
بسم الله الرحمن الرحيم

دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

آتنا: چارچوبی برای دفاع در برابر حملات تخاصمی با استفاده از مدل های مولد مشروط بر کلاس

پایان‌نامه کارشناسی ارشد مهندسی کامپیوتر - هوش مصنوعی و رباتیکز

آرین تشکر

استاد راهنما

دکتر محمد حسین منشئی

۱۴۰۳

دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

پایان‌نامه کارشناسی ارشد رشته مهندسی کامپیوتر – هوش مصنوعی و رباتیکز آقای آرین تشکر

تحت عنوان

آتنا: چارچوبی برای دفاع در برابر حملات تخاصمی با استفاده از مدل های مولد مشروط بر کلاس

در تاریخ ۱۴۰۳/۰۱/۰۱ توسط کمیته تخصصی زیر مورد بررسی و تصویب نهایی قرار گرفت:

۱ - استاد راهنمای پایان‌نامه	دکتر محمد حسین منشئی
۳ـ استاد داور	ایکس وای زی
۴ـ استاد داور	ایکس وای زی
سرپرست تحصیلات تکمیلی دانشکده	دکتر بهزاد نظری

تشکر و قدردانی

قدردان راهنمایی های ارزنده استاد گرانقدر جناب آقای دکتر محمد حسین منشئی هستم که بدون شک پشتیبانی و راهنمایی‌هایشان روشنگر مسیر انجام پژوهش بود.

تقدیم به

پدر و مادر عزیزم

مستحکم ترین پشتوانه و حامیانم که وجودشان منشأ آرامش و انگیزه است...

و خواهرم

مشوقان و همراهان همیشگی...

فهرست مطالب

فصل سوم : پیشنهاد روشی نوین برای مقابله با حملات تخاصمی

فصل چهارم : شبیه سازی و نتایج ارزیابی

فصل پنجم : نتیجه‌گیری و پیشنهادها

فهرست شکل‌ها

فهرست جداول

فهرست اختصارات

C

CW ..... Carlini-Wagner

F

FGSM ..... Fast Gradient Sign Method

G

GI-AT ..... Geometry-aware Instance-reweighted Adversarial Training

P

PGD ..... Projected Gradient Descent

Y

YOPO ..... You Only Propagate Once

چکیده

این یک متن نمونه است... این یک متن نمونه است...این یک متن نمونه است...این یک متن نمونه است...این یک متن نمونه است...این یک متن نمونه است...این یک متن نمونه است...این یک متن نمونه است...این یک متن نمونه است...این یک متن نمونه است...

کلمات کلیدی: ۱- یادگیری ماشین، ۲- یادگیری عمیق، ۳- مدل های مولد[۱]، ۴- دسته بند ها[۲]، ۵- حملات تخاصمی[۳]

[1] Generative Models
[2] Classifiers
[3] Adversarial Attacks

فصل اول

مقدمه

در این فصل به طور مختصر ابتدا به بیان اهمیت دفاع در برابر حملات تخاصمی روی دسته‌بندها پرداخته خواهد شد و سپس در ادامه، ساختار گزارش پیشِ رو بسط داده خواهد شد.

۱ - ۱ اهمیت مسئله

امروزه یادگیری عمیق[1] به عنوان ابزاری قدرت مند و بهینه برای حل گستره‌ی وسیعی از مسائل پیچیده روزمره شناخته می‌شود که حل آن‌ها با روش های یادگیری ماشین سنتی بسیار دشوار و بعضاًغیر ممکن بود. در سال های اخیر، یادگیری عمیق چنان دستخوش پیشرفت های ژگرفی شده که اکنون قادر است در عدیده‌ای از اهداف یادگیری، از کارایی انسان نیز پیشی بگیرد. با توجه به گسترش روز افزون استفاده از هوش مصنوعی - و به خصوص یادگیری عمیق- در مصارف روزانه و صنایع، اهمیت اندیشیدن تسهیلاتی برای مقابله با حملات سایبری احتمالی به چنین سیستم هایی نیز به تبع دو چندان شده است. به عنوان مثال، در [۱] نشان داده شده است که می توان به یک سیستم ماشین خودران[2] که توسط یک کنترل کننده‌ی هوش مصنوعی اداره می شود، در کمتر از ۲ ثانیه حمله و آن را از مسیر خارج کرد. در [۲] نمونه دیگری از یک حمله به سیستم ماشین خودران نشان

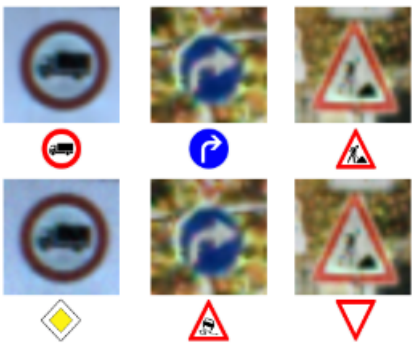
[1] Deep Learning
[2] Self-driving Car

داده شده است که در آن می‌توان این سیستم را در تشخیص علائم رانندگی دچار خطا کرد. این حمله در شکل ۱ - ۱ نشان داده شده است. همچنین در [۳] نمونه عملی یک حمله به سه نمونه الگوریتم تاجر خودکار[1] حمله شده است که در طی آن سیستم تاجر خودکار مبتنی بر مدل های یادگیری ماشین دچار خطا در پیشبینی قیمت آینده یک سهم می شوند.

دسته خاصی از حملات سایبری[2] اعمال پذیر روی دسته‌بند های مبتنی بر یادگیری ماشین، حملات تخاصمی هستند که تمرکز اصلی این تحقیق می باشند. در فصل ۲ به تفصیل راجع به این حملات توضیح داده خواهد شد.

۱ - ۲ ساختار گزارش

در ادامه این گزارش، در فصل ۲ پیشینه پژوهش و مبانی لازم برای درک بهتر فصل‌های آتی مورد بررسی قرار خواهند گرفت. در فصل ۳ روش پیشنهادی مسئله مورد بررسی به طور مشخص مطرح شده و روش پیشنهادی برای حل آن بیان خواهد شد. سپس در فصل ۴ نتایج شبیه‌سازی و مقایسه‌های لازم با استفاده از سنجه[3] های مناسب مورد بررسی قرار خواهند گرفت. در نهایت در فصل ۵ به جمع‌بندی نهایی و جهت های احتمالی تحقیقات آینده پرداخته خواهد شد.



شکل ۱ - ۱: نمونه حمله به یک سیستم خودران. ردیف بالا نشان دهنده تصاویر با تشخیص برچسب درست توسط مدل دسته‌بند می باشند. ردیف پایین همان تصاویر به همراه نویزی نامحسوس برای چشم غیر مسلح می باشند که باعث ایجاد خروجی اشتباه توسط مدل دسته‌بند می شود [۲].

[1] Auto Trader
[2] Cyber Attacks
[3] Metric

فصل دوم

پیشینه پژوهش و مبانی هوش مولد

۲-۱ مقدمه

در این فصل به بیان مقدماتی در مورد حملات تخاصمی، روش های دفاع در برابر آن ها پرداخته خواهد شد. سپس مبانی هوش مولد و به خصوص مدل های مولد تخاصمی و مدل های انتشاری به تفصیل مورد بررسی قرار خواهند گرفت.

۲-۲ حملات تخاصمی

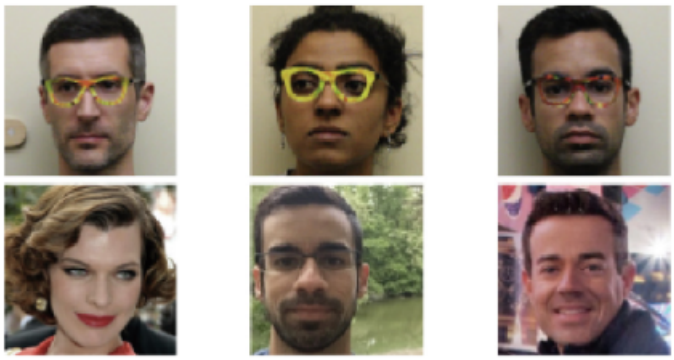
حملات تخاصمی به تکنیک هایی در حوزه یادگیری ماشین گفته می شود که برای هدف خاص فریب دادن مدل‌های مختلف به کمک توسعه و اعمال ورودی‌های آسیب زننده به این مدل ها طراحی شده اند [۴–۶]. حملات تخاصمی عموماً طی فرایندی تحت عنوان **اختلال تخاصمی**[۱] ایجاد می‌شوند. این پروسه شامل افزودن مقادیر ناچیزی نویز[۲] به ورودی مدل دسته‌بند قربانی[۳] می شود که با وجود نامحسوس بودن به چشم غیرمسلح، باعث اختلال در خروجی دسته‌بند خواهند شد [۷]. تحقیقات انجام پذیرفته روی این دسته از حملات عموماً

[1] Adversarial Perturbation
[2] Noise
[3] Victim

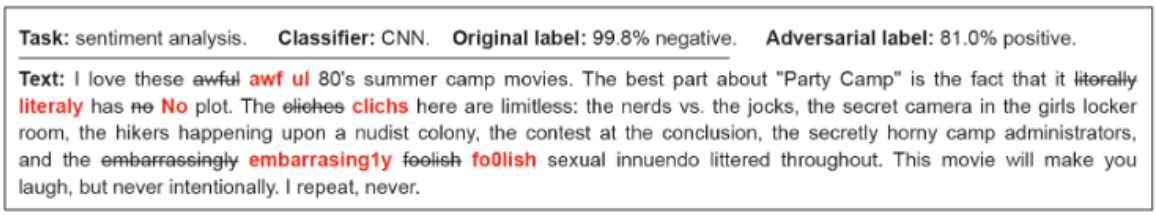
در حوزه تصویر می باشد که با تغییرات جزئی و نامحسوس روی مقادیر پیکسل[1] های به خصوص، خروجی یک دسته‌بند تصویر تغییر خواهد کرد. با این وجود،از آنجایی که اکثر روش های طراحی حملات تخاصمی از ماهیت ورودی مدل به طور مستقیم برای طراحی حمله استفاده نمی کنند،این روش ها می توانند قابل تعمیم به تمامی دسته‌بندها، چه در حوزه تصویر و چه خارج از آن، باشند [۵، ۸]. در شکل ۲ - ۱ نمونه‌ای از یک حمله تخاصمی به یک مدل تشخیص چهره را می توان مشاهده کرد. با قرار دادن یک عینک حاوی نویز تخاصمی روی صورت افرادی که در سطر بالا قرار دارند، مدل تشخیص چهره، چهره‌ی این افراد را به اشتباه به عنوان افراد نظیر در سطر پایین تشخیص داده است. همچنین در ۲ - ۲ نمونه ای از یک حمله تخاصمی به یک مدل دسته‌بند متن برای وظیفه تحلیل احساسات[2] را می توان مشاهده کرد.

به طور رسمی یک حمله تخاصمی را می توان طبق تعریف ارائه شده در [۹] بررسی کرد. فرض کنیم که مدل دسته‌بند $f(\cdot)$ را در اختیار داشته باشیم که خروجی آن روی یک نمونه ورودی x یک توزیع آماری روی تمامی کلاس های ممکن باشد. کلاس تشخیص داده شده توسط این مدل در معادله ۲ - ۱ نمایش داده شده است.

$$y = \arg_c \max f(x) \qquad (۱-۲)$$



شکل ۲ - ۱: نمونه ی حمله تخاصمی به یک مدل تشخیص چهره [۷]. افراد سطر بالا به اشتباه توسط مدل تشخیص چهره به فرد متناظر در سطر پایین تشخیص داده شده اند.



شکل ۲ - ۲: یک مدل دسته‌بند متن مبتنی بر CNN که با تغییرات جزئی در متن نظر ارسال شده، با وجود حفظ محتوای کلی نظر، دچار اشتباه شده است [۷].

[1] Pixel
[2] Sentiment Analysis

اکنون یک نمونه تخاصمی[1] به ورودی دستکاری شده‌ی $\hat{x}$ گفته می شود که طبق معادلات ۲-۲ و ۲-۳ از افزودن مقدار ناچیزی نویز δ به ورودی اولیه x به دست می آید به طوری که توزیع آماری خروجی مدل و یا کلاس تشخیص داده شده توسط مدل را طوری تغییر دهد که با حالت اولیه متفاوت باشد.

$$\hat{x} = x + \delta, \tag{۲-۲}$$

$$s.t. \|\delta\| < \epsilon,\ f(\hat{x}) \neq f(x) \vee\ \hat{y} = \arg\max_{c} f(\hat{x}) \neq y \tag{۲-۳}$$

۲-۲-۱ دسته‌بندی حملات تخاصمی

امنیت یک مدل یادگیری ماشین با توجه به اهداف خصمانه مورد نظر و قابلیت‌های مهاجم[2] ارزیابی می شود [۴، ۵]. پیش از پرداختن به دسته‌بندی حملات تخاصمی، ابتدا کمی راجع به مدل‌های تهدید[3] در حوزه یادگیری ماشین با توجه به قدرت و دسترسی مهاجم صحبت خواهد شد.

۲-۲-۱-۱ رویه‌ی حمله

رویه‌ی حمله[4] عبارتیست که به تمام روش های ممکن موجود برای یک مهاجم برای حمله به یک سیستم اطلاق می شود. یک سیستم تصمیم‌گیرنده مبتنی بر یادگیری ماشین را می‌توان عملاً به عنوان یک خط لوله[5] برای پردازش داده‌های ورودی متصور شد. بدین ترتیب، دنباله ای از عملیات ساده روی داده‌های ورودی در زمان استفاده از یک مدل یادگیری ماشین را می‌توان به صورت زیر خلاصه کرد:

۱. **جمع آوری** داده‌های ورودی از سنسورها و یا انبارهای داده

۲. **انتقال** داده‌های جمع‌آوری شده در مرحله قبل به دامنه دیجیتال

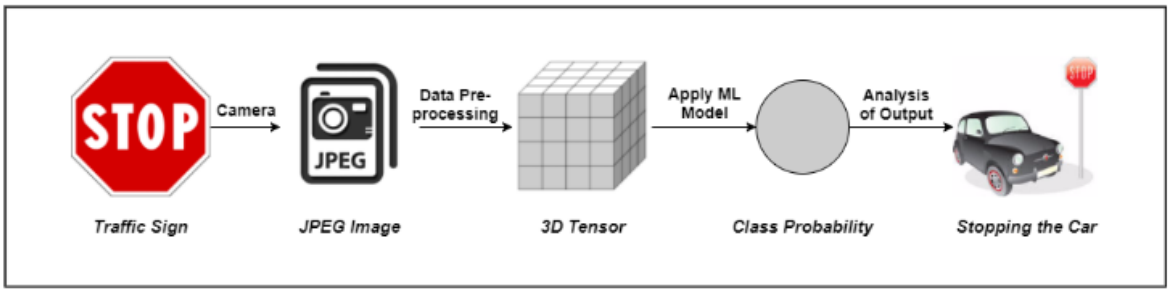
۳. **پردازش** داده‌های دیجیتال برای تبدیل آن‌ها به قالب قابل استفاده توسط مدل یادگیری ماشین و دریافت خروجی از مدل

۴. **اتخاذ تصمیم** بر اساس خروجی مدل

نمونه چنین دنباله ای را می توان در شکل ۲-۳ مشاهده کرد.

در این خط لوله، سیستم ابتدا با استفاده از دوربین های نصب شده در اقصی نقاط خودرو، تصاویری را به عنوان ورودی دریافت می کند. این تصاویر طی عملیات پیش‌پردازش مناسب به فرمت قابل استفاده برای مدل

[1] Adversarial Sample
[2] Attacker
[3] Threat Models
[4] Attack Surface
[5] Pipeline



شکل ۲-۳: نمونه خط لوله یک سیستم ماشین خودران [۴]

یادگیری ماشین در می آیند (در این مثال خاص یک تنسور[1] سه بعدی از مقادیر پیکسل های تصاویر دریافت). مدل پس از استخراج ویژگی از ورودی های دریافتی، خروجی مطلوب را تولید می‌کند (به عنوان مثال، احتمال مشاهده علامت ایست) و در نهایت یک تصمیم‌گیرنده بر اساس این خروجی، تصمیمی را اتخاذ می کند (در این مثال، توقف خودرو). در این مثال خاص، رویه‌ی حمله را می‌توان با توجه به خط لوله پردازش اطلاعات ورودی تعریف کرد. به طور دقیق‌تر، یک مهاجم می‌تواند با ایجاد اختلال در هر یک از مراحل *جمع‌آوری* و یا *پردازش* مدل قربانی را مسموم و در اثر آن، خروجی نهایی مدل را دستکاری کند.

