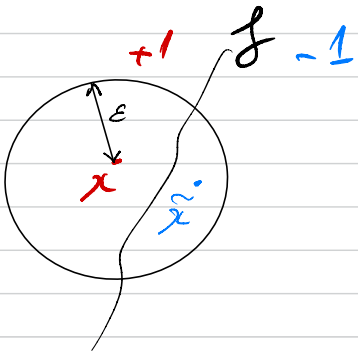
Adversarial Machine Learning

Adversarial Example:



- Classification problem $(X, Y) \sim D$; $y \in \{-1, 1\}$
  - Standard Risk:

$$Z(f) = \mathbb{E}_D[\mathbb{1}_{f(x) \neq y}] = \mathbb{P}_D[f(x) \neq y]$$



  - **Adversarial Risk:**

$$R_\varepsilon(f) = \mathbb{E}_D\left[\max_{\hat{x},\, \|x - \hat{x}\| \leq \varepsilon} \mathbb{1}_{f(\tilde{x}) \neq y}\right]$$

Questions?

- How should we define the adversary power?
  $l_2, l_\infty, l_1, l_0 \ldots$
  = Threat model (what an adversary can do)
- If $Z(f) \leq \delta$, How large can $R_\varepsilon(f)$ be?
- How can I design classifier so that it is robust?
- Given a non-robust classifier, can we make it robust?
- How can we find adversarial example.
- Which access we have on the NN to attack it.
- Why NN are non-robust?

Adversarial example:

White Box Attacks

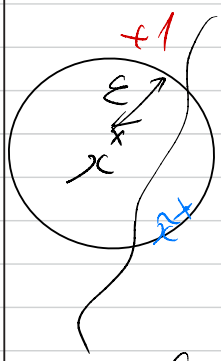
Task: Given $\mathcal{X}, \mathcal{Y}, f$ and $x \in \mathcal{X}$, a metric, and ε-Budget

→ find an adversarial input $\tilde{x}$ such that $\|x - \tilde{x}\| \leq \varepsilon$

$\hookrightarrow f(\tilde{x}) \neq y$

- Separable Data, $\exists h : \mathcal{X} \to \{-1, 1\}$ s.t. $h(x) = y$
  - Given x, if $f(x) \neq y \Rightarrow x = \tilde{x}$

    let's assume that $f(x) = y$



1) Ideally: $B_\varepsilon(x) \cap \{f(x) = -1\} \neq \emptyset$

2) Use gradient descent (gradient wrt input)

- we assume, we have $g \in [0,1]$

  $(g(x) = \mathbb{P}(y = 1 \mid x))$ $\quad f(x) = \begin{cases} +1 & \text{if } \frac{1}{2} \leq g(x) \leq 1 \\ -1 & \text{o.w.} \end{cases}$

- Wrong direction to follow will be: $\to h(x) \nabla_x g(x)$

- Smooth loss fn classif. $\ell(y \cdot g(x))$

$$\frac{d}{dx} \ell(y \cdot g(x)) = \underbrace{\ell'(y \cdot g(x))}_{\leq 0} \cdot y \cdot \nabla_x g(x) \quad \text{(loss is } \downarrow\text{)}$$

(Max. loss) → dir. to follow $\propto$ $\boxed{-y \nabla_x g(x)}$

- $h(x) \nabla_x g(x) \in \left[\begin{matrix} 1 \\ \\ d \end{matrix} \right]$

$$\Rightarrow \tilde{x} = x - \varepsilon h(x) \frac{\nabla_x g(x)}{\|\nabla_x g(x)\|_2}$$ under ℓ_2 const

optimal local updt.

Taylor:

$b(\tilde{x}) \sim b(x) + \nabla_x b(x)^T (\tilde{x} - x)$

$\rightarrow \max_{\|x - \tilde{x}\| \leq \varepsilon} b(\tilde{x}) \Leftrightarrow \max_{\|\tilde{x} - x\| \leq \varepsilon} \nabla_x b(x)^T (\tilde{x} - x)$

$\Leftrightarrow \max_{\|v\| \leq \varepsilon} \nabla_x b(x)^T v$

$\Rightarrow \quad v^* = \dfrac{\nabla_x b(x)}{\|\nabla_x b(x)\|} \varepsilon$

Apply this to $b(x) = -y\, g(x)$

Rq Sometimes we do multiple steps of this procedure.

$\Rightarrow$ this is called white box attack!

