تمرین سری سوم SPML تئوری امیرمحمد عزتی 402212269

سوال ①

$$\begin{cases} f(x) = Sign(w^T x + b) & \text{دسته بند} \\ y \in [-1, 1] \end{cases}$$

الف) شعاع مقاومت :

$f(x) = Sign(w^T x + b)$ $w^T x + b = 0$ مرز کلاس بندی

$f(x+\varepsilon) = Sign(w^T(x+\varepsilon) + b)$ $w^T x + b + w^T \varepsilon = 0$

اگر $w^T \varepsilon = -(w^T x + b)$ باشد، آنگاه به مرز کلاس بندی این دسته بند می رسیم.

پس بزرگترین اندازه ε که به مرز تصمیم گیری برسیم از رابطه زیر می آید :

$$|w^T \varepsilon| = |w^T x + b|$$

مقدار $|w^T \varepsilon|$ زمانی ماکزیمم می شود که ε هم راستا با بردار w^T باشد (بدترین حالت)

$$|w^T \varepsilon| = \|w^T\| \, \|\varepsilon\| \underbrace{\cos\theta}_{1} = \|w^T\| \, \|\varepsilon\|$$

پس :

$$\|\varepsilon\|_2 = \frac{|w^T x + b|}{\|w\|}$$

$r = \|\varepsilon\|_2 = \dfrac{|w^T x + b|}{\|w\|}$: پس ε همان شعاع مقاومت است L_2 نرم مقدار

مقدار ε نیز برابر است با :

$$\varepsilon = \|\varepsilon\|_2 \frac{w}{\|w\|}$$

سوال ①

ب)

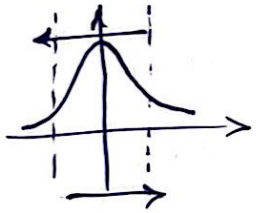
$$f(x+\epsilon) = \text{Sign}\left(w^T x + b + w^T \epsilon\right)$$

$$\epsilon \sim N(0, \sigma^2 I) \Rightarrow w^T \epsilon \sim N(0, \sigma^2 \|w\|^2)$$

$$w^T x + b + w^T \epsilon \sim N\left(w^T x + b, \sigma^2 \|w\|^2\right)$$

$$P\left[f(x+\epsilon) = 1\right] = P\left[w^T x + b + w^T \epsilon > 0\right]$$

$$P[Z > z] = P[Z \leq z] \quad = P\left[w^T \epsilon > -(w^T x + b)\right]$$



بخاطر تقارن توزیع نرمال با میانگین 0

$$= P\left[w^T \epsilon \leq w^T x + b\right]$$

$w^T \epsilon \sim N(0, \sigma^2 \|w\|^2)$

به توزیع نرمال استاندارد تبدیل می‌کنیم: $N(0, I)$

$$= P\left[\frac{w^T \epsilon}{\sqrt{\sigma^2 \|w\|^2}} \leq \frac{w^T x + b}{\sqrt{\sigma^2 \|w\|^2}}\right]$$

$\hookrightarrow N(0, I)$

$$= \phi\left[\frac{w^T x + b}{\sqrt{\sigma^2 \|w\|^2}}\right]$$

ϕ: CDF توزیع نرمال $N(0,I)$

پس می‌دانیم اگر $P\left[f(x+\varepsilon)=1\right] = \phi\left[\frac{w^T x+b}{\sqrt{\sigma^2\|w\|^2}}\right]$ آنگاه:

$$P\left[f(x+\varepsilon)=-1\right] = 1-\phi\left[\frac{w^T x+b}{\sqrt{\sigma^2\|w\|^2}}\right]$$

با توجه به رفتار CDF توزیع $\mathcal{N}(0,I)$ می‌دانیم اگر $w^T x+b>0$ باشد مقدار $P\left[f(x+\varepsilon)=1\right]$ بیشتر از $P\left[f(x+\varepsilon)=-1\right]$ خواهد بود و برعکس.

if $w^T x+b>0 \Rightarrow f(\cdot)=1$, $g(\cdot)=1$

if $w^T x+b<0 \Rightarrow f(\cdot)=-1$, $g(\cdot)=-1$

پس $f(x)=g(x)$ ■

همچنین متوجه می‌شویم که این تساوی فارغ از σ نویز گاوسی است و فقط به ورودی اولیه (بدون نویز) وابسته است.

سوال ②

الف) باتوجه به نتایج ارائه شده از این روش مقاومت می‌توان فهمید:

مدل نسبت به حمله black-box ضعف بیشتری داشته است. این یعنی حمله BB موفقیت بیشتری به حمله‌های قوی‌تر white-box داشته است. پس این روش مقاومت باعث ایجاد مبهم سازی گرادیان (obfuscating) می‌شود که حملات white-box ناموفق‌تر هستند.