سناریوهای اصلی حملات براساس رویه‌ی حمله مورد استفاده به شرح زیر می باشد [۴–۷، ۱۰]:

۱. **حمله گریزانه**[2]: معمول‌ترین دسته‌ی حملات خصمانه. در این حالت مهاجم در صدد گریز از یک سیستم امنیتی به وسیله‌ی اعمال تغییرات در نمونه‌های ورودی در زمان تست، بر می آید. در این حالت هیچ پیش‌فرضی در رابطه با دسترسی مهاجم به داده‌های آموزشی مدل، وجود ندارد.

۲. **حمله مسموم‌کننده**[3]: در این دسته از حملات که در زمان آموزش مدل قابل پیاده سازی هستند، مهاجم تلاش می‌کند که داده‌های آموزشی مدل را پیش از فرایند یادگیری، مسموم کند. به عبارت دقیق‌تر، مهاجم با افزودن نمونه‌های آموزشی جدیدی که با دقت و با هدف خاصی برای انحراف مدل آموزش دیده شده‌ی نهایی طراحی شده‌اند، به مجموعه داده های آموزشی، سعی می کند که کل فرایند آموزش مدل را مختل و یا خروجی آن را پس از اتمام آموزش دچار تغییرات نامطلوب نماید. واضح است که در این دسته از حملات فرض شده است که مجموعه داده آموزشی مدل قربانی در دسترس مهاجم قرار دارد.

۳. **حمله اکتشافی**[4]: این حملات که در با پیش فرض دسترسی جعبه سیاه[5] به مدل پیاده سازی می شوند، برخلاف دسته پیشین، تاثیری روی مجموعه داده‌های آموزشی ندارند. در یک حمله اکتشافی، مهاجم سعی

[1] Tensor
[2] Evasion Attack
[3] Poisoning Attack
[4] Exploratory Attack
[5] Black Box

می کند با داشتن دسترسی محدود به مدل، تمام اطلاعات ممکن را راجع به الگوریتم یادگیری استفاده شده در سیستم مورد حمله و الگوهای احتمالی موجود در مجموعه داده‌های آموزشی را دریافت کند.

برای تعریف مدل تهدید نیازمند آن هستیم که نوع دسترسی مهاجم به قربانی را نیز در نظر بگیریم [۴، ۵] اگر به مثال ماشین خودران باز گردیم، برای یک مهاجم قوی می‌توان دسترسی به معماری مدل استفاده شده و مجموعه داده‌های آموزشی را متصور شد در حالی که یک مهاجم ضعیف تر فقط احتمالاً به داده‌های زمان تست دسترسی دارد. با وجود این که هر دو مهاجم از یک رویه برای حمله به مدل استفاده می کنند، مهاجم اول به دلیل در اختیار داشتن اطلاعات کامل تر، **قوی تر** تلقی می شود.

بدین ترتیب، حمله ها را می‌توان بر اساس گستره دسترسی مهاجم به دو دسته مهم حملات جعبه سیاه و جعبه سفید[1] دسته‌بندی کرد [۲، ۱۱، ۱۲].

- **حملات جعبه سفید**: در این دسته از حملات، قوی‌ترین دسترسی ممکن برای مهاجم فرض می‌شود. مهاجم در این نوع حمله از اطلاعات کامل راجع به مدل مورد استفاده برای تصمیم‌گیری (مثلاً معماری دقیق مدل، تابع هزینه مورد استفاده برای آموزش، گرادیان خروجی مدل نسبت به هر متغیر مطلوب مهاجم، وزن های مدل و غیره) برخوردار است. همچنین مهاجم از نحوه آموزش مدل (مثلاً الگوریتم بهینه‌سازی مورد استفاده) مطلع بوده و به مجموعه داده‌های آموزشی مدل دسترسی کامل دارد. با این مفروضات، مهاجم سعی می‌کند از نقاط ضعف مدل در فضای ویژگی[2] مطلع شود و از آن‌های برای تخریب عملکرد مدل سوء استفاده کند.

- **حملات جعبه سیاه**: در این حملات که در نقطه مقابل حملات جعبه سفید قرار می‌گیرند،هیچ پیش‌فرض خاصی برای مهاجم در نظر گرفته نمی شود و نمایانگر یک سناریوی حمله محتمل‌تر است که در آن مهاجم سعی می‌کند با در اختیار داشتن اطلاعات محدود راجع به عملکرد مدل و خروجی های دریافتی از ورودی‌هایی که خودش به مدل ارسال می‌کند، از نقاط ضعف مدل پرده برداری کند.

در جدول ۲ - ۱ مقایسه‌ای اجمالی بین این دو دسته از حملات آمده است.

مدل تهدید علاوه بر نوع دسترسی مهاجم، به هدف غایی او نیز وابسته است. اهداف یک مهاجم از حمله به یک سیستم تصمیم‌گیرنده مبتنی بر یادگیری ماشین را می توان به موارد زیر خلاصه کرد:

۱. **تقلیل اطمینان**[3]: در این حالت، مهاجم سعی می کند سطح اطمینان[4] خروجی دسته‌بند را برای دسته تشخیص داده شده y_{pred} کاهش دهد در حالی که خروجی کلاس دسته‌بند دچار تغییر نشود. به عبارت

[1] White Box
[2] Feature Space
[3] Confidence Reduction
[4] Confidence

دیگر:

$$y_{pred} = \hat{y} = y,\ f(x)_y > f(\hat{x})_y$$

۲. **دسته‌بندی اشتباه غیر هدفمند**[1]: در این حالت، مهاجم در صدد تغییر خروجی مدل به **هر** کلاس $\hat{y} \neq y_{pred}$ بر می آید.

۳. **دسته‌بندی اشتباه هدفمند**[2]: در این حالت، مهاجم تلاش می‌کند خروجی مدل را به کلاس خاص مطلوب $\hat{y_{desired}} \neq y_{pred}$ **تغییر دهد.**

هدف و تمرکز اصلی این پژوهش روی حملات تخاصمی گریزانه‌ی جعبه سفید است که از گرادیان خروجی یک مدل نسبت به ورودی آن مطلع هستند. در این حملات یک ورودی ساختگی در زمان تست به صورت مصنوعی توسط مهاجم با علم به اطلاعات خصوصی سیستم مورد حمله و با هدف ایجاد اختلال در خروجی آن،

جدول ۲ - ۱ : مقایسه حملات جعبه سیاه و جعبه سفید

	حملات جعبه سیاه	حملات جعبه سفید
تعریف	به دسته ای از حملات گفته می شود که در آن ها ساختار درونی و طراحی سیستم مورد حمله از مهاجم مخفیست.	به دسته ای از حملات گفته می شود که در آن ها مهاجم اطلاعات کامل از معماری و ساختار درونی سیستم مورد حمله دارد.
سطح اطلاعات قابل دسترسی	نیازمند اطلاعات بنیادی راجع به سیستم قربانی و نحوه عملکرد آن نیست.	نیازمند اطلاعات محرمانه راجع به سیستم مانند وزن های مدل مورد استفاده، معماری آن، کتابخانه های استفاده شده برای پیاده سازی و غیره می باشد.
مزایا و معایب	پیاده سازی معمولاً ساده تر، اما ضعیف تر. فرض واقع گرایانه تر راجع به مهاجمین احتمالی. کارآیی بالا در کشف ایرادات رفتاری مدل.	پیاده سازی پیچیده تر، اما به مراتب قوی تر حصول امنیت نسبی در برابر این حملات معمولاً وضعیت آرمانیست.
استراتژی حمله	در این حملات معمولاً یک مدل محلی به آموزش داده می شود که بتواند رفتار مدل قربانی را تقلید کند. این کار با استفاده از تولید نمونه های ورودی ساختگی توسط مهاجم و استفاده از برچسب های خروجی مدل قربانی روی همین ورودی ها، صورت می گیرد.	این حملات از خصوصات درونی مدل مورد استفاده سوء استفاده می کنند. به عنوان مثال حملات مبتنی بر گرادیان و یا استفاده از دانش قبلی راجع به ضعف های موجود در سیستم با فرض دانش کامل راجع به ساختار داخلی مدل استفاده شده در این دسته از حملات قرار می گیرند.

[1] Untargeted Misclassification
[2] Targeted Misclassification

طراحی و به مدل داده می‌شود. در سال های اخیر حملات گریزانه متعددی با موفقیت روی شبکه‌های یادگیری عمیق اعمال شده اند. خواننده برای مرور کاملی بر روش‌های روز به [۴–۸، ۱۰] ارجاع داده می‌شود.

به طور کلی، فرایند تولید یک نمونه تخاصمی را می توان به صورت معادله ۲-۴ فرمول‌بندی کرد [۱۳]:

$$\begin{aligned} &\text{minimize} \;\; \mathcal{D}(x, x+\delta) \\ &\text{s.t.} \;\; C(x+\delta) = t \\ &\quad x+\delta \in [0,1]^n \end{aligned} \tag{۲-۴}$$

که در آن $\mathcal{D}$ یک معیار فاصله مشخص است، C یک مدل دسته‌بند و t برچسبی غیر از برچسب اصلی متناظر با ورودی x است. به عبارت ساده‌تر، معادله ۲-۴ بیان می‌کند که به ازای یک ورودی x ثابت، هدف یافتن δ مناسب است طوری که علاوه بر کمینه شدن مقدار $\mathcal{D}(x, x+\delta)$ ، خروجی دسته‌بند C روی $x+\delta$ تغییر کند.

از میان حملات جعبه سفید موجود دو حمله معروف برای آزمون مدل دفاعی ارائه شده در ادامه این گزارش، استفاده شده‌اند. حمله **Fast Gradient Sign Method (FGSM)** حاصل یکی از اولین پژوهش‌ها در زمینه حملات تخاصمی بود و امروزه به عنوان یک حمله‌ی سریع، ساده، اما نسبتاً ضعیف در میان حملات تخاصمی جعبه سفید شناخته می شود [۹] . این حمله معمولاً در حالت غیر هدفمند پیاده سازی می شود ولی پیاده سازی آن در حالت هدفمند نیز ممکن است. در مقابل این حمله یکی دیگر از معروف ترین حملات جعبه سفید، حمله **Carlini-Wagner (CW)** است که در سال ۲۰۱۷ طراحی شد [۱۴] و هنوز یکی از قوی‌ترین حملات جعبه سفید شناخته با زمان اجرای معقول است که در هر دو نوع هدفمند و غیر هدفمند قابل پیاده‌سازی است. حمله **Projected Gradient Descent (PGD)** [۱۵] مستقیماً در این پژوهش برای آزمایش مدل‌ها مورد استفاده قرار نگرفته است ولی از آنجایی که در ادامه مباحث در بخش ۲-۳ به تعریف آن احتیاج است، این حمله نیز که یکی از حملات قوی و شناخته‌شده جعبه‌سفید در ادبیات حملات تخاصمیست، مورد بررسی قرار خواهد گرفت.

۲-۱-۲-۲ حمله FGSM

فرض کنید مدل f با پارامتر های θ را در اختیار داشته باشیم. همچنین فرض کنید که (X, y) زوج های مرتبی از ورودی‌ها و خروجی‌های متناظر به f باشند و نیز تابع هزینه J که مدل به وسیله آن و طی یک فرایند بهینه‌سازی،آموزش داده شده است. اکنون عبارت ۲-۵ را در نظر بگیرید:

$$\nabla_x J(f(x;\theta), y) \tag{۲-۵}$$

این عبارت مقدار گرادیان تابع هزینه آموزش مدل f را نسبت به **ورودی** مدل (و نه پارامتر های مدل، θ) نشان می دهد. بدین ترتیب اگر به هر نحو این گرادیان به ورودی اولیه افزوده شود، ورودی تولید شده احتمالاً منجر به زیاد شدن مقدار تابع هزینه نهایی و موفقیت حمله خواهد شد. در [۹] نویسندگان از نویز تخاصمی ارائه شده در ۲-۶ استفاده می کنند و نمونه تخاصمی نهایی از معادله ۲-۷ بدست می‌آید.

$$\delta = \epsilon \cdot sgn(\nabla_x J(f(x;\theta), y)) \tag{۲-۶}$$

$$\hat{x} = x + \delta \tag{۲-۷}$$

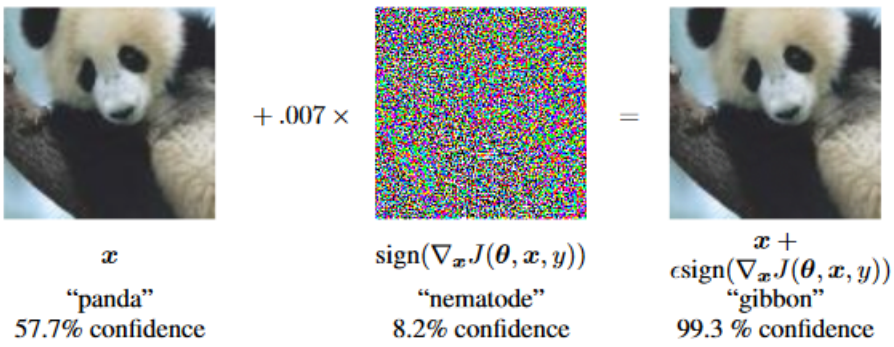
که در آن مقدار ϵ که کنترل‌کننده اندازه نویز تخاصمی بوده و بسته به مجموعه داده مورد استفاده و ضریب اطمینان لازم برای موفقیت حمله قابل تنظیم است. نویسندگان در این پژوهش مقدار 0.007 را پیشنهاد کرده اند.

به طور خلاصه در این حمله مقدار هر پیکسل از یک تصویر ورودی x به اندازه‌ی برابر بسته به جهت گرادیان تابع هزینه نسبت به x کم و یا زیاد می شود به طوری که تصویر حاصل نسبت به برچسب صحیح y هزینه بیشتر داشته باشد.

توجه شود که فرمول‌بندی ارائه شده در ۲-۶ و ۲-۷ خصوصاً برای دور کردن نتیجه دسته‌بند از برچسب حقیقی و پیاده‌سازی یک حمله غیرهدفمند است. اگر بخواهیم با استفاده از همین روش یک حمله هدفمند را با برچسب مطلوب $y_{\hat{desired}}$ اجرا کنیم، کافیست که مقدار δ به شکل معادله ۲-۸ تغییر داده شود:

$$\delta_{targeted} = -\epsilon \cdot sgn(\nabla_x J(f(x;\theta), y_{\hat{desired}})) \tag{۲-۸}$$

شکل ۲-۴ یک نمونه از حمله FGSM آورده شده است. قربانی این حمله یک شبکه GoogLeNet است و ورودی یک تصویر "پاندا"ست که مدل آموزش دیده شده می تواند با سطح اطمینان 57.7% برچسب این تصویر را به درستی تشخیص دهد. اکنون با افزودن 0.007 از نویز تخاصمی تولید شده ـ که خود با سطح اطمینان 8.7%



شکل ۲-۴: نمونه حمله FGSM. در این حمله تصویر یک پاندا با افزودن مقدار ناچیزی نویز تخاصمی به عنوان یک میمون دست‌دراز شناخته شده است [۹].

توسط مدل به عنوان "کرم‌لوله‌ای" دسته‌بندی شده‌ـ تصویر نهایی که همچنان به چشم غیرمسلح مانند تصویر اولیه است، با سطح اطمینان 99.3% توسط مدل به کلاس "میمون دست‌دراز" تعلق گرفته است.