Black Box Attack:

- $x \rightarrow \boxed{g} \rightarrow g(x)$ $\Rightarrow$ we can approx $\nabla_x g(x)$ using finite difference.

$\Rightarrow g(x + \delta e_i) - g(x) \sim \delta \nabla_i g(x) \rightarrow \nabla_x g(x)$

- If we only have $\{+1, -1\}$: Build $S = \{x_i, y_i \text{ (output)} \mid y_i = f(x_i)\}$
  - $\rightarrow$ use S to train a different N.N
    - $\rightarrow$ Attack this NN with white Box Attack
    - $\rightarrow$ So very often we can transfer it to the unknown network

Attack of a physical object

(i.e. Stop sign)

Figure 2: To stop or not to stop.

• Robust vs Non-Robust features:

$\tilde{x} \xrightarrow{T} x \in \mathbb{R}^D$

$$x_1 = y + z_1$$

$$x_2 = y\sqrt{\frac{\log d}{d-1}} + z_2 \qquad z_i \sim \mathcal{N}(0,1)$$

$$\vdots$$

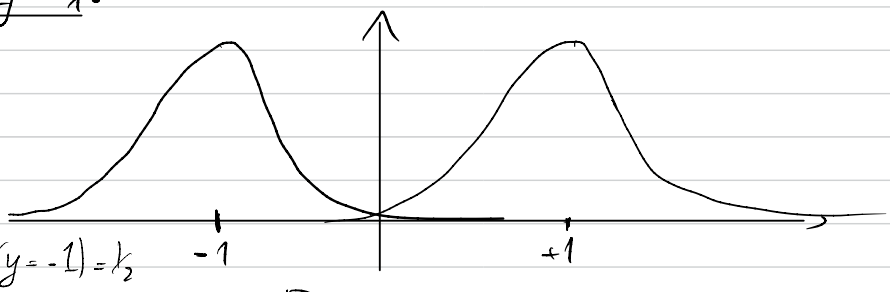
$$x_d = y\sqrt{\frac{\log d}{d-1}} + z_d$$

("SNR")

x_1 has a very high signal

$(x_2, \dots, x_d)$ has a very low signal (but we have a lot of them)

• Only using x_1:



$\mathbb{P}(y=1) = \mathbb{P}(y=-1) = \frac{1}{2}$

$$\Rightarrow \mathbb{P}(y \mid x_1) = \frac{\mathbb{P}(x_1 \mid y)\,\mathbb{P}(y)}{\mathbb{P}(x_1)}$$

$$\Rightarrow \max_{y \in \{-1,1\}} \mathbb{P}(x_1 \mid y)$$

• How often will you be wrong?

prob of making a mistake $\sim 0{,}16$

- Using all components:

$$\underset{\hat{y}\in\{-1,1\}}{\operatorname{argmax}}\ \mathbb{P}(\hat{y}\mid x) = \underset{\hat{y}}{\operatorname{argmax}}\ \frac{\mathbb{P}(x\mid\hat{y})\,\mathbb{P}(\hat{y})}{\mathbb{P}(x)}$$

$$= \underset{\hat{y}}{\operatorname{argmax}} \prod_{i=1}^{d} \mathbb{P}(x_i\mid\hat{y})$$

$$= \underset{\hat{y}}{\operatorname{argmax}} \prod_{i=1}^{d} \log\left(\frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}(x_i-\hat{y}\alpha_i)^2}\right)$$

$$= \underset{\hat{y}}{\operatorname{argmin}} \sum_{i=1}^{d} (x_i-\hat{y}\alpha_i)^2$$

$$= \underset{\hat{y}}{\operatorname{argmax}}\ \hat{y}\sum_{i=1}^{d} x_i\alpha_i$$

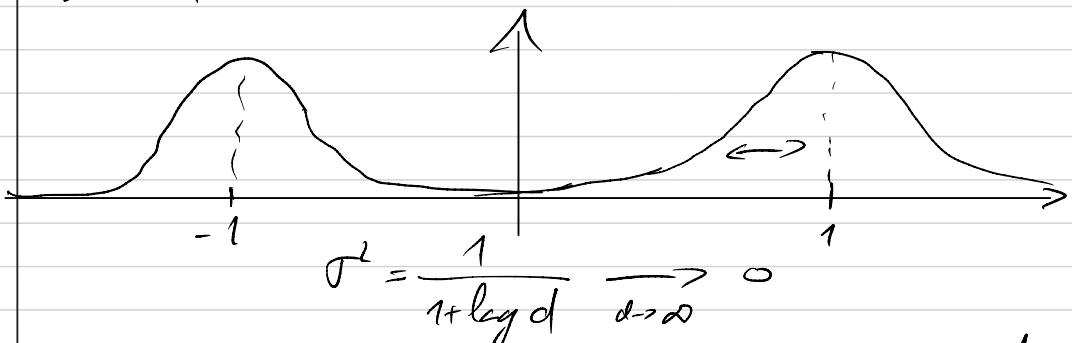
$$= \underset{\hat{y}}{\operatorname{argmax}}\ \hat{y}\,y\left(\sum_{i=1}^{d}\alpha_i^2\right) + \hat{y}\sum_{i=1}^{d} z_i\alpha_i$$

