a) اگر target_class = t باشد: هدف تغییر کلاس پیش‌بینی شده مدل به کلاس $t$ است:

$$X_{adv} = X - \varepsilon \, sign \, \nabla_m \, loss(\theta; X, t)$$

b)

$$X = X_0 + \delta$$

$$w^T X + w_0 = w^T X_0 + w^T \delta + w_0$$

$$\Rightarrow \text{optimization: } \min_{\delta} \; w^T \delta + w^T X_0 + w_0 \quad s.t \quad \|\delta\|_\infty \le \varepsilon$$

با توجه به نابرابری Holder $\longrightarrow$ $w^T \delta \ge -\|\delta\|_\infty \|w\|_1 \ge -\varepsilon \|w\|_1$

$w^T \delta = -\varepsilon \|w\|_1$ وقتی بدست می‌آید که $\delta = -\varepsilon \cdot sign(w)$ زیرا:

$$w^T \delta = -\varepsilon w^T sign(w) = -\varepsilon \sum_{i=1}^{d} w_i \, sign(w_i) = -\varepsilon \sum_{i=1}^{d} |w_i| = -\varepsilon \|w\|_1 \quad (1)$$

FGsm loss function two-class binary classifier:

objective function در بیشینه attack مجاز

$$\rho(X) = w^T X + w_0 = (w_i^T X + w_{i0}) - (w_t^T X + w_{t0})$$ (مطلوب) بیانگر تابع کلاس $i$ و کلاس $t$ (نامطلوب) است.

اگر $\rho(X) < 0$ آنگاه $X$ ، misclassify شده است. در نتیجه تابع هزینه را می‌توان اینگونه تعریف کرد: $J(X) = (w_t^T X + w_{t0}) - (w_i^T X + w_{i0})$.

با بیشینه کردن این تابع هزینه، تخمین می‌شود عملکرد دسته‌بند بدتر می‌شود. $J(X; w) = -(w^T X + w_0)$

first order approximation: $J(X; w) = J(X_0 + \delta; w) \approx J(X_0; w) + \nabla_m J(X_0; w)^T \delta$.

بنابراین پیدا کردن $\delta$ که $J(X_0+\delta; w)$ را بیشینه می‌کند تقریباً معادل پیدا کردن $\delta$ ای است که $J(X_0; w) + \nabla_m J(X_0; w)^T \delta$ را بیشینه می‌کند است.

$$\max_{\delta} \; J(X_0; w) + \nabla_\delta J(X_0; w)^T \delta \quad s.t \quad \|\delta\|_\infty \le \varepsilon \quad \Rightarrow \quad \min_{\delta} \; -\nabla_m J(X_0; w)^T \delta - J(X_0; w) \quad s.t \quad \|\delta\|_\infty \le \varepsilon$$

این مسئله optimization معادل همان مسئله بهینه‌سازی رابطه (1) است. بنابراین جواب خوبی برای است با:

$$\delta = -\varepsilon \cdot sign(\nabla_m J(X_0; w)) \Rightarrow X = X_0 + \varepsilon \cdot sign(\nabla_m J(X_0; w)) \Rightarrow \text{FGsm}$$

یک بهینه برای این تابع هدف است.

c) با توجه به اینکه شبکه‌های عصبی معمولاً غیرخطی هستند پس $\nabla_m loss(x; w)$ همیشه یک چیز ثابت نیست و با تک مرحله و طی گام بالا ممکن است آشفتگی بهینه روی حاصل نشود و نیاز به یک نزدیک iterative با طی گام کمتر باشد.

(a

PGD برخلاف FGSM که در یک مرحله سعی در حل مسئله دارد، به صورت iterative حل می‌کند.

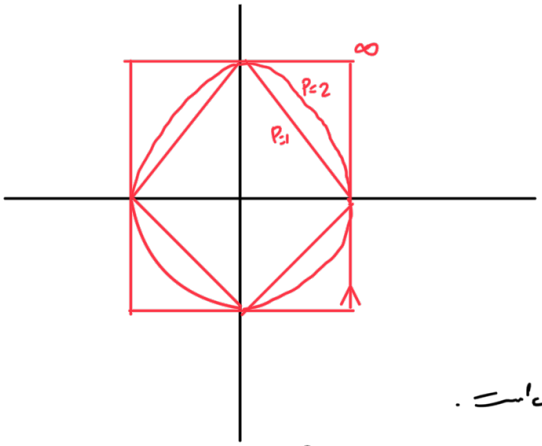
$$x_{adv}^{t} = \Pi_{\varepsilon}\left(x_{adv}^{t-1} + \alpha \, \mathrm{sign}\, \nabla_{x} Loss(\theta; x_{adv}^{t-1} + \delta, y)\right)$$

$\alpha$ بسیار کوچکتر $\varepsilon$ است که باعث می‌شود با یک طول گام کوچکتر در راستای گرادیان ولی با تعداد گام بیشتر، حرکت کند.
و یک Projection مانع از خروج از محدوده $\varepsilon$ می‌شود. به خاطر iterative بودن و یک مرحله‌ای نبودن با طول گام زیاد می‌تواند در خیلی از موارد، تغییرات کمتر نسبت به نقطه شروع و حمله ضعیف به مدل داشته باشد.

(b

بله می‌تواند از $l_0$، $l_1$، $l_2$، $l_\infty$ استفاده کند. در حوزه تصویر

$l_0$ سعی می‌کند با تغییر تعداد کمی پیکسل و $l_\infty$ با تغییر همه پیکسل‌ها ولی به میزان کم، $l_1$ و $l_2$ ترکیبی از این دو رویکرد را دارند.



P_norms in $\mathbb{R}^2$

هم چنین مسئله optimization مربوط به $l_0$ غیر محدب و سخت است.
$l_1$ در مقایسه با $l_2$ تعداد کمتری پیکسل را معمولاً تغییر می‌دهد و هدفش کاهش قدر مطلق تغییرات اندازه پیکسل‌هاست، optimization آن محدب است.
هدف $l_2$ اما کاهش ناحیه اقلیدسی تغییرات پیکسل‌هاست و optimization آن محدب است.
$l_\infty$ سعی در کاهش بیشینه قدر مطلق تغاضل پیکسل‌هاست و " " " "

(c

$l_2$ بر دقت با اندازه تصویر وابسته است، اگر با احتیاط استفاده نشود ممکن است باعث ایجاد نمونه‌های خصمانه اشتباه شود (یعنی نمونه‌هایی که باعث تغییر کلاس آن تصویر یا به عبارت بهتر، تصویری غیر قابل تشخیص تولید کند ولی $l_\infty$ باعث تغییر پیکسل‌ها به میزان بسیار کم می‌شود و کنترل اینکه نمونه خصمانه اشتباه تولید نکند راحت‌تر است.