همانطور که مشخص است، این حمله در یک گام انجام می شود و یکبار محاسبه گرادیان تابع هزینه برای تولید حمله کفایت می‌کند و بنابراین حمله FGSM بسیار سریع قابل پیاده‌سازی است. با وجود این که امروزه حملات جعبه‌سفید به مراتب قوی‌تری برای سنجش مدل‌های دسته‌بند موجود است، FGSM همچنان به عنوان یک روش آسان، قابل فهم و سریع برای آزمایش‌های اولیه مورد استفاده قرار می‌گیرد.

۲-۲-۱-۳ حمله CW

حمله‌ی دیگر مورد استفاده در این پژوهش، حمله‌ی CW است که در [۱۴] ارائه شده است. برای توضیح نحوه‌ی ساختن یک نمونه تخاصمی با این روش به معادله ۲-۴ باز می‌گردیم. در این حمله،از آنجایی که شرط $C(x+\delta)=t$ بسیار غیرخطی و غیرقابل بهینه‌سازیست، این شرط باید ابتدا به فرمی قابل بهینه‌سازی در آورده شود. برای این کار محققین این پژوهش تابع هدف f را طوری تعریف می کنند که شرط $C(x+\delta)=t$ برقرار باشد اگر و تنها اگر $f(x+\delta)\leq 0$. اکنون به جای بیان مسئله به صورت

$$\begin{aligned} &\text{minimize} \quad \mathcal{D}(x, x+\delta) \\ &\text{s.t.} \quad f(x+\delta)\leq 0 \\ &\quad x+\delta\in[0,1]^n \end{aligned} \tag{۲-۹}$$

از فرمول‌بندی معادل ۲-۱۰ استفاده می شود:

$$\begin{aligned} &\text{minimize} \quad \mathcal{D}(x, x+\delta)+c\cdot f(x+\delta) \\ &\text{s.t.} \quad x+\delta\in[0,1]^n \end{aligned} \tag{۲-۱۰}$$

که در آن $c>0$ مقداری ثابت است که قدرت حمله انجام شده را در ازای از دست دادن کمینگی $\mathcal{D}(x, x+\delta)$ تنظیم می‌کند. این دو فرمول‌بندی معادلند چرا که می‌توان نشان داد مقدار $c>0$ وجود دارد که راه‌حل بهینه در معادله ۲-۱۰ با راه‌حل بهینه ۲-۹ برابر است. در [۱۴] از نرم[1]

$$L_p(\vec{v})=\|\vec{v}\|_p=\left(\sum_{i=1}^{n}|v_i|^p\right)^{\frac{1}{p}}$$

به عنوان معیار فاصله $\mathcal{D}$ استفاده می شود. و بدین ترتیب معادله کلی نهایی حمله CW به صورت زیر در

[1] Norm

خواهد آمد:

$$\begin{aligned} &\text{minimize} \quad \|\delta\|_p + c \cdot f(x+\delta) \\ &\text{s.t.} \quad x+\delta \in [0,1]^n \end{aligned} \tag{۱۱-۲}$$

اکنون برای حل این مسئله با استفاده از روش‌های بهینه سازی تکراری، نیاز به انتخاب p ، c و تابع هزینه مناسب f است.

این حمله با نرم های L_0 ، L_2 و L_∞ قابل پیاده سازیست. از آنجایی که احتمال موفقیت حمله و نیز اندازه نرم نمونه تخاصمی بر حسب مقدار ثابت c استفاده شده، توابعی اکیداً نزولی هستند (طبق شکل ۲-۵)، می توان مقدار c بهینه را با استفاده روش جستجوی دودویی[1] بدست آورد.

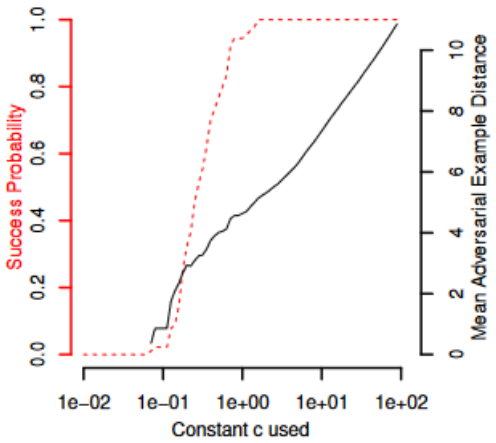
شروطی مانند $x+\delta \in [0,1]^n$ به نام محدودیت های جعبه‌ای[2] معروف هستند. برای اجرا کردن این شرط، در حمله CW از یک تغییر متغیر استفاده می شود. به جای بهینه‌سازی مستقیم مقدار δ در معادله اصلی، متغیر جدید w معرفی میشود و مقدار

$$\delta = \frac{1}{2}(\tanh(w)+1) - x$$

بهینه می شود. از آنجایی که $-1 \leq \tanh(w) \leq 1$ محدودیت جعبه‌ای ذکر شده به طور خودکار اعمال خواهد شد چراکه $0 \leq x+\delta \leq 1$ خواهد بود.

قوی ترین نوع حمله CW، L_2-CW است که مسئله بهینه‌سازی آورده شده در ۲-۱۲ را حل می کند.

$$\text{minimize} \quad \left\| \frac{1}{2}(tanh(w)+1) - x \right\|_2^2 + c \cdot f\left(\frac{1}{2}(tanh(w)+1)\right) \tag{۱۲-۲}$$



شکل ۲-۵: نمودار حساسیت موفقیت حمله و اندازه نویز تخاصمی بر حسب مقدار ثابت c [۱۴]

[1] Binary Search
[2] Box Constraints

که تابع هزینه f در آن به صورت زیر تعریف می شود:

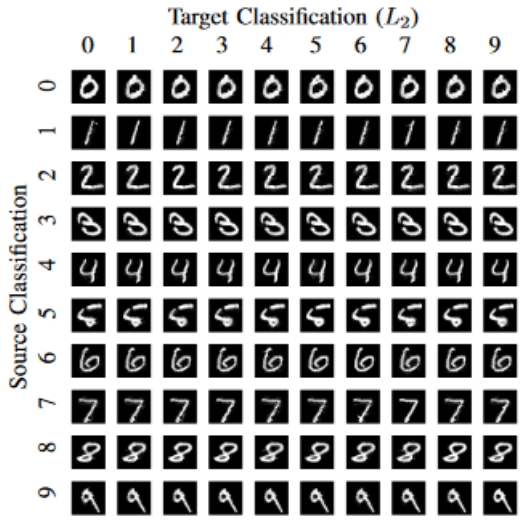
$$f(x) = \max(\max\{Z(x)_i : i \neq t\} - Z(x)_t, -\kappa)$$

و $Z(\cdot)$ خروجی های logit مدل دسته‌بند قربانی F هستند (به طوری که $F(x) = softmax(Z(x)) = y$) و پارامتر $\kappa > 0$ میزان سطح اطمینان حمله را مشخص می کند به طوری که مقادیر بزرگتر κ الگوریتم بهینه‌ساز را مجبور به یافتن پاسخ‌هایی می کند که با احتمال قوی‌تری می‌توانند خروجی دسته‌بند را تغییر دهند. به طور معمول از مقدار 14 برای این پارامتر برای تولید حملات قوی استفاده می شود. مسئله بهینه‌سازی ۲-۱۲ توسط بهینه‌ساز Adam [۱۶] حل می‌شود و در نهایت نمونه‌های تخاصمی به دست خواهند آمد. نمونه هایی از این حمله را می توان در شکل ۲-۶ مشاهده کرد. همانطور که می توان دید، بدون توجه به این که کلاس هدف دارای چه مقداریست تمامی نمونه‌های تخاصمی متناظر با تصویر سالم به چشم غیرمسلح کاملاً شبیه نمونه اصلی هستند.

۲-۲-۱-۴ حمله PGD

در نهایت در این بخش به توضیح حمله PGD [۱۵] خواهیم پرداخت. در این حمله دیدگاه جدیدی برای تولید حملات در نظر گرفته شده است. بر خلاف تعاریف پیشین، در این پژوهش محققین عملیات حمله و دفاع را به عنوان یک بازی تخاصمی بین مهاجم و مدل قربانی بررسی می کنند که تحت عنوان مسئله نقطه‌زینی زیر به عنوان تابع هزینه‌ی مدل قربانی بیان می شود:

$$\min_{\theta} \max_{\delta \in \mathcal{S}} \mathbb{E}_{(x,y)\sim\mathcal{D}} L(\theta, x+\delta, y) \qquad (۲-۱۳)$$



شکل ۲-۶: نمونه های تخاصمی تولید شده توسط L_2-CW روی مجموعه داده MNIST [۱۴].

که در آن وظیفه مهاجم حل مسئله بیشینه‌سازی درونی و وظیفه قربانی حل مسئله کمینه‌سازی برونیست. اکنون اگر فرمول ۲-۶ تشکیل حمله تخاصمی در حمله FGSM بازگردیم، می توان گفت که این حمله در واقع می تواند به عنوان یک گام از حل مسئله بیشینه‌سازی مهاجم در فرمول‌بندی ۲-۱۳ در محدوده‌ی L_∞ حول یک نمونه‌ی سالم عمل کند. اکنون اگر بخواهیم این مسئله را با استفاده از نزول گرادیان[1] حل کنیم، کافیست همین گام معرفی شده را چندین بار تکرار کنیم و حاصل را روی ϵـ کره‌ی حول نمونه‌ی سالم بیافکنیم. این روش بهینه‌سازی با محدودیت که مبنای حمله‌ی PGD است، نزول گرادیان افکنده[2] نام دارد. به طور دقیق‌تر اگر بخواهیم همچنان مانند حمله FGSM از نرم L_∞ بهره بگیریم، تشکیل یک حمله تخاصمی در PGD از تکرار گام زیر به‌دست خواهد آمد:

$$x^{t+1} = \Pi_{x+\mathcal{S}} \left(x^t + \alpha \cdot sgn(\nabla_x L(\theta, x, y)) \right) \qquad (۱۴-۲)$$

که در آن $\Pi_{x+\mathcal{S}}$ نشان دهنده عملیات افکنش روی ϵـ کره‌ی $x + \mathcal{S}$ بوده و این‌بار α نمایانگر نرخ یادگیری[3] نزول گرادیان افکنده است. نکته قابل توجه در این رابطه آن است که عملیات افکنش حول ϵـ کره‌ی نمونه سالم اولیه x انجام می‌شود(و نه روی x^t). بدین ترتیب پس از چندین گام به حمله‌ای خواهیم رسید که طبق تعریف، شرایط یک حمله تخاصمی را ـ در صورت امکان ـ برآورده خواهد کرد.

در جدول ۲-۲ مقایسه‌ای اجمالی میان حملات بررسی شده، آمده است.

جدول ۲-۲: مقایسه حملات FGSM، PGD و CW

	FGSM	PGD	CW
تعداد گام های لازم برای پیاده سازی حمله	یک گام	چند گام	چند گام
قوت حمله	نسبتاً کم ولی همچنان موثر	متوسط	زیاد
زمان اجرای حمله	بسیار سریع، چراکه حمله در یک گام صورت می پذیرد	کند، چون به چند گام از نزول گرادیان افکنده احتیاج است	کند تر، زیرا علاوه بر چندین گام نزول گرادیان، نیاز به انجام جستجوی دودویی برای بهینه سازی مقدار یکی از پارامترها است.
پیچیدگی پیاده سازی حمله	ساده، فقط نیاز به یکبار محاسبه گرادیان تابع هزینه مدل نسبت به ورودی اعمال شده به مدل است	نسبتاً پیچیده، نیاز به اعمال چندین گام نزول گرادیان افکنده است که کمی دشوارتر از نزول گرادیان معمولیست	پیچیده تر، علاوه بر استفاده از تابع هزینه مخصوص، به پیاده سازی جستجوی دودویی نیز احتیاج است
نوع دسترسی مهاجم	جعبه سفید	جعبه سفید	جعبه سفید

[1] Gradient Descent
[2] Projected Gradient Descent
[3] Learning Rate

۲-۳ روش های دفاع در برابر حملات تخاصمی

دفاع در برابر حملات تخاصمی از دو جنبه کلی مورد بررسی قرار می‌گیرد [۱۰]:

۱. **پاکسازی**[1]: در این حالت، هدف پاکسازی ورودی‌های دسته‌بند و یا مقاوم‌سازی دسته‌بند از لحاظ ساختاری در راستای اصلاح خروجی آن در هنگام بروز حمله است به طوری که به خروجی در زمان های دیگر آسیبی وارد نشود.

۲. **تشخیص**[2]: در این حالت - که تمرکز اصلی این پژوهش است- هدف صرفاً تشخیص حمله‌ی تخاصمی پیش از ورود آن به مدل و اتخاذ تصمیم بر اساس خروجی اشتباه احتمالی دسته‌بند است. این امر عموماً با یک دسته‌بند مجزا برای تشخیص حملات صورت می‌گیرد.

ابتدا مختصری راجع به روش های پاکسازی و سپس در مورد روش های تشخیص بحث خواهد شد.

۲-۳-۱ روش های پاکسازی حمله

در ادبیات مرتبط با دفاع در برابر حملات تخاصمی، عبارت **پاکسازی** به دسته خاصی از روش‌های ممکن برای دفاع اطلاق می‌شود. در این پژوهش، برای سهولت دسته‌بندی، کمی از تعریف رسمی پاکسازی تخاصمی[3] دور شده‌ایم و آن را مطابق تعریف ارائه شده در بخش ۲-۳ در نظر می گیریم. بنابراین پاکسازی تخاصمی خود در این دسته قرار خواهد گرفت. با این مقدمه، روش های پاکسازی را می‌توان در ابعاد زیر دسته‌بندی نمود [۵، ۶، ۱۰]:

۱. **آموزش تخاصمی**[4]: یکی از ابتدایی ترین روش‌های دفاع که می‌تواند قابل اعمال روی هر شبکه عصبی مورد حمله باشد. در [۱۳] پیشنهاد شده است که انجام فرایند آموزش روی ترکیبی از نمونه‌های سالم و نمونه‌های تخاصمی می‌تواند باعث منظم‌سازی[5] شبکه عصبی آموزش دیده و پرورش توانایی برای مقابله در برابر حملات تخاصمی باشد. در [۹] این نظریه با اعمال حمله FGSM روی مجموعه داده‌های آموزشی و افزودن این نمونه‌ها به مجموعه داده و سپس آموزش نهایی مدل روی این مجموعه داده جدید، مورد آزمون قرار گرفته و تاثیر آن به طور تجربی ثابت شده است.

همانطور که پیش‌تر توضیح داده شد، در [۱۵] روشی نوین برای آموزش تخاصمی ارائه شده‌است که در

---

[1] Purification
[2] Detection
[3] Adversarial Purification
[4] Adversarial Training
[5] Regularization

آن تابع ضرر[1] به فرم یک مسئله بهینه‌سازی کمینه-بیشینه[2] بیان شده است (رابطه ۲-۱۳) در این رابطه، مسئله بیشینه‌سازی درونی سعی می‌کند که شبیه فرمول‌بندی ارائه شده در بخش ۲-۲-۱-۲ برای حمله FGSM ، قوی‌ترین مقدار δی ممکن برای بزرگ کردن مقدار تابع هزینه پیدا شود. این در حالیست که در مسئله کمینه‌سازی برونی هدف کمینه کردن مقدار این تابع ضرر تخاصمی با تنظیم کردن پارامترهای مدل (θ) است. بدین ترتیب،اگر مقدار تلورانس[3] حمله را ϵ در نظر بگیریم، مدل با نمونه‌های تخاصمی در یک ϵ-کره حول هر نمونه‌ی سالم آموزش داده خواهد شد.