$$= \underset{\hat{y}}{\operatorname{argmax}}\ \hat{y}\,y\,(1+\log d) + \hat{y}\,Z \qquad Z\sim\mathcal{N}(0,1+\log d)$$

$\alpha_1 = 1$, $\alpha_i = \sqrt{\frac{\log d}{d-1}}$, $i>1$

rescaling $\rightarrow$ $\underset{\hat{y}}{\operatorname{argmax}}\ y\hat{y} + \hat{y}Z$ with $Z\sim\mathcal{N}\left(0,\frac{1}{1+\log d}\right)$

So it's equivalent;



$$\sigma^2 = \frac{1}{1+\log d} \xrightarrow[d\to\infty]{} 0$$

d: If we train well we can expect to have a standard error close to 0 when d grows.

• adversarial Risk :

- Assume we allow an adversary to move every component by ε. where $\varepsilon = 2\sqrt{\frac{\log d}{d-1}}$
  - what can the adversary do?
    - First, learn the concept.
    - then set $\tilde{x}_i = x_i - y\,2\sqrt{\frac{\log d}{d-1}}$

      $\tilde{x}_i = -y\sqrt{\frac{\log d}{d-1}} + z_i$
  - The optimal classifier will predict $-y$ whereas the true label was y.
    → Adversarial Risk = 1

→ We could have only considered x_1 and we still get a risk of 0,16

d There is a trade off between having a small :
- adv Risk
- Standard Risk

Summary

Standard Risk : $\mathcal{L}(f) = \mathbb{E}_{x,y\sim p}[\mathbb{1}_{f(x)\neq y}]$.

Adversarial example : $\tilde{x}$ s.t. $\|x-\tilde{x}\| \leq \varepsilon$
- $f(\tilde{x}) \neq y$

Adversarial Risk: $R_\varepsilon(f) = \mathbb{E}_{x,y\sim p}\left[\max_{\tilde{x}, \|x-\tilde{x}\|\leq\varepsilon} \mathbb{1}_{f(\tilde{x})\neq y}\right]$

→ It is possible to have $\mathcal{L}(f) \ll 1$ and $R_\varepsilon(f) = 1$

Why are ML algo vulnerable to attack?

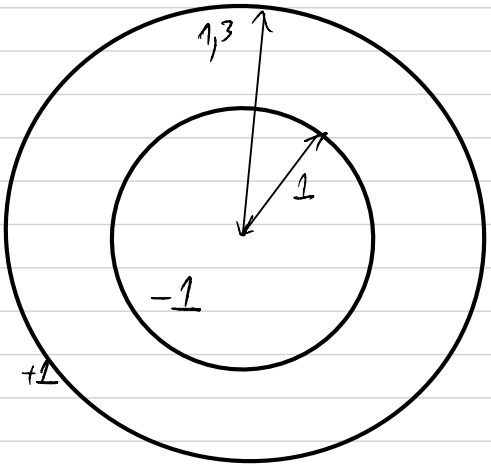
- Robust vs Non-Robust features

$$d+1 \begin{cases} x_1 = y + z_1 \\ x_2 = \alpha y + z_2 \\ \vdots \\ x_{d+1} = \alpha y + z_d \end{cases} \qquad \alpha = \sqrt{\frac{\log d}{d-1}}, \quad z_i \sim \mathcal{N}(0,1)$$

conclusion :
- If we use all features, we can make $\mathcal{L}(f) \xrightarrow{d\to\infty} 0$. But if we consider l_∞ perturbation $\varepsilon = 2\alpha$ then $R_\varepsilon(f) \to 1$
- If we use only the first feature of x, $\mathcal{L}(f) = R_\varepsilon(f) = 0,16$

Adversarial Sphere:

$D = 500$, $X \in \mathbb{R}^D$, $y \in \{\pm 1\}$



X is uniform on the sphere

⚠ D is very large

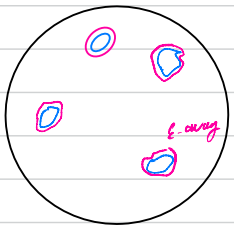
Experiment: $\cdot |S_{Train}| = 50$ Million, NN 2 hidden layers, ReLU act. 500 hidden nodes $\rightarrow f$

$\cdot |S_{Test}| = 20$ Million $\rightarrow$ No error on the test set !!! $\Rightarrow$ $\mathcal{L}(f)$ is very small

But the Bayes. Risk $R_\varepsilon \sim 1$ for $\varepsilon = \frac{1}{\sqrt{D}}$

• Effect of high dimension:

X is uniform



E: set of $x \in \mathcal{U}$, $\|x\| = 1$ and $f(x) = 1$ "error"

- Standard Risk:

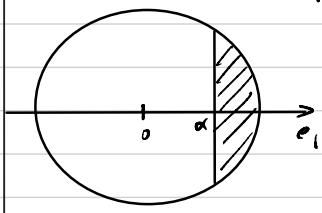
$\cdot \mathcal{L}(f) \sim \mathbb{P}(\|x\| = 1, f(x) = 1) = \frac{\text{Area of } E}{\text{Area of The Sphere}}$

- Adversarial Risk:

$R_\varepsilon(f) \sim \frac{\text{Area in } \varepsilon\text{-away}}{\text{Area of The Sphere}}$

Goal: Show that $\mathcal{L}(f) \ll 1$ but $R_\varepsilon(f) \sim 1$

- **E: spherical cap:**



$E = \{x, \|x\| = 1, x_1 > \alpha\}$; $\alpha \in [0,1]$

Probability of error p ; $0 \leq p \leq 1/2$

i) $\alpha = 0$ $\quad p = 1/2$

ii) $\alpha = 1$ $\quad p = 0$

→ $\alpha = f(p)$? intuition: proportional.

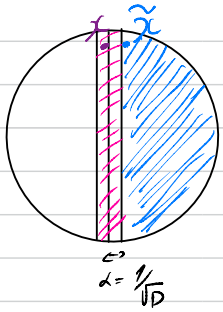
- p is very small, D is very large $\quad \boxed{\alpha = \frac{c(p)}{\sqrt{D}}}$

→ whatever p, in large dim, the cap is almost half the sphere.
⇒ all the mass you have on the surface of the sphere is on the equator !!

- p small, D large

$\varepsilon = 2\alpha = O\left(\frac{1}{\sqrt{D}}\right)$

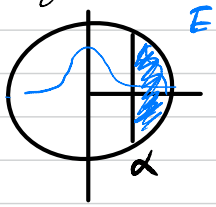
$\|x - \tilde{x}\| \leq \varepsilon$



$E\ \alpha = \frac{1}{\sqrt{D}}$

$\chi(f) = p$ (small) , $R_\varepsilon(f) = 1 - p \sim 1$

Proof:



simplif. $X \sim \text{Unif}(\text{sphere})$

$X \sim \mathcal{N}(0, \sigma^2 I)$; $\sigma^2 = 1/D$

X is spherical, $\|X\| \sim 1$ with high prob.

$p = \mathbb{P}(x_1 > \alpha) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_\alpha^{+\infty} e^{-x_1^2/2\sigma^2} dx_1$

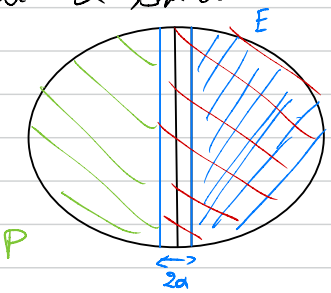
$= Q(\alpha\sqrt{D})$

where $Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$

$\Rightarrow \boxed{\alpha = \frac{Q^{-1}(p)}{\sqrt{D}}}$ ⇒ all the mass is on the equator

D large p small α small

let x uniform on the sphere with proba $1-p$, T_{wist} to be on E.