همچنین PGD-20 که 20 مرحله FGSM استفاده می‌کند خیلی موفقیت بهتری نسبت به FGSM ندارد. درصورتی که حملات iterative باید موفقیت بیشتری داشته باشند.

برای ادامه ارزیابی روش و مطمئن‌تر شدن از مبهم سازی گرادیان می‌توان راه‌های زیر را پیش گرفت:

① مقدار ϵ مجاز را برای ساخت نمونه خصمانه بیشتر کنیم. اگر موفقیت باز هم بیشتر نشد پس مبهم سازی وجود دارد.

② با استفاده از جستجوی رندوم سعی در ساخت نمونه خصمانه کنیم. اگر نویز رندوم موفق به تولید نمونه خصمانه شود اما روش‌های white-box نشدن باشند، نشانه وجود مبهم‌سازی گرادیان است.

ب) این مقاله AutoAttack را معرفی کرده است که یک گروه از حمله‌های بدون پارامتر است که برای benchmark کردن مدل‌ها استفاده می‌شود.

۱- حمله APGD: این یک حمله white box مبتنی بر PGD است. با این تفاوت که در این حمله نیاز به تنظیم دستی stepsize (α) نیست و بصورت اتوماتیک تنظیم می‌شود. این حمله stepsize را براساس پیشرفت با استفاده از یک ترم شبیه به momentum برای افزایش بهره‌وری تطبیق می‌دهد.

دو نسخه مختلف از این حمله وجود دارد: APGD-CE و APGD-DLR

اولی از Cross entropy loss استفاده می‌کند که برای حمله untargeted مناسب است.

ادامه ب) دومی از تفاضل logits ها و برقرای یک نسبت استفاده می‌کند که نسبت به Shift و rescaling تغییر ناپذیر است

$$DLR(x,y) = -\frac{z_y - \max_{i \neq y} z_i}{z_{\pi_1} - z_{\pi_3}}$$

و سعی می‌کنیم در این روش maximize را DLR کنیم تا یک adversary بسازیم.

2- حمله Square Attack: یک حمله black box است. این حمله مبتنی بر score است. این حمله مبتنی بر جستجوی رندوم است تا یک آشفتگی در نرم مجاز پیدا کند. این حمله از لحاظ محاسباتی بهینه است و نیاز به اطلاعات گرادیان ندارد. همچنین زمانی که بقیه حمله‌های white box fail شوند، باز هم عملکرد خوبی دارد.
در زمانی که obfuscating gradient داشته باشیم این حمله مناسب است.

3- حمله FAB: این یک حمله white box است.
این حمله با کمک minimize کردن نرم اغتشاش حضمانه سعی می‌کند باعث misclassification تارگت شود. از لحاظ محاسباتی بهینه است و حتی در زمانی که gradient masking وجود دارد موثر است.

دو حمله APGD, FAB دو حالت targeted و untargeted دارند که پیاده‌سازی شده است و این فریم ورک شامل این سه حمله است.

سوال (3)

۱) رفتارهایی که نشان می دهند obfuscated gradient بوجود آمده عبارتند از:

a) حمله تک مرحله ای بهتر از حمله تکرارشونده (چند مرحله ای) باشد.

چون در حمله چند مرحله ای چندین بار براساس گرادیان تصویر را perturb می کنیم انتظار می رود که به تصویری برسیم که با احتمال بیشتری باعث mis classification شود و دقت بیشتری داشته باشد. پس اگر این طور نشد احتمال دارد، مبهم سازی در گرادیان بوجود آمده باشد.

b) حمله black-box بهتر از حمله white-box باشد.

چون در حمله white-box ما اطلاعات بیشتر ازجمله معماری شبکه، وزن ها، گرادیان ها و... را داریم، انتظار می رود با بهینه سازی دقیق، نمونه های خصمانه قوی تری بسازیم. پس اگر این طور نشد می توان نتیجه گرفت مبهم سازی و اشکالی در گرادیان وجود دارد که حملات black-box با محدودیت، موفقیت بیشتری داشته اند.

c) حمله بدون محدودیت به موفقیت 100% نرسد.

برای هر کلاس

این حالت یکی ما بدون داشتن محدودیتی برای ϵ، نتوانیم نمونه خصمانه ای تولید کنیم که شبکه عصبی را بتواند گول بزند. امکان اینکه نتوانیم نمونه خصمانه ای بسازیم که محدودیتی در ϵ نداشته باشد، وجود ندارد. پس اگر این رخ داد حتما مشکل یا ابهامی در گرادیان وجود دارد.

d) نمونه رندوم (با نویز رندوم) بتواند یک نمونه خصمانه باشد.