مهم‌ترین ایراد آموزش تخاصمی این که است که تولید نمونه‌های تخاصمی قوی در زمان آموزش -خصوصاً روی مجموعه داده های بزرگ مانند ImageNet - می‌تواند بسیار زمان بر باشد و بدین ترتیب اکثر روش‌های آموزش تخاصمی از حملات تک-مرحله‌ای مانند FGSM برای افزایش مجموعه‌داده های آموزشی استفاده می کنند. برای حل این مسئله روش Free Adversarial Training [۱۷] ارائه شد که از اطلاعات گرادیان مدل در زمان آموزش برای تولید حملات تخاصمی (PGD)استفاده می‌کند و بنابراین نیازمند تولید مجدد حمله نیست. همچنین در این روش، برای کاهش زمان همگرایی حمله PGD نویز تخاصمی بدست آمده برای یک دسته از ورودی‌های سالم، به عنوان نقطه آغازین حمله PGD روی دسته ورودی‌های بعد مورد استفاده قرار می گیرد. این دفاع روی حمله چند -مرحله‌ای PGD [۱۵] مورد آزمایش قرار گرفته و نتایج امیدوار کننده‌ای داشته است. همچنین در You Only Propagate Once (YOPO) [۱۸] این مسئله مورد بررسی قرار گرفته که نیاز به یک لایه دفاعی در برابر حملات تخاصمی را می‌توان تقریباً فقط به اولین لایه‌ی یک شبکه عمیق خلاصه کرد. بنابراین آموزش تخاصمی می تواند بسیار ارزان‌تر صورت بگیرد. این روش که مستقیماً با Free Adversarial Training مورد مقایسه قرار گرفته است، نشان می‌دهد که می‌تواند در زمانی کمتر دارای عملکرد مشابه باشد. از دیگر ایده‌های جالب توجه مطرح شده در این زمینه می توان به Geometry-aware Instance-reweighted Adversarial Training (GI-AT) [۱۹] و Fast Adversarial Training [۲۰] اشاره کرد. در GI-AT از این نکته بهره گرفته می‌شود که نمونه داده‌هایی که نزدیک مرزهای تصمیم‌گیری یک مدل قرار دارند، بیشتر می‌توانند در موفقیت یک حمله تخاصمی تاثیرگذار باشند. بدین ترتیب این روش با بهره‌گیری از راهکارهای استاندارد آموزش تخاصمی، هر یک از نمونه داده‌های آموزشی را بر مبنای این که تولید حمله‌ی تخاصمی موفق از روی آن‌ها چقدر دشوار است، وزن‌دهی کرده و در فرایند پس‌انتشار[4] دخیل می کند. نهایتاً در Fast Adversarial Training این

[1] Loss
[2] Min-max
[3] Tolerance
[4] Backpropagation

نکته به صورت تجربی نشان داده می‌شود که آموزش تخاصمی با استفاده از حمله تک‌ـ مرحله‌ای FGSM ولی با آغاز تصادفی[۱] (بر خلاف [۱۷]) می تواند به اندازه آموزش تخاصمی با استفاده از حملات قوی‌تر چندـ مرحله‌ای (مانند PGD) اثر بخش باشد. در این پژوهش محققین موفق شده‌اند که در کسری از زمان گزارش شده در [۱۷] به نتیجه مشابه دست بیابند.

۲. **تغییرات در مدل قربانی و فرایند آموزش**: پژوهش [۲۱] یکی از اولین کار های انجام شده در این زمینه که با فاصله بسیار اندکی از کشف حملات تخاصمی و آسیب‌پذیری شبکه های عصبی به این حملات، صورت گرفت. با وجود این که حملات تخاصمی ایجاد شده پس از این مقاله، می توانند به راحتی راهکارهای ارائه شده در این کار را دور بزنند، ایده‌های مطرح شده همچنان شایان ذکر به نظر می‌رسند. ایده‌ی اصلی این پژوهش ارائه روش های آموزش جدید به گونه‌ایست که نمونه‌های تخاصمی تولید شده توسط حمله L-BFGS [۱۳] دارای اعوجاج[۲] بیشتری نسبت به نمونه‌های متناظر سالم دارند و دیگر قابل صرف نظر نیستند. در این پژوهش سه روش پیش پردازش برای مقابله با حمله L-BFGS مورد بررسی قرار گرفته‌اند:

(آ) *تزریق نویز* : افزودن نویز گاوسی اندک به ورودی‌ها می‌تواند به تشخیص درست تعداد بیشتری از نمونه‌های تخاصمی در ازای مقدار اندکی هدر رفت دقت دسته‌بندی بیانجامد.

(ب) *خودرمزگذار*[۳] : در این روش یک خودرمزگذار با هدف بازسازی نمونه‌های سالم از روی نمونه‌های تخاصمی، آموزش داده شده است که می تواند از حمله جلوگیری کند.

(ج) *خودرمزگذار نویز گیر* : در این روش، شبیه روش قبلی، یک خودرمزگذار نویز گیر معمولی بدون دانش پیشین از توزیع آماری نویز تخاصمی و صرفاً با هدف از بین بردن نویز آموزش داده شده است. سپس در زمان آموزش، هر پیکسل از نمونه‌ی آموزشی با یک نویز گاوسی با میانگین 0 و انحراف معیار متغیر σ ترکیب می شود. نشان داده شده است که با قرار دادن $\sigma = 0.1$ این خودکذگذار می تواند به خوبی خودگذار قبلی عمل کند.

ایراد اصلی وارد به موارد ۲ب و ۲ج این است که با سری کردن خودرمزگذار و مدل قربانی و حمله به مدل سری شده، همچنان می‌توان حمله موفقی داشت. این در حالیست که روش ۲آ نمی توانست در برابر حملات جعبه‌سفید قوی‌تر که بعدها ارائه شدند (مانند PGD و CW) مقاومت کند.

ایده‌ی مهم دیگری که کمی بعدتر ارائه شد، ایده‌ی تقطیر دفاعی[۴] [۲۲] است. همانطور که در شکل ۲-۷

[1] Random Initialization
[2] Distortion$(x, \hat{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$ where $x, \hat{x} \in \mathbb{R}^n$
[3] Autoencoder
[4] Defensive Distillation

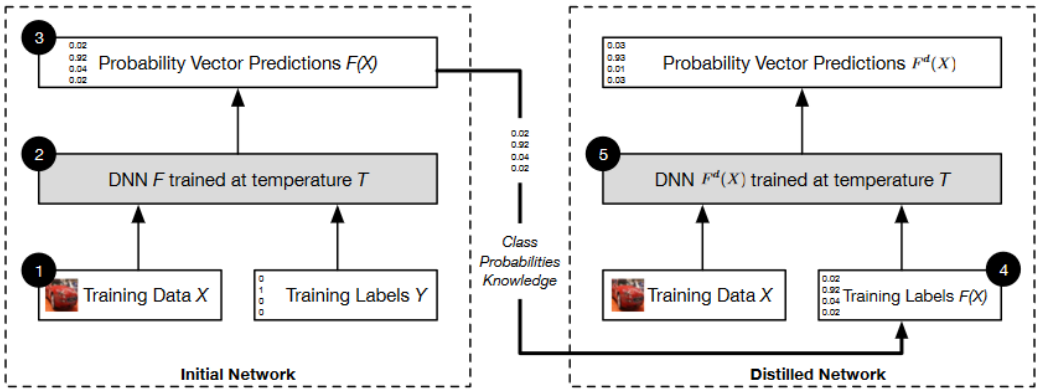
نشان داده شده است، ایده‌ی اصلی در این روش استفاده از پیش‌بینی های یک مدل از پیش آموزش داده شده روی مجموعه‌داده‌های سالم، به عنوان برچسب های جدید مدل تقطیر شده است. بدین ترتیب، مدل تقطیر شده برخلاف مدل اصلی دارای برچسب هایی با مقادیر پیوسته هستند که این امر آن‌ها را نسبت به حملات تخاصمی مقاوم‌تر می‌سازد. این روش در برابر حمله CW ناتوان است ولی با توجه به موفقیت آن در برابر گستره نسبتاً وسیعی از حملات دیگر، در سال های آینده ایده‌های قوی‌تری بر مبنای همین پژوهش ارائه شدند که مطالعه آن‌ها به خواننده واگذار می‌شود [۲۳–۲۵] .

در [۲۶] روشی تحت عنوان "منظم‌سازی گرادیان" ارائه شده است. ایده‌ی کلی در این روش اعمال یک جمله منظم‌سازی به تابع هزینه آموزش مدل است به طوری که هدف آموزش را می‌توان به صورت زیر بازنویسی کرد:

$$\arg\min_{\theta} H(y,\hat{y}) + \lambda \left\| \nabla_x H(y,\hat{y}) \right\|_2^2$$

که در آن y و $\hat{y}$ به ترتیب برچسب حقیقی و پیش‌بینی مدل و λ پارامتر تنظیم کننده میزان جریمه منظم‌سازی می باشد. هدف این روش (که با افزودن جمله منظم‌ساز شامل $\nabla_x H(y,\hat{y})$ محقق می‌شود)این است که از موضوع اطمینان حاصل شود که در صورت ایجاد تغییرات اندک در یک نمونه ورودی، دیورژانس KL بین پیشبینی مدل و برچسب واقعی تغییر چندانی نخواهد کرد و بنابراین نمونه‌های تخاصمی که در یک ϵـ کره محدود می شوند، نخواهند توانست خروجی مدل قربانی را تغییر دهند.

در روش DeepCloak [۲۷] که در شکل ۲-۸ نمایش داده شده، ایده مطرح شده آن است که با قرار دادن یک لایه ماسک‌کننده دقیقاً قبل از لایه خطی که تولید کننده logit های مدل، ویژگی های بی‌اهمیت در خروجی نهایی مدل را از صفر کرده و آن را نسبت به نویز تخاصمی مقاوم‌تر ساخت. آموزش این لایه با دادن ورودی‌های سالم و تخاصمی به مدل و encode کردن اختلاف بین ویژگی‌های آن ها در لایه پیشین،



شکل ۲-۷: چارچوب ارائه شده در روش تقطیر دفاعی [۲۲]

صورت می گیرد. قوت این روش در برابر حمله FGSM به صورت تجربی نشان داده شده است.

در ایده‌ای مشابه پژوهش قبلی و ترکیب آن با [۲۱] ، در Random Self-Ensemble [۲۸] یک لایه نویز برای افزودن مقدار ناچیزی نویز ایزوتروپیک گاوسی به ورودی‌های مدل و نیز خروجی های لایه‌های پنهان درون مدل قرار داده می شود:

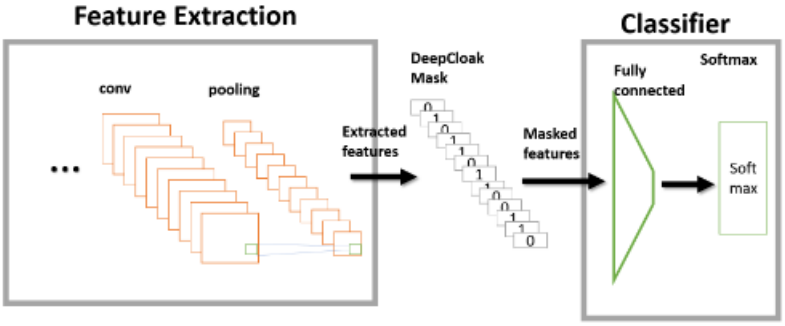
$$NoiseLayer(x) \rightarrow x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I).$$

مقادیر بزرگتر σ در این لایه مقاومت در برابر حملات تخاصمی را در ازای افت دقت دسته‌بندی، افزایش می‌دهند. برای کاهش تاثیر منفی σ از این لایه هم در زمان آموزش (با استفاده از تکنیک reparameterization) و هم در زمان تست،استفاده می‌شود. سپس، با تغییر مقدار σ - و به تبع آن، مقدار نویز افزوده شده به خروجی لایه‌ها، ϵ- می توان بدون overhead اضافی، عملاً به ensemble ای از مدل‌ها دست پیدا کرد که بر اساس آن‌ها می‌توان تصمیم‌گیری‌های دقیق‌تری نسبت به نمونه‌های تخاصمی انجام داد.

۳. **استفاده از شبکه‌های جانبی**: در [۲۹] دو علت اصلی برای اشتباه دسته‌بندها در مواجهه با یک نمونه تخاصمی برشمرده شده است:

(آ) نمونه تخاصمی از مرز های خمینه[1] وظیفه مورد نظر دور است. به عنوان مثال اگر وظیفه تشخیص اعداد دست نویس و مجموعه‌داده MNIST را در نظر بگیریم، یک نمونه تخاصمی ممکن است تصویری باشد که اصلاً شامل یک عدد دست‌نویس نیست ولی دسته‌بند از آنجایی که مجبور به تولید خروجی است، دچار اشتباه خواهد شد.

(ب) نمونه تخاصمی به مرزهای خمینه وظیفه مورد نظر خیلی نزدیک است. در این حالت که اکثر



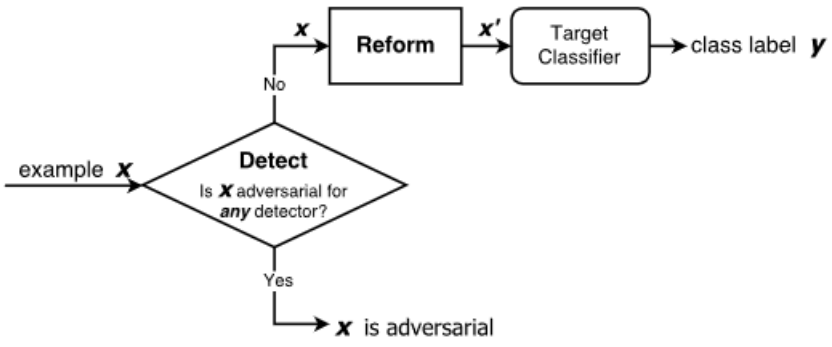
شکل ۲ - ۸: نحوه عملکرد DeepCloak [۲۷]

[1] Manifold

حملات تخاصمی جدید از آن استفاده می کنند، اگر دارای یک دسته‌بند باشیم،دسته‌بندی که در فضای اطراف یک نمونه تخاصمی قدرت تعمیم‌پذیری کمی دارد، در مواجهه با این نمونه خروجی اشتباه تولید خواهد کرد.

با توجه به دلایل ارائه شده، در این پژوهش چارچوبی به نام MagNet ارائه می شود. برای دفاع در برابر علت اول مطرح شده، MagNet از چندین شبکه تشخیص‌دهنده استفاده می‌کند برای آن که فاصله یک نمونه‌ی زمان تست را با نمونه‌های آموزشی بسنجد. به عبارت دقیق تر، یک تشخیص‌دهنده تابع $f : \mathcal{X} \to (0, 1)$ را یاد می گیرد که در آن $\mathcal{X}$ مجموعه تمام نمونه‌های زمان تست است و خروجی این تشخیص‌دهنده معیاری از فاصله نمونه زمان تست با خمینه نمونه‌های سالم زمان آموزش است. و سپس برای برطرف کردن مشکل دوم، این چارچوب از یک بهساز[1] برای تصحیح نمونه‌های دور از خمینه نمونه‌های سالم استفاده می شود. این شبکه بهساز توسط یک خود_رمزگذار پیاده‌سازی می شود. در صورت تشخیص نمونه تخاصمی توسط حداقل یکی از تشخیص دهنده‌ها، این نمونه به شبکه بهساز داده می شود و خروجی شبکه بهساز در نهایت به دسته‌بند می رسد (شکل ۲ - ۹).