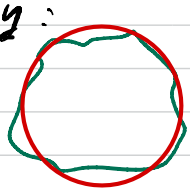
- What is the proba of obtaining a point ε-away of the cap?

$$\varepsilon = 2\alpha \sim \frac{1}{\sqrt{D}}$$

(Standard risk = p)

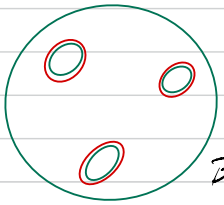
The adversarial risk is $R(f) = 1 - p \sim 1$

- **Isoperimetric inequality :**



- The fixed volume body with the smallest surface area is the sphere

In our case :



given a shape, we can add points ε away
→ the spherical cap is the shape which minimizes the total area with a fixed initial area.

By assuming the spherical cap, we have a lower bound on the adversarial risk.

Adversarial training:

$\theta = \min \mathcal{L}(f_\theta)$, But we don't want small $\mathcal{L}$ we want small R_ε

$\Rightarrow \theta = \min R_\varepsilon(f_\theta)$

Pb: • We don't know D $\Rightarrow$ we approx. the true risk by a sample avg.

• The loss function is not smooth $\Rightarrow$ approx. by a smooth loss

i.e $g(x) = \mathbb{P}(y=1|x)$; $f(x) = \begin{cases} +1 \text{ if } g(x) \in [1/2, 1] \end{cases}$

$\rightarrow$ we can consider the loss $\ell(y, x, g) = 1/2 - y(g(x) - 1/2)$

$\left.\begin{array}{l} \text{if } y=1 \rightarrow \ell = 1 - g(x) \\ \text{if } y=-1 \rightarrow \ell = g(x) \end{array}\right\}$ prob. of wrong pred.

Goal:

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{N} \sum_{i=1}^{N} \max_{\tilde{x}_i, \|x_i - \tilde{x}_i\| \leq \varepsilon} \ell(y_i, \tilde{x}_i, \theta)$$

How: At θ_t: 1) For every i, find $\tilde{x}_i = \arg\max_{\|x_i - \tilde{x}_i\| \leq \varepsilon} \ell(y_i, \tilde{x}_i, \theta)$ (adv. example) (max)

2) Compute grad: $\nabla_\theta \ell(y_i, \tilde{x}_i, \theta)$

and do a grad. step (min)

$\Rightarrow$ this is called **adversarial Training**, and obtains state of the art result for robust classifier.

Randomized Smoothing:

- How to make a given classifier robust?
- Let $f: \mathbb{R}^d \to \mathcal{Y}$ a given classifier.

$\Rightarrow$ Then consider g s.t. $g(x)$ will be the class the most likely returned by f when evaluated at $x+Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I)$

- g is the <u>smooth classifier</u>

$\Rightarrow$ consider x and y, we assume $g(x) = +1$

For $\tilde{x}$ s.t $\|x - \tilde{x}\| \leq \varepsilon$, we see that $g(\tilde{x}) = +1$

Robust Training: an equivalent method is to consider a ball of radius $\sqrt{D}\sigma$ (with a uniform dist.)

- <u>Idea</u>: . consider the binary case, and original classifier f.
  Take a point X, and let's say that label $y=1$ is most likely for $X+Z$.
  . now, if we take $\tilde{X}+Z$, where $\|X - \tilde{X}\|_2 \leq \varepsilon$ for some small ε.

It is intuitive that in average we are still most likely to return $y=1$ rather than $y=-1$
$\to$ i.e we will return the same label for points not too far away.

<u>Example</u>: $D=1$,

- let $p = \mathbb{E}[\mathbb{1}_{\{f(x+Z)=1\}}]$ and assume $\frac{1}{2} < p \leq 1$
  $\Rightarrow$ most of the points in the neighborhood of x are given $y=1$ rather than $y=-1$

now $\tilde{p} = \mathbb{E}[\mathbb{1}_{\{f(\tilde{x}+Z)=1\}}]$ when $\|x - \tilde{x}\| \leq \varepsilon$

assume we move x to the right by ε, How small can $\tilde{p}$ be?

$\to$ in the worst case all the points labeled $y=-1$ would be on the right of x in the tail of the gaussian. More precisely, the worst case happens if all the points to the left of $x + \sigma Q^{-1}(1-p)$ are labeled $y=1$ and all the points to the right $y=-1$.

$\Rightarrow$ How far can we move $\tilde{x}$ from x to the right so still the majority is $y=1$?

$\Rightarrow$ We see that we can move it at most by $\sigma Q^{-1}(1-p)$

$\Rightarrow$ The larger the p, (more biased) the original average was, the more we are adversarial robust.

- **Why not just take a very large σ?**

The averaging (smoothing) will in general <u>increase</u> the standard Risk.
(Trade off)