درحالتی که با یک نمونه رندوم بتوان شبکه را گول زد امّا امکان ساخت نمونه خصمانه با گرادیان (white-box) وجود نداشته باشد یعنی مشکلی در گرادیان شبکه وجود دارد.

ادامه ۱) e) افزایش محدودیت میزان انحراف باعث افزایش موفقیت حمله نشود.

با افزایش ϵ یا مقدار انحراف مجاز، انتظار می‌رود باکس بزرگتری حول یک نمونه را explore کنیم و احتمال اینکه نمونه حضمانه برای یک عکس بیابیم، بیشتر می‌شود. پس اگر این اتفاق رخ ندارد، یعنی مشکلی برای گرادیان و فرآیند بهینه‌سازی مان پیش آمده است.

---

2) * روش DBPA برای مقابله با گرادیان های پراکنده (shattered) استفاده می‌شود. در مواقعی که با توابع مشتق ناپذیر و یا ناپیوسته طرف هستیم و امکان می‌سپه دقیق گرادیان وجود ندارد، می‌توانیم با این تکنیک، گرادیان را تخمین بزنیم. مثلا در شبکه‌ای که $f(g(x))$ را خروجی می‌دهد به شکلی که $g(\cdot)$ تابعی برای preprocessing و $f(\cdot)$ تابعی برای کلاس بندی است؛ اگر $g(\cdot)$ یک تابع مشتق ناپذیر باشد می‌توانیم $\nabla_x g(x) = 1$ در نظر بگیریم تا $\nabla_x f(g(x))$ را تخمین بزنیم.

* روش EOT برای مقابله با گرادیان تصادفی (randomized) است.

در مواقعی که مثلا یک transform تصادفی ای روی ورودی شبکه رخ می‌دهد که باعث می‌شود هر باری که یک ورودی خاص را به شبکه می‌دهیم خروجی متفاوتی بگیریم، می‌توان از EOT استفاده کرد. به این صورت که به جای محاسبه‌ی $\nabla f(t(x))$ $E_{t\sim T}$ می‌توانیم $\nabla E_{t\sim T} f(t(x))$ را محاسبه کنیم و این دو مقدار را تقریبا برابر در نظر بگیریم. یعنی transform های مختلف روی یک ورودی بزنیم و از آن‌ها expectation بگیریم.

(2 ادامه) * روش reparametrization برای مقابله با exploding/vanishing gradient استفاده می‌شود.

مثلا در حالتی که یک دسته بند $f(g(x))$ داریم که g یک تابع مشتق ناپذیر است که x را به $\hat{x}$ تبدیل می‌کند. می‌توان با یک تغییر متغیر $x=h(z)$، یک تابع h ای یافت که اگر خروجی اش را به g بدهیم دقیقا همان مقدار h را بدهد یعنی $g(h(z))=h(z)$. اینطور تواستیم تابع مشتق ناپذیر g را دور بزنیم و از h که مشتق پذیر است استفاده کنیم آنگاه:

$$\nabla_x f(g(x)) = \nabla_z f(h(z))$$

(3 در این روش encoding به این صورت عمل می‌شود که به ازای هر پیکسل از عکس x که می‌تواند بین ۰ تا ۱ باشد یک encoding با سطح l انجام می‌دهد. بطوریکه

$$\begin{cases} \text{if } x_{i,j,c} > \frac{k}{l}, & \tau(x_{i,j,c})_k = 1 \\ \text{else} & \tau(x_{i,j,c}) = 0 \end{cases}$$

چون این روش encoding یک نوع discretization محسوب می‌شود و باعث گرادیان پراکنده می‌شود، می‌توانیم با روش DBPA یک تابع دیگر مشابه جای بخش discrete کننده قرار بدهیم و به کمک آن adversarial example بسازیم و با آن‌ها adversarial training را انجام بدهیم.

تابع زیر به عنوان تابع جایگزین، که مشتق پذیر است، می‌تواند استفاده شود:

$$\hat{\tau}(x_{i,j,c})_k = \min\left(\max\left(\frac{x_{i,j,c}}{k/l}, 0\right), 1\right)$$

$$\tau(x_{i,j,c})_k = \text{floor}\left(\hat{\tau}(x_{i,j,c})\right)$$

به جای استفاده از تابع $\tau(x_{i,j,c})$ دیگر می‌توانیم از $g(\alpha) = \hat{\tau}(x_{i,j,c})$ استفاده کنیم که مشتق پذیر است. با استفاده از این تابع g می‌توان به شبکه حمله کرد و نمونه های خصمانه تولید کرد. از این نمونه های خصمانه تولید شده نیز می‌توان در train استفاده کرد تا شبکه نسبت به نمونه خصمانه نیز مقاومت واقعی پیدا کند.