یکی از مهم‌ترین ایده‌های مطرح شده در زمینه استفاده از شبکه‌های جانبی چارچوب Defense-GAN [۳۰] است. در این پژوهش برای پاکسازی یک نمونه تخاصمی از یک شبکه مولد تخاصمی[2] بهره برده می‌شود. در این چارچوب که در شکل ۲ - ۱۰ نمایش داده شده است، ابتدا یک WGAN روی مجموعه‌ای از داده‌های سالم آموزش داده می‌شود [3] . سپس در زمان تست، پیش از ورود یک نمونه به دسته‌بند، با استفاده از شبکه مولد GAN آموزش دیده شده، این نمونه به خمینه‌ی توزیع احتمالی یادگیری شده توسط



شکل ۲ - ۹: نحوه‌ی عملکرد MagNet [۲۹]

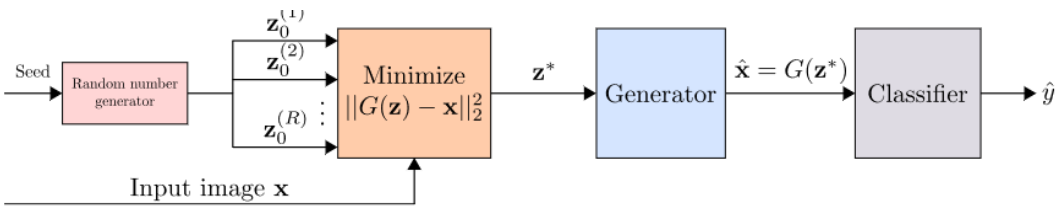
[1] Reformer
[2] Generative Adversarial Network (GAN)
[3] برای توضیحات مقدماتی راجع به GAN ها، می‌توانید به بخش ۲ - ۴ - ۱ مراجعه کنید

مولد، افکنده می‌شود. به عبارت دقیق تر، ابتدا R نمونه بردار نهفته[1] به صورت تصادفی تولید می‌شوند. سپس با استفاده از نزول گرادیان، تمام این R بردار پنهان در کمینه کردن تابع هزینه

$$\|G(z) - x\|_2^2$$

شرکت داده می شوند. بدین ترتیب، بردار برتر z^* که $G(z^*)$ نزدیک ترین نمونه‌ی ساختگی ممکن توسط مولد به نمونه‌ی احتمالاً تخاصمی ورودی یافته خواهد شد و در نهایت $\hat{x} = G(z^*)$ به جای نمونه‌ی اصلی (x) به دسته‌بند داده خواهد شد. هدف از این افکنش آن است که با توجه به این که GAN در زمان آموزش روی نمونه‌های سالم آموزش داده شده است، یافتن z^* به ترتیب توضیح داده شده، به از بین بردن هرگونه نویز تخاصمی کمک خواهد کرد. یکی از نقاط قوت Defense-GAN ماهیت غیرخطی آن به دلیل وجود یک حلقه بهینه‌سازی نزول گرادیان در پروسه‌ی پیاده‌سازی مکانیزم دفاع می‌باشد. این امر، Defense-GAN را نسبت به حملات جعبه‌سفید مقاوم می‌سازد. همچنین، این چارچوب هیچ پیش‌فرضی راجع به نوع حمله تخاصمی مورد استفاده ندارد و از آنجایی که صرفاً با تخمین توزیع احتمالی نمونه‌های سالم کار می‌کند، می‌تواند برای دفاع در برابر هر حمله‌ای مورد استفاده قرار بگیرد.

مشابه این پژوهش، در APE-GAN[2] [۳۱] مجدداً از یک شبکه مولد تخاصمی برای پاکسازی نویز تخاصمی استفاده شده است. این چارچوب که در شکل ۲ - ۱۱ نمایش داده‌شده است، از یک خودرمزگذار به عنوان مولد و یک شبکه ممیز تشکیل شده است. هدف نهایی این چارچوب آن است که مولد G طوری آموزش ببیند که بتوان نویزهای ناچیز تخاصمی را از روی ورودی تخاصمی احتمالی، حذف کند، بدون آن که خروجی مدل در زمان دریافت ورودی‌های سالم دچار تغییر محسوسی بشود. برای تحقق این هدف از یک ساختار تخاصمی و برقراری یک بازی خصمانه بین این G و یک مدل ممیز D استفاده می‌شود. وظیفه D در این وضعیت تشخیص دادن نمونه‌های پاکسازی‌شده توسط G از نمونه‌های سالم متناظر است. در نهایت این مولد قادر به تولید نمونه‌های پاکسازی شده‌ای خواهد بود که توسط ممیز غیر قابل تشخیص

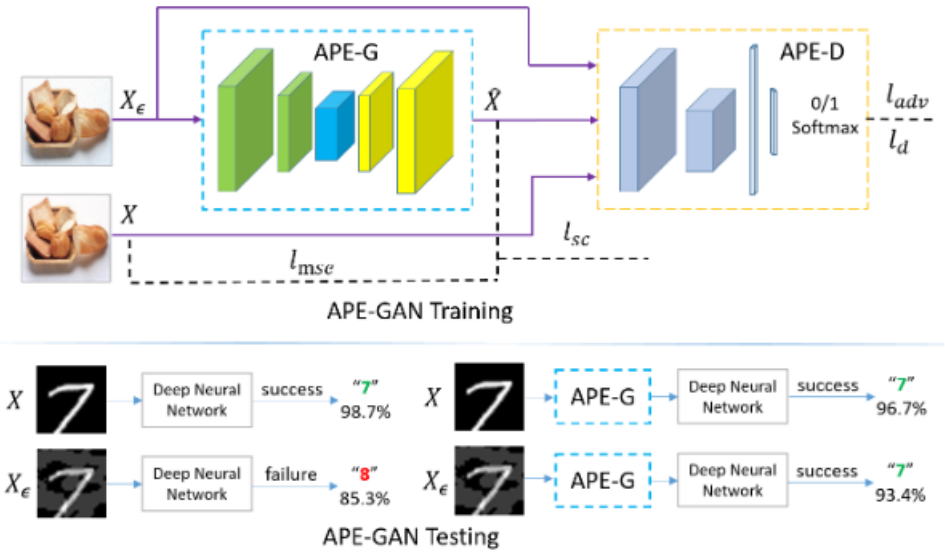


شکل ۲ - ۱۰ : دورنمای عملکرد چارچوب Defense-GAN [۳۰]

[1] Latent
[2] Adversarial Perturbation Elimination Generative Adversarial Network

هستند و در نتیجه تغییرات ایجاد شده در آن‌ها نسبت به ورودی اصلی، ناچیز است. برای حصول اطمینان از ناچیز بودن تغییرات اعمال شده توسط G چندین جمله به تابع هزینه G افزوده می‌شود که پایداری فرایند آموزش مولد را بهبود ببخشد. در نهایت در زمان تست، تمامی ورودی‌ها از مولد G عبور کرده و سپس به مدل قربانی تحویل داده می‌شوند.

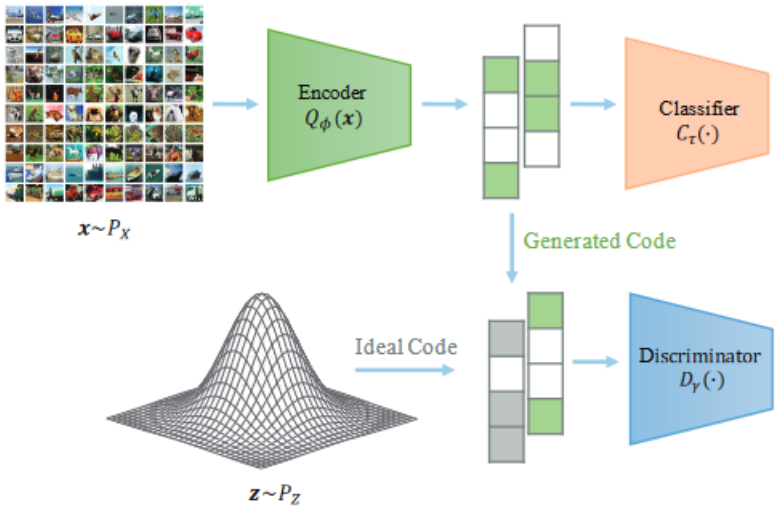
در [۳۲] استفاده از یک دسته‌بند با Embedding منظم شده (Embedding Regularized Classifier) پیشنهاد شده است. ایده‌ی اصلی این پژوهش آن است که نمونه‌های تخاصمی و نمونه‌های سالم از توزیع‌های احتمالی متفاوتی تولید می‌شوند، بدین ترتیب، اگر بتوانیم به نحوی یک قسمتی از دسته بند را که موظف به استخراج ویژگی از ورودی پیش از انجام دسته‌بندی است، به سمت تولید ویژگی‌هایی از توزیع سالم ترغیب کنیم، دسته‌بند عملکرد بهتری در برابر نمونه‌های تخاصمی خواهد داشت. همان‌طور که در شکل ۲-۱۲ نشان داده‌شده است، هر دسته‌بند را می‌توان به صورت دو زیرشبکه متصور شد: یک Encoder که وظیفه آن استخراج ویژگی از ورودی دسته‌بند است و یک هِد دسته‌بند برای تولید خروجی کلاس نهایی. اکتون در این چارچوب توزیع بردار های نهفته تولید شده توسط شبکه Encoder با یک توزیع پیشین[1] مقایسه می‌شوند و توسط یک شبکه جانبی ممیز[2] این توزیع پیشین مطلوب و توزیع تولید شده توسط Encoder مورد مقایسه قرار می گیرند. بدین ترتیب شبکه Encoder به سمت ایجاد بردار های نهفته از توزیع مطلوب، منظم می‌شود و احتمال موفقیت حملات تخاصمی کاهش خواهد یافت.



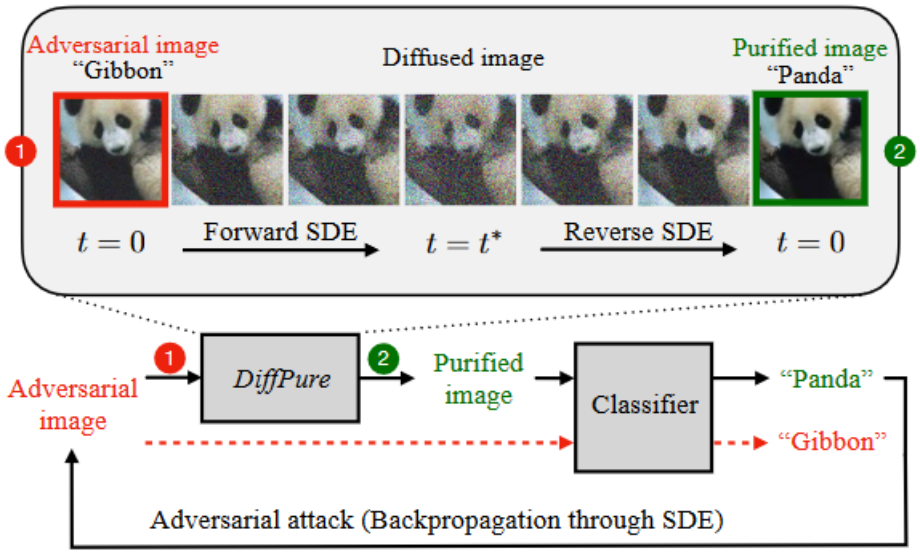
شکل ۲-۱۱: دورنمای APE-GAN در زمان آموزش و تست [۳۱]

[1] Prior
[2] Discriminator

یکی از جدیدترین ایده های مطرح شده در زمینه پاکسازی نمونه‌های تخاصمی DiffPure [۳۳] است. در این چارچوب از یک مدل انتشاری[۱] برای پاکساری نویز تخاصمی تزریق شده به یک تصویر استفاده می‌شود[۲] . نحوه عملکرد این چارچوب در شکل ۲-۱۳ نمایش داده شده است. مدل‌های انتشاری از دو فرایند انتشار پیش‌رو[۳] و انتشار معکوس[۴] تشکیل می‌شوند. ایده‌ی اصلی DiffPure آن است که با اعمال انتشار پیش‌رو روی ورودی تخاصمی احتمالی و به تبع آن افزودن مقداری نویز ایزوتروپیک گاوسی به آن،



شکل ۲-۱۲: دورنمایی از ER-classifier [۳۲]



شکل ۲-۱۳: نحوه عملکرد DiffPure [۳۳]

[1] Diffusion Model

[2] برای اطلاعات بیشتر راجع به مدل‌های انتشاری می‌توانید به بخش ۲-۴-۲ مراجعه کنید

[3] Forward Diffusion

[4] Reverse Diffusion

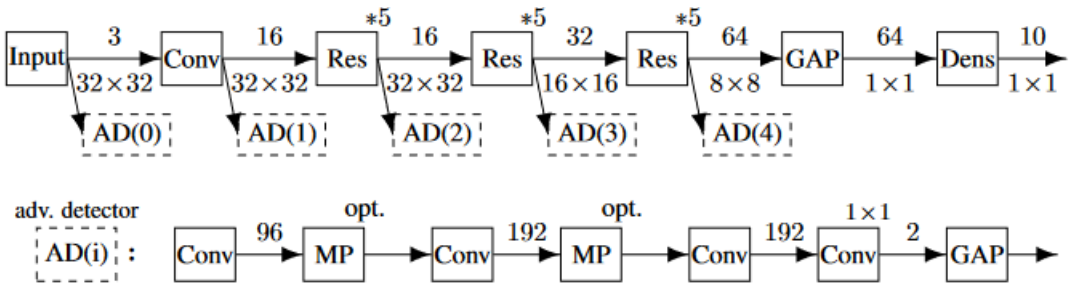
تاثیر نویز تخاصمی تزریق شده به ورودی از بین خواهد رفت. سپس برای آن که خروجی دسته‌بند از نویزی بیش از حد اعمال شده به ورودی دچار تغییر نشود، به همان تعداد گام پیش‌روی، فرایند انتشار معکوس انجام خواهد شد تا در نهایت دوباره به ورودی اصلی اما این بار بدون نویز تخاصمی دست یافته شود. این ایده به صورت تجربی و همچنین با شهود ریاضی در این مقاله مورد بررسی و به اثبات رسیده است.

۲-۳-۲ روش های تشخیص حمله

روش‌های تشخیص حمله را می‌توان عموماً از دو جهت بررسی کرد:

۱. **تشخیص‌دهنده‌های مبتنی بر ورودی یا خروجی‌های مدل قربانی**: ساده‌ترین روش تشخیص نمونه‌های تخاصمی، پیاده‌سازی یک دسته‌بند دو_کلاسه برای تمییز دادن نمونه‌های سالم از نمونه‌های تخاصمیست. این ایده در [۳۴] مورد بررسی قرار گرفته است. یکی از نقاط قوت اصلی این روش آن است که هیچ پیش‌فرضی راجع به مدلی که از آن دفاع می‌کند ندارد. ضعف اصلی ذکر شده برای این روش در همین مقاله، این است که چنین دسته‌بندی قدرت تعمیم‌پذیری کمی دارد و خصوصاً نسبت به نوع حمله مورد استفاده شده بسیار حساس است. در این پژوهش که از حملات FGSM و JSMA [۳۵] برای ارزیابی مدل استفاده شده است، نتایج تجربی نشان می‌دهند که تشخیص‌دهنده‌هایی که روی حملات FGSM شده‌اند، نمی‌توانند در برابر حملات JSMA به خوبی دفاع کنند و برعکس.

به طور مشابه در [۳۶] از یک زیر_شبکه تشخیص‌دهنده درون دسته‌بند استفاده می‌شود. بدین ترتیب که پس از یکی از لایه‌های پنهان[1] ، شبکه اصلی به دو شاخه تقسیم می شود و ویژگی های مستخرج از لایه‌ی پنهان قبلی به عنوان ورودی زیر شبکه تشخیص دهنده و نیز ادامه دسته‌بند، داده خواهند شد. این معماری در شکل ۲-۱۴ نشان داده شده است. برای آموزش این شبکه، ابتدا دسته‌بند اصلی روی نمونه‌های سالم آموزش داده خواهد شد. سپس، زیرشبکه‌های تشخیص دهنده در لایه‌های میانی به دسته‌بند اضافه



شکل ۲-۱۴: معماری تشخیص‌دهنده Metzen [۳۶]

[1] Hidden Layers

می‌شوند و نمونه‌های تخاصمی به تعداد برابر نمونه‌های آموزشی، تشکیل خواهند شد. در نهایت وزن‌های دسته‌بند اصلی freeze شده و فقط زیرشبکه‌های تشخیص‌دهنده روی نمونه‌های سالم و تخاصمی به عنوان یک دسته‌بند دوـ کلاسه، آموزش داده خواهند شد. در [۳۷] ایده‌ی مشابه ولی با ارتباط بین زیرشبکه‌های دسته‌بند با الهام گرفتن از ایده‌ی Boosting برای دسته‌بندهای آبشاری[۱] [۳۸]، مطرح شده است.

در [۳۹] چارچوبی تحت عنوان I-defender مطرح شده است که در آن توزیع احتمالی خروجی‌های لایه‌های کاملاً متصل[۲] یک مدل دسته‌بند در حین دسته‌بندی نمونه‌های سالم و تخاصمی مورد بررسی قرار گرفته است. در این پژوهش نشان داده شده است که توزیع احتمالی لایه‌های پنهان کاملاً متصل نه تنها برای کلاس‌های مختلف مجموعه‌داده آموزش، تفاوت دارد، بلکه برای یک کلاس در حالت سالم و تخاصمی نیز دارای تفاوت‌های چشمگیری هستند. با الهام گرفتن از این نتیجه، محققین این پژوهش ایده‌ی تخمین زدن این توزیع‌های احتمالی را با استفاده از GMM[۳] مطرح می کنند. بدین ترتیب برای هر کلاس از کلاس‌های مجموعه داده مورد استفاده،

$$p(\mathcal{H}(x)|\theta, c) = \sum_{k=1}^{K} w_i \mathcal{N}(\mathcal{H}(x)|\mu_{ck}, \Sigma_{ck})$$

احتمال بروز توزیع احتمالی لایه پنهان کاملاً متصل برای ورودی x به شرط پارامترهای مدل θ و کلاس c به صورت ترکیب وزن‌داری از توزیع های گاوسی تخمین زده می شود و سپس با یافتن یک حد آستانه TH_c برای هر کلاس، می توان از

$$Reject(x, c) = p(\mathcal{H}(x)|\theta, c) < TH_c$$

برای رد یا قبول کردن یک ورودی در زمان تست استفاده کرد.

در ML-LOO [۴۰] مفهومی تحت عنوان Feature Attribution برای تشخیص حملات تخاصمی مطرح می‌شود. این مفهوم که برای هر ورودی $x \in \mathbb{R}^d$ با $\Phi(x) : \mathbb{R}^d \to \mathbb{R}^d$ نشان نمایش داده شده است، معیاری از میزان تاثیر هر ویژگی از ورودی (برای تصاویر، هر پیکسل از تصویر) در خروجی دسته‌بندی نهاییست. برای اندازه‌گیری Φ از روش Leave-One-Out(LOO) استفاده می شود، بدین ترتیب که به ازای هر ویژگی از ورودی اختلاف احتمال خروجی محتمل‌ترین کلاس در حالت عادی و در زمانی که آن ویژگی با یک مقدار مرجع (مثلاً ۰) جایگزین شده است، محاسبه می‌شود. به عبارت دقیق تر:

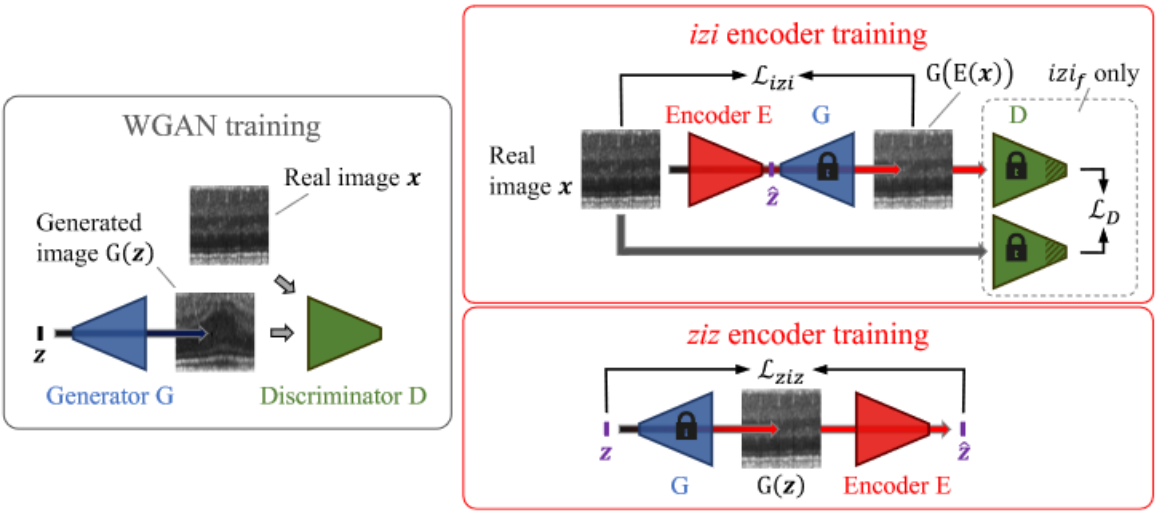
$$\Phi(x)_i := f(x)_c - f\left(x_{(i)}\right)_c, \quad \text{s.t.} \quad c = \arg\max_{j \in C} f(x)_j .$$

[1] Cascade
[2] Fully Connected
[3] Gaussian Mixture Model

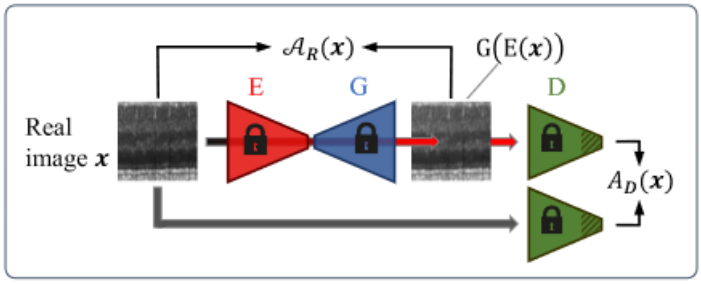
محققین این پژوهش مشاهده کردند که در نمونه‌های تخاصمی و سالم تفاوت قابل توجهی بین مقدار Φ آن‌ها وجود دارد. از این تفاوت به عنوان معیاری برای تشخیص حملات استفاده می‌شود.

f-AnoGAN [۴۱] چارچوب جالب توجه دیگری با استفاده شبکه‌های مولد تخاصمیست که برای منظور خاص تشخیص آنومالی‌ها در تصاویر پزشکی به کار رفته است ولی ایده‌ی اصلی آن می‌تواند قابل تعمیم به هر محیطی باشد. در این روش که در شکل ۲-۱۵ آمده است، ابتدا یک WGAN[1] روی داده‌های سالم آموزش داده می‌شود. سپس وزن‌های شبکه‌های مولد و ممیز Freeze می شوند و یک رمزگذار به کمک این دو در دو مرحله آموزش می بیند(مراحل izi_f و ziz در شکل). در نهایت برای تشخیص آنومالی از ترکیب دو سنجه‌ی $A_R(x)$ و $A_D(x)$ استفاده می شود (شکل ۲-۱۶):

$$A(x) = A_R(x) + \kappa \cdot A_D(x)$$



شکل ۲-۱۵: دورنمای نحوه آموزش f-AnoGAN [۴۱]



شکل ۲-۱۶: نحوه عملکرد f-AnoGAN در زمان تست [۴۱]

[1] Wasserstein Generative Adversarial Network

که در آن

$$A_R(x) = \frac{1}{n} \cdot \|x - G(E(x))\|^2$$

معیاری از خطای بازسازی[1] است چرا که انتظار می رود پس از آموزش رمزگذار E روی داده‌های سالم، در حالت ایده‌آل، شبکه‌های G و E توابع عکس نظیر یک دیگر باشند و بنابراین

$$G(E(x)) \approx x.$$

ولی اگر در نمونه x آنومالی وجود داشته باشد، از آنجایی که E فقط می تواند به فضای پنهان نمونه‌های سالم رمز کند، نُرم اختلاف دو مقدار x و $G(E(x))$ می تواند سنجه خوبی برای تشخیص آنومالی باشد. از طرف دیگر

$$A_D(x) = \frac{1}{n_d} \cdot \|D_{interm}(x) - D_{interm}(G(E(x)))\|^2$$

که در آن $D_{interm}(\cdot)$ ویژگی های یکی از لایه‌های میانی شبکه ممیز است، با استدلالی مشابه $A_R(x)$ و با الهام گرفتن از روش تطبیق ویژگی[2] [۴۲] به عنوان سنجه‌ی دیگر برای تشخیص آنومالی استفاده می شود. در نهایت ترکیب این دو سنجه امتیاز $A(x)$ را بدست خواهد داد که از آن و با کمک یک حد آستانه تنظیم‌شده می‌توان برای تشخیص آنومالی استفاده کرد.

در نهایت نگاهی به روش ارائه شده در [۴۳] خواهیم انداخت [3] . در این روش - که نزدیک ترین کار انجام شده به کار ماست- بار دیگر از شبکه‌های مولد تخاصمی، ولی این بار مشروط بر کلاس[4] کمک گرفته شده است. در ACGAN-ADA ابتدا یک شبکه ACGAN [۴۴] روی نمونه‌های سالم آموزش داده می‌شود. سپس از تمام بخش های این شبکه برای ایجاد سنجه‌هایی برای تشخیص حمله استفاده می شود. به طور مشخص، سنجه‌های مورد استفاده موارد زیر هستند:

$$S_R = D(x)$$

$$S_C = p_D(\hat{c}|x)$$

$$S_g = \min_z \|x - G(z|\hat{c})\|^2$$

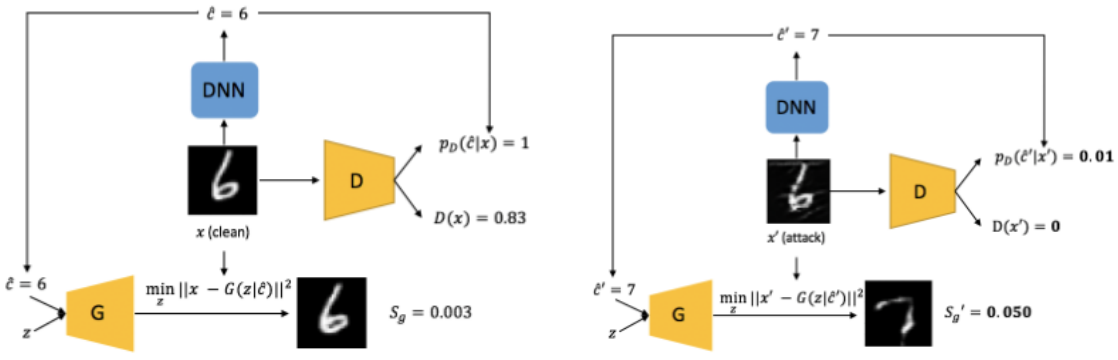
که در آن $\hat{c}$ خروجی موقت دسته‌بند مورد دفاع نسبت به ورودی اعمال شده است. S_R خروجی زیرشبکه ممیز ACGAN و معیاری از واقعی بودن نمونه ورودی با توجه به مشاهدات صورت گرفته در زمان آموزش

[1] Reconstruction Error
[2] Feature Matching
[3] برای ارجاع دادن در این روش و با توجه به این که محققین اسم خاصی در مقاله به آن نسبت نداده‌اند، از این جا به بعد این روش را ACGAN-ADA خطاب خواهیم کرد
[4] Class Conditional

از داده‌های سالم است. این سنجه حملاتی را که نویز تخاصمی آن‌ها دارای نُرم بزرگی باشد، جریمه می‌کند. S_C احتمال پسین[1] کلاس تشخیص داده شده توسط مدل مورد دفاع، به شرط ورودی داده شده ولی از دسته‌بند اضافی[2] شبکه‌ی ACGAN است. مشخص است در صورتی که مقدار S_C کم باشد احتمال بروز حمله وجود دارد چرا که بین دسته‌بند اضافی و مدل مورد دفاع روی کلاس نهایی توافق کمی وجود دارد. در نهایت S_g کمینه‌ی نرم افکنش ورودی به شرط کلاس تشخیص داده شده توسط مدل مورد دفاع است. با استدلالی همانند روش ارائه شده در Defense-GAN، کوچک بودن این مقدار به منزله احتمال بیشتری برای سالم بودن ورودیست.نحوه عملکرد این روش در شکل ۲-۱۷ نشان داده شده است. در حین اعمال ورودی سالم (شکل ۲-۱۷ب) مقدار $D(x)$ و $p_D(\hat{c}|x)$ بالا بوده در حالی که مقدار S_g ناچیز است. در مقابل، در حین اعمال ورودی تخاصمی (شکل ۲-۱۷آ) دقیقاً برعکس این حالت اتفاق خواهد افتاد. در ادامه این پژوهش ترکیب های مختلفی از این سنجه‌ها را مورد استفاده قرار داده است و بهترین حالت ممکن گزارش شده، حالت D-AD است که فقط از سنجه‌های مربوط به زیرشبکه ممیز (S_R و S_C) استفاده می‌کند.

۲. **تشخیص‌دهنده‌های مبتنی بر ویژگی‌های خاص نمونه‌های تخاصمی**: منظر دیگری که می‌توان برای تشخیص نمونه‌های تخاصمی اتخاذ کرد آن است که به جای استفاده از ورودی‌ها و یا خروجی‌های مدل مورد دفاع، به طور مشخص از خواص نمونه‌های تخاصمی برای تشخیص آن‌ها استفاده کنیم.

برای تشخیص نمونه‌های تخاصمی که دور از خمینه نمونه‌های واقعی قرار دارند روشی تحت عنوان KD-Detection [۴۵] پیشنهاد شد. در این روش از تخمین چگالی هسته‌ای[3](KDE) برای تخمین چگالی



(ب) در حین اعمال ورودی سالم

(آ) در حین اعمال ورودی تخاصمی

شکل ۲-۱۷: عملکرد چارچوب ACGAN-ADA در زمان تست

[1] Posterior
[2] Auxiliary Classifer
[3] Kernel Density Estimation

نمونه‌های سالم از روی ویژگی‌های استخراج شده توسط آخرین لایه پنهان برای کمک به تشخیص نمونه‌های تخاصمی استفاده می‌شود. به طور کلی، تخمین چگالی به عنوان معیاری از فاصله یک نمونه از یک خمینه هدف مورد استفاده قرار می‌گیرد. فرض کنید

$$x_1, x_2, \cdots, x_n$$

نمونه‌هایی از نمونه‌های آموزشی هستند که از توزیع احتمالی مجهول و احتمالاً بسیار پیچیده $p_X(x)$ نمونه برداری شده‌اند. به ازای هر ورودی x، می توان چگالی تخمین زده شده حول x را به صورت زیر محاسبه کرد:

$$\hat{p}_X(x) = \frac{1}{n}\sum_{i=1}^{n} K_\sigma(x, x_i),$$

که در آن $K_\sigma(\cdot,\cdot)$ یک تابع هسته است. در روش KD-Detection برای هر کلاس یک مدل چگالی هسته‌ای تخمین زده می‌شود به طوری که اگر x دارای کلاس **پیش‌بینی شده** y است، تنها نمونه‌های آموزشی دارای برچسب y برای تخمین چگالی هسته‌ای آن مورد استفاده قرار می‌گیرند. پس از آن که تمامی مدل های KDE روی نمونه‌های سالم آموزش دیده شدند، نمونه‌های زمان تست ابتدا به صورت موقت به دسته‌بند مورد دفاع داده می‌شوند و پس از مشخص شدن برچسب پیش‌بینی شده توسط این مدل (چه درست، چه غلط بر اثر حمله تخاصمی)، این ورودی به KDE متناظر فرستاده خواهد شد و در نهایت از یک مدل Logistic Regression برای تشخیص نمونه‌های تخاصمی از روی امتیاز چگالی آن نمونه خاص توسط KDE متناظر، استفاده می‌شود.

ایده‌ی دیگر مطرح شده در این زمینه، ADA [۴۶] است. بنای اصلی این پژوهش، این فرضیه است که نمونه‌های تخاصمی موفق در فضای مرز تصمیم ایجاد شده توسط دسته‌بند، به اندازه کافی به کلاس هدف شبیه هستند ولی از آنجایی که نُرم نویز تخاصمی اعمال شده به نمونه باید محدود شده باشد، انتظار است که این نمونه‌های تخاصمی از کلاس نمونه‌ی سالم متناظرشان نیز چندان دور نخواهند بود. در همین راستا، در این پژوهش با استفاده از مدل‌های تخمین چگالی، فضای تولید شده توسط لایه‌های پنهان دسته‌بند مورد دفاع را به صورت ریاضی مدل می‌کنند. سپس از دیورژانس[1] Kullback-Leibler (KL) [۴۷] بین معیارهای تخمین زده شده توسط خود دسته‌بند و مدل‌های تخمین چگالی و یک حد آستانه مناسب، برای تشخیص حملات استفاده می‌شود.

در [۴۸] روش دیگری تحت عنوان Mahalanobis Detector (MD) مطرح شده است. فرض کنید یک

[1] Divergence

دسته‌بند عصبی softmax آموزش دیده شده در اختیار داشته باشیم که احتمال پسین

$$P(y=c|x)=\frac{\exp\left(w_c^{\mathbf{T}}f(x)+b_c\right)}{\sum_{c'}\exp\left(w_{c'}^{\mathbf{T}}f(x)+b_{c'}\right)}$$

را تولید می کند و w_c و b_c وزن‌ها و بایاس‌های لایه آخر دسته‌بند، و $f(\cdot)$ نمایانگر خروجی لایه‌ی ماقبل آخر دسته هستند. اکنون بدون هیچ تغییری در دسته‌بند آموزش دیده شده، اگر فرض کنیم که توزیع احتمالی نمونه‌ها مشروط بر هر کلاس از یک گاوسی چند متغیره پیروی می‌کند، می توان یک دسته‌بند **مولد** تعریف کرد. به عبارت دقیق تر، می توان C توزیع گاوسی با کواریانس مشترک Σ را در نظر گرفت:

$$P(f(x)|y=c)=\mathcal{N}(f(x)|\mu_c,\Sigma)$$

که μ_c میانگین مختص به کلاس $c\in\{1,\cdots,C\}$ است. علت منطقی بودن این فرض آن است که می توان نشان داد چنین دسته‌بند مولدی تحت Gaussian Discriminant Analysis (GDA) با یک دسته‌بند softmax معادل است. اکنون برای تشکیل معیاری از سطح اطمینان نسبت به یک نمونه از فاصله ماهالانوبیس[1] بین نمونه x و نزدیک ترین گاوسی مشروط بر کلاس موجود از بین C گاوسی ممکن، استفاده می شود:

$$M(x)=\max_c -(f(x)-\mu_c)^{\mathbf{T}}\Sigma^{-1}(f(x)-\mu_c).$$

در نهایت با قرار دادن یک حد آستانه روی $M(x)$ می تواند تشخیص داد که نمونه مورد آزمون x سالم و یا دارای آنومالی می‌باشد.

در [۴۹] روش آماری دیگری برای تشخیص نمونه‌های تخاصمی ارائه شده است. اگر logit های پیش از اعمال softmax در یک دسته‌بند با بردار $f(\vec{x})$ نشان داده شوند، $f_y(x)$ درایه‌ی yـام این بردار خواهد بود که به عبارتی دیگر، log-odds برای کلاس y نیز پنداشته می‌شود. همچنین فرض کنیم که مقادیر جفتی log-odds نیز برای دو کلاس y و z به صورت زیر تعریف شوند:

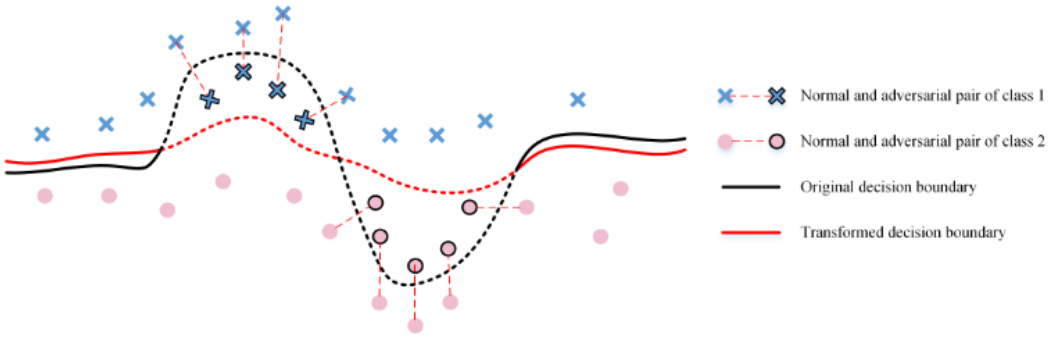
$$f_{y,z}(x)=f_z(x)-f_y(x) \qquad (۱۵-۲)$$

اکنون، ایده‌ی اصلی مطرح شده در این پژوهش آن است که این مقدار log-odds جفتی چگونه با افزودن مقدار کمی نویز η به ورودی x تغییر می کنند هنگامی که $y=y_{true}$ اگر برچسب‌های واقعی در اختیار باشند و هنگامی که $y=F(x)=\arg\max_y f_y(x)$ در زمان تست. یافته‌های این پژوهش نشان می دهند که این مقدار log-odds می تواند بسته به این که نمونه‌ی ورودی x سالم و یا تخاصمی بوده است، به شدت

[1] Mahalanobis Distance

متفاوت باشد. همچنین، اگر فرض شود که نویز تخاصمی افزوده شده به نمونه‌های سالم نسبت به نویز افزوده η مقاوم نیست، بنابراین اگر نمونه تخاصمی $\hat{x} = x + \delta$ در اختیار باشد که $F(\hat{x}) = \hat{y} \neq F(x) = y$ ،آنگاه انتظار می رود که $f_{\hat{y},y}(\hat{x} + \eta) > f_{\hat{y},y}(\hat{x})$ چرا که نویز کوچک η اضافه شده مقدار کمی از تاثیر نویز تخاصمی δ را از بین برده و مقدار log-odds جفتی بین کلاس به اشتباه تشخیص داده شده توسط دسته‌بند و کلاس اصلس را طبق معادله ۲-۱۵ افزایش می دهد. بنابراین از امید ریاضی این مقدار log-odds جفتی (روی η های مختلف) می‌توان به عنوانی معیاری برای تشخیص نمونه‌های تخاصمی با استفاده از یک حد آستانه مطلوب بهره برد.

در چارچوبی تحت عنوان SID[1] [۵۰] این ایده مطرح شده است که نمونه‌های تخاصمی از هل دادن نمونه‌های سالم به سمت دیگر مرز تصمیم در جاهایی که مرز تصمیم یک دسته‌بند دارای تلاطم زیادیست، تشکیل می‌شوند(شکل ۲-۱۸). برای حل این مشکل ایده‌ی مطرح شده آن است که مرز تصمیم دسته‌بند را طوری تغییر داد که دیگر نتوان با تغییرات اندک نمونه‌های سالم را به سوی دیگر مرز تصمیم هل داد و موجب تشکیل نمونه‌های تخاصمی شد بدون آن که مرز تصمیم در نقاط کم تلاطم دچار تغییرات چشم‌گیری بشود. به طور دقیق‌تر، اگر $\mathcal{B}_{i,j}$ مرز تصمیم بین دو کلاس i و j در دسته‌بند اصلی $(\mathcal{F})$ باشد، و $\tilde{\mathcal{B}}_{i,j}$ مرز تصمیم دسته‌بند جدید $(\mathcal{G})$ باشد، مطلوب آن است که $\mathcal{B}_{i,j}$ و $\tilde{\mathcal{B}}_{i,j}$ در همه‌ی نقاط، به جز در اطراف نقاط پر تلاطم $\mathcal{B}_{i,j}$ شبیه یک دیگر باشند که این مطلوب را می‌توان به تولید بردارهای احتمال مشابه در دو دسته‌بند در تمام نقاط غیر از نقاط مذکور، کاهش داد. این مسئله را می‌توان به صورت یک مسئله بهینه‌سازی کمینه‌ـبیشینه بیان کرد که هزینه متحمل شده در بدترین حالت اختلاف مرز تصمیم



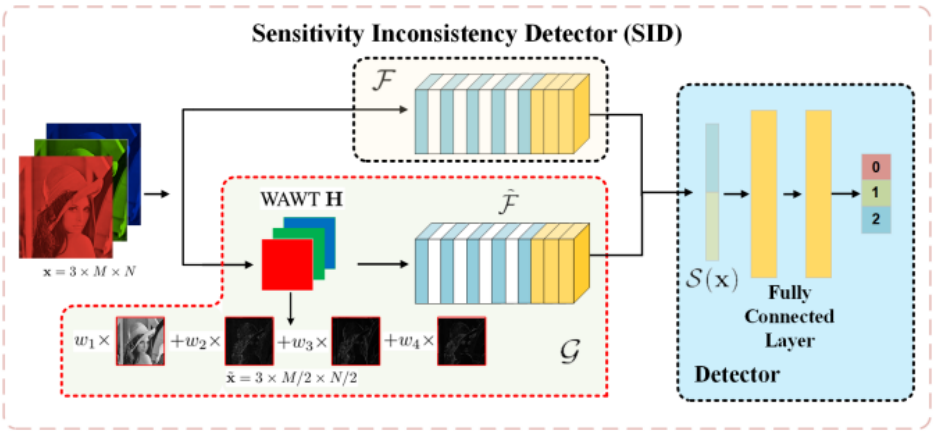
شکل ۲-۱۸: تشکیل نمونه‌های تخاصمی در اطراف خمیدگی‌های مرز تصمیم یک دسته‌بند [۵۰]

[1] Sensitivity Inconsistency Detector

ایجاد شده توسط $\mathcal{F}$ و $\mathcal{G}$ را حداقل می‌کند:

$$\min_{\mathcal{G}} \max_{x \in X} \|\mathcal{F}(x) - \mathcal{G}(x)\|_2^2, \quad \text{s.t.} \|\mathcal{F}(\hat{x}) - \mathcal{G}(\hat{x})\|_2^2 \geq \xi, \quad \forall \hat{x} \in \hat{X} \qquad (۱۶\text{-}۲)$$

که در آن X مجموعه نمونه‌های سالم، $\hat{X}$ مجموعه نمونه‌های تخاصمی و ξ یک متغیر لَنگی[1] است که به $\mathcal{G}$ اجازه می‌دهد در اطراف نقاط پر تلاطم مرز تصمیم $\mathcal{F}$ به بتواند به اندازه کافی از آن فاصله بگیرد. برای آن که دسته‌بند جدید بتواند مرز تصمیم مطلوب ارائه شده در معادله‌ی ۲-۱۶ را برآورده کند، در این پژوهش از تبدیل Wavelet با میانگین وزن‌دار[2] استفاده می‌شود. در نهایت، مطابق شکل ۲-۱۹ خروجی‌های دو دسته‌بند به یک شبکه تشخیص‌دهنده‌ی نهایی تحویل داده می‌شوند و انتظار می‌رود که بر اساس اختلاف مرز تصمیم‌های ایجاد شده توسط دو دسته‌بند، شبکه تشخیص دهنده بتواند سالم و یا تخاصمی بودن یک نمونه‌ی ورودی را تشخیص دهد.



شکل ۲-۱۹: عملکرد SID در زمان تست [۵۰]

[1] Slack Variable
[2] Weighted Average Wavelet Transform (WAWT)

۲-۴ مختصری در مورد هوش مولد

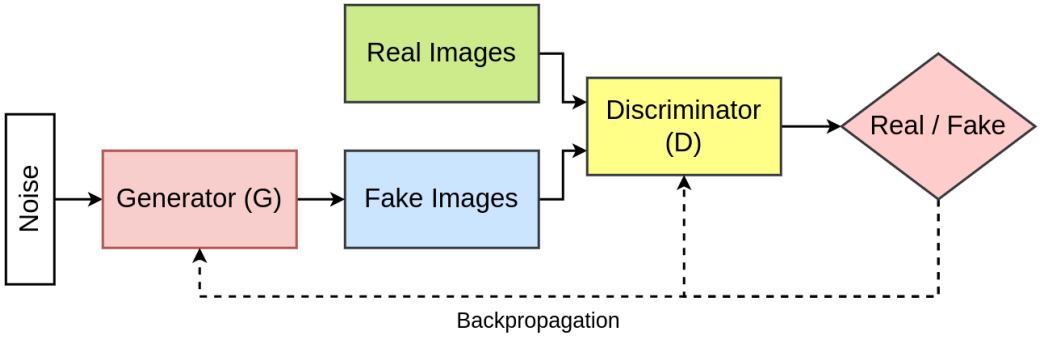
هوش مولد[1] عبارتیست که به یکی از زیر شاخه های هوش مصنوعی اطلاق می‌شود که تمرکز اصلی آن تولید رسانه‌های مختلف از جمله متن، تصویر، فیلم و مدل‌های سه‌بعدی و غیره است. در هوش مولد از محتوای موجود کنونی و برای آموزش مدل‌هایی استفاده می‌شود که توانایی خلق محتوای جدید را داشته باشند بدون آن که لزوماً نمونه‌های آموزشی را دقیقاً در خروجی تکرار کنند. در حقیقت مدل‌های مولد تخمینی پارامتری از توزیع احتمالی پیچیده محتوایی که قرار است تولید کنند را یاد میگیرند و در نهایت روشی برای نمونه گیری از این تخمین پارامتری به ما ارائه خواهند کرد.

در ادامه این بخش به بررسی دو مورد از اصلی ترین مدل‌های مولد در حوضه‌ی تصویر خواهیم پرداخت که همچنان به طور گسترده مورد پژوهش و تحقیق فعال هستند: شبکه‌های مولد تخاصمی و مدل های انتشاری.

۲-۴-۱ شبکه های مولد تخاصمی

شبکه‌های مولد تخاصمی (GAN) اولین بار در پژوهش معروف [۵۱] توسط آقای Goodfellow و همکارانش، در یکی از هوشمندانه‌ترین استفاده‌های نظریه‌ی بازی‌ها در هوش مصنوعی، معرفی شدند. بنای اصلی این شبکه‌ها به عنوان چارچوبی جدید برای یادگیری بدون نظارت[2] و به عنوان گام بعدی در مدل‌های مولد بعد از خود رمزگذارها، نهاده شده.

همانطور که در شکل ۲-۲۰ نشان داده شده است، ایده‌ی کلی GANها برقراری یک بازی مجموع‌ـصفر[3] بین دو شبکه به نام‌های مولد[4] و ممیز است. هدف شبکه‌ی مولد در حالت ایده‌آل تولید تصاویری از توزیع تصاویر حقیقی موجود است. برآورده کردن این هدف به کمک بازی ایجاد شده میان مولد و ممیز صورت می‌گیرد.



شکل ۲-۲۰: دورنمای شبکه‌های مولد تخاصمی

[1] Generative AI
[2] Unsupervised Learning
[3] Zero-sum
[4] Generator

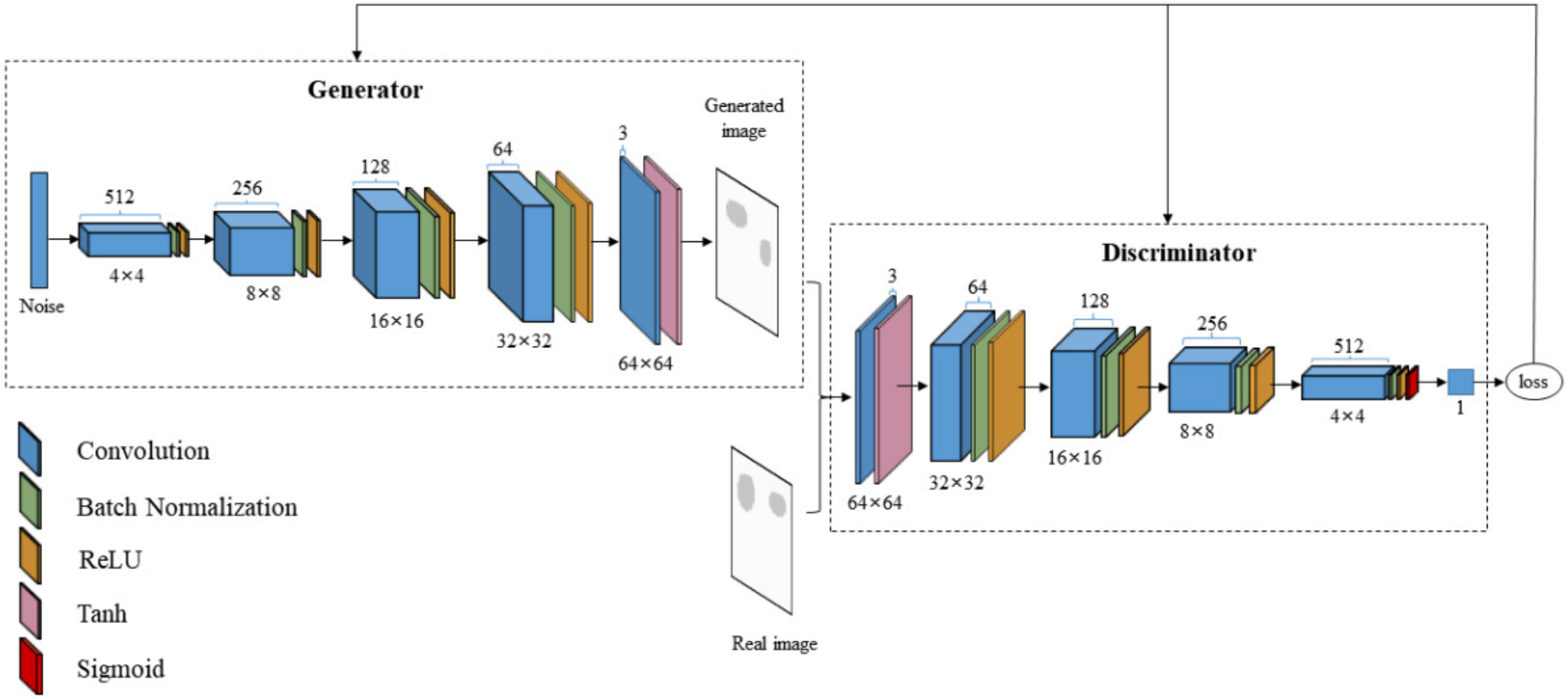
مولد تلاش می‌کند با تولید نمونه‌هایی که به نمونه‌های واقعی شبیه هستند، ممیز را فریب دهد. از طرف دیگر، ممیز در تلاش است که با تقویت خودش در برابر فریب خوردن از مولد مصون بماند و همچنان بتواند نمونه‌های ساختگی و واقعی را از هم تفکیک کند. به طور دقیق تر، مولد G و ممیز D بازی کمینه-بیشینه زیر را با تابع مقدار $V(D,G)$، انجام خواهند داد:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log(1-D(G(z)))\right] \qquad (۲\text{-}۱۷)$$

که در آن p_{data} توزیع احتمالی آرمانی نمونه‌های واقعی و p_z توزیع احتمالی بردار پنهانیست که مدل مولد بر پایه‌ی آن نمونه‌های جدید تولید می‌کند. اگر این بازی به صورت پایدار بین دو بازیکن مولد و ممیز بازی شود، در نهایت می توان نشان داد که با استفاده از فرمول بندی مناسب برای بازی، نقطه تعادل بهینه بازی در جایی اتفاق خواهد افتاد که توزیع آماری نمونه‌های تولید شده توسط مولد دقیقاً با توزیع نمونه‌های واقعی برابر باشد (قضیه ۱ در [۵۱]).

در ادامه به طور خاص نگاهی مختصر به شبکه‌های مولد تخاصمی عمیق کانولوشنی (DCGAN) خواهیم داشت که اولین نمونه شبکه‌های GAN با هدف تولید تصاویر بودند. همانطور که در شکل ۲-۲۱ مشاهده می شود، مولد یک DCGAN از سری کردن چندین لایه Upsampling (که در اولین نمونه‌های DCGAN صرفاً با استفاده از کانولوشن ترانهاده[1] پیاده سازی می‌شدند) به همراه تابع فعال ساز ReLU (به غیر از در آخرین لایه که از tanh استفاده می کند)به دست می‌آید. با شروع از یک بردار پنهان (یا نویز) که از یک توزیع تصادفی نمونه برداری می شود، در نهایت به یک تصویر سه-کاناله می رسیم. در مقابل شبکه ممیز ساختاری دقیقاً عکس مولد را اتخاذ می‌کند. با شروع از یک تصویر سه-کاناله و اعمال پی در پی لایه‌های کانولوشن به همراه فعال ساز ReLU و در نهایت یک sigmoid در لایه آخر، به خروجی ممیز خواهیم رسید که عملاً معیاری از احتمال واقعی بودن تصویر دریافت شده توسط ممیز است. شبکه‌های ممیز با گرفتن دسته‌ای از نمونه‌های واقعی و ساختگی مولد که دارای برچسب‌های به ترتیب ۱ و ۰ هستند، با استفاده از یک تابع هزینه Binary Crossentropy آموزش خواهد دید در حالی که شبکه‌ی مولد ثابت در نظر گرفته می‌شود. سپس، با ثابت در نظر گرفتن ممیز، شبکه مولد سعی می‌کند همان تابع هزینه را ماکزیمم کند تا ممیز را به اشتباه بیاندازد. بدین ترتیب دو شبکه به صورت متوالی تا رسیدن به همگرایی آموزش خواهند دید. در شکل ۲-۲۲ نمونه‌های تولید شده از این شبکه‌ی GAN را می توان مشاهده کرد. با وجود پیشرفت‌های ژرفی که در راستای بهبود GANها صورت گرفته است، این شبکه‌ها همچنان از سه ایراد اصلی رنج می‌برند [۵۲، ۵۳]:

[1] Transposed Convolution



شکل ۲-۲۱: معماری شبکه DCGAN

۱. **فروپاشی مُد**[1]: اگر در حین فرآیند آموزش به طور اتفاقی مولد بتواند نمونه‌ی ساختگی بسیار خوبی تولید کند، بازخوردی که از ممیز بابت این نمونه‌ی به خصوص دریافت می کند بسیار مثبت خواهد بود. به عبارت دیگر از آنجایی که این یک نمونه‌ی خاص احتمالاً می‌تواند ممیز را فریب دهد، مولد تشویق به تولید نمونه‌های مشابه این نمونه می‌شود. اگر هیچ مکانیزمی برای جلوگیری از این اتفاق وجود نداشته باشد، در بدترین حالت مولد یاد می‌گیرد که کل فضای حالت بردار پنهانی که به عنوان ورودی دریافت می‌کند را به یک نمونه یا چندین نمونه‌ی بسیار مشابه نگاشت کند. به چنین حالتی که مولد فقط یکی از مدهای توزیع احتمالی ورودی را یاد می‌گیرد، فروپاشی مد گفته می‌شود.

۲. **عدم همگرایی**: رسیدن به نقطه‌ی تعادل تعادل نَش[2] در GANها به دلیل ساختار بازی کمینه‌ـ بیشینه برقرار شده بین دو شبکه و پیچیدگی حل یک مسئله بهینه‌سازی نقطه زینی،امری آرمانیست. از آنجایی که دو بازیکن این بازی مجبورند به نوبت بازی کنند، باید تعادل دقیقی بین مولد و ممیز برقرار باشد تا بتوان به نقطه تعادل آرمانی نزدیک شد. در شرایط واقعی، کنترل کردن تمامی این پارامترها امری بسیار زمان بر و گاهاً غیر ممکن است و بنابراین ممکن است یک GAN هیچوقت به نقطه تعادل نرسد. در بدترین حالت این امکان وجود دارد که یکی از دو شبکه به واگرایی میل کند که به تبع کیفیت خروجی‌های نهایی مولد را به شدت کاهش خواهد داد.

۳. **سستی فرایند آموزش**: حالت مقابل فروپاشی مد را تصور کنید. اگر در چندین گام اول آموزش مولد خروجی های مولد بسیار دور از فضای ورودی‌های واقعی باشد، از آنجایی که وظیفه‌ی ممیز نسبت به

(آ) نمونه‌های تولید شده روی مجموعه داده MNIST (ب) نمونه‌های تولید شده روی مجموعه داده CIFAR

شکل ۲-۲۲: نمونه‌های تولید شده توسط یک GAN معمولی [۵۱]

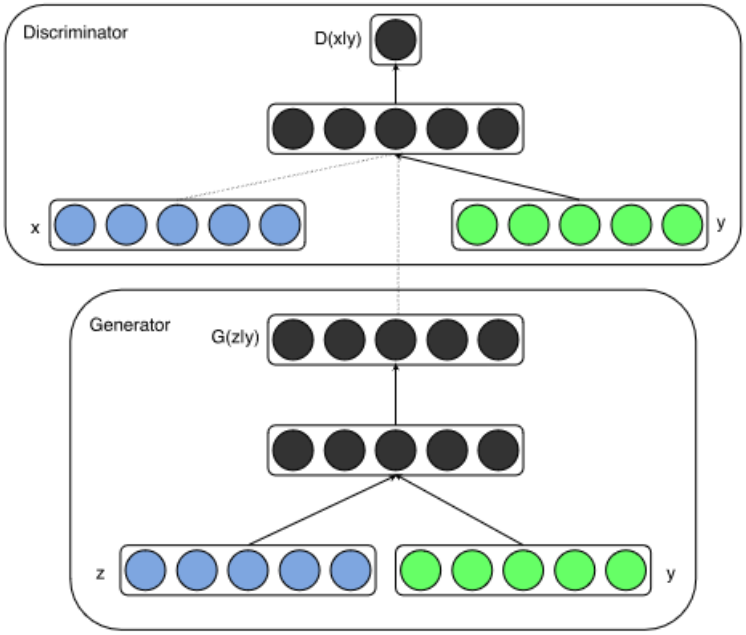
[1] Mode Collapse
[2] Nash Equilibrium

مولد به مراتب ساده‌تر است، ممیز به سرعت از مولد در بازی پیشی می‌گیرد و می تواند به راحتی نمونه‌های ساختگی و واقعی را از هم تفکیک کند. بنابراین از آنجایی که خروجی ممیز به ازای هر ورودی دلخواه عددی بسیار نزدیک به صفر (برای ورودی‌های ساختگی) و یا بسیار نزدیک به ۱ (برای ورودی های واقعی) است، مولد دچار پدیده‌ی گرادیان محو شونده[1] می شود و نمی تواند باز خورد مفیدی از ممیز برای تقویت خود دریافت کند.

در سالیان پس از گسترش استفاده از GANها راه حل‌های زیادی برای حل پاره‌ای از مسائل ذکر شده، ارائه شده است. یکی از اولین بهبودهایی که خصوصاً برای حل مشکل فروپاشی مد مطرح شد اما نشان داده شده است که می‌تواند روی بهبود همگرایی و پایداری فرایند آموزش نیز تاثیرگذار باشد [۵۴، ۵۵]، مشروط کردن خروجی مولد به برچسب مورد انتظار است. این ایده منجر به معرفی ساختارهای جدیدی به نام‌های cGAN و کمی بعدتر ACGAN شد که در ادامه کمی بیشتر این دو ساختار را مورد بررسی قرار خواهیم داد.

۲-۴-۱-۱ cGAN

مشروط کردن شبکه‌های مولد و ممیز یک GAN ایده‌ای بود که در خود مقاله اصلی معرفی GANها به عنوان راستایی برای پژوهش‌های آینده معرفی شده بود و ظرف مدت چند ماه، اولین تحقیق در این زمینه به ثمر رسید و در [۵۶] محققین معماری Conditional Generative Adversarial Network (cGAN) را معرفی کردند. ایده‌ی



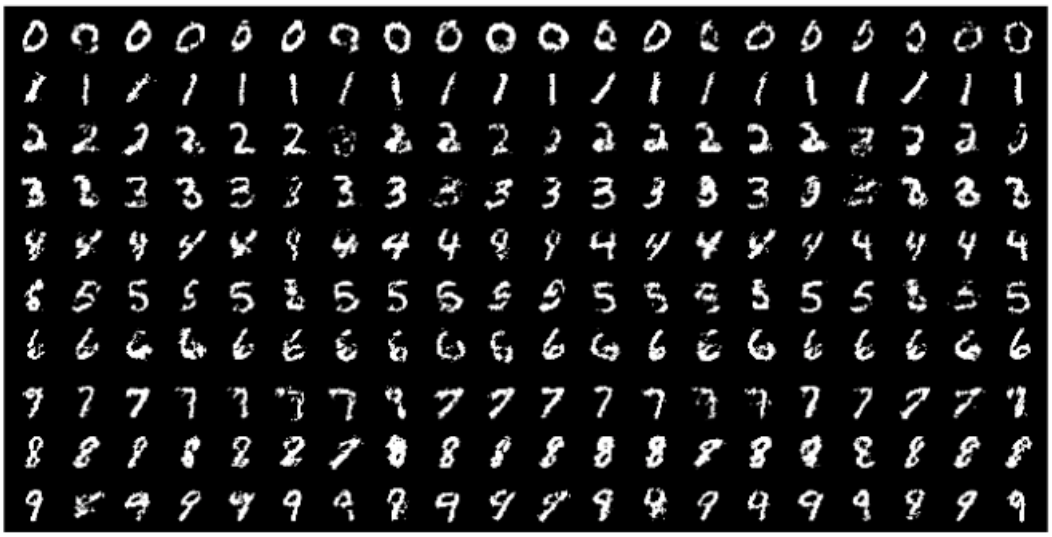
شکل ۲-۲۳: معماری cGAN [۵۶]

[1] Vanishing Gradient

تزریق یک شرط به ورودی‌های مولد و ممیز در این پژوهش به صورت بسیار ابتدایی و بدیهی مطرح شده است و به عنوان عامل مشروط کننده از برچسب کلاس نمونه‌ها استفاده می‌شود. مطابق شکل ۲-۲۳ برای تزریق این شرط در هر دو شبکه‌ی مولد و ممیز یک لایه‌ی embedding قابل یادگیری معرفی می شود که برچسب را به فضای چند بعدی دلخواه ببرد. پس از دریافت embedding متناظر با برچسب ورودی، این بردار به بردار ورودی چسبانده [۱] می‌شود و لایه‌های بعدی مانند یک GAN معمولی عمل خواهند کرد. در این وضعیت مولد G و ممیز D توزیع‌های شرطی $G(z|y)$ و $D(x|y)$ را تخمین می زنند و بازی ۲-۱۷ به شکل ساختار یافته تر زیر در خواهد آمد:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x|y)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log(1-D(G(z|y)))\right]$$

که پایداری بیشتری نسبت به بازی یک GAN معمولی دارد و منجر به تولید خروجی‌های با کیفیت تری می‌شود. علاوه بر آن، همانطور که در هر سطر از شکل ۲-۲۴ مشاهده می‌شود، مولد با استفاده از این روش قادر به یادگرفتن اطلاعاتی راجع‌به برچسب مورد انتظار است و می‌تواند خروجی‌هایی متناظر با برچسب ورودی تولید کند.



شکل ۲-۲۴: تاثیر تغییر برچسب ورودی مولد در نمونه‌های تولید شده توسط یک cGAN. هر سطر از شکل متناظر با یک برچسب ورودی در مولد است [۵۶].

[1] Concatenate

فصل سوم

پیشنهاد روشی نوین برای مقابله با حملات تخاصمی

فصل چهارم

شبیه سازی و نتایج ارزیابی

فصل پنجم

نتیجه‌گیری و پیشنهادها

واژه‌نامه انگلیسی به فارسی

A

حملات تخاصمی ..... Adversarial Attacks

اختلال تخاصمی ..... Adversarial Perturbation

پاکسازی تخاصمی ..... Adversarial Purification

نمونه تخاصمی ..... Adversarial Sample

آموزش تخاصمی ..... Adversarial Training

رویه‌ی حمله ..... Attack Surface

مهاجم ..... Attacker

تاجر خودکار ..... Auto Trader

خودرمزگذار ..... Autoencoder

دسته‌بند اضافی ..... Auxiliary Classifer

B

پس‌انتشار ..... Backpropagation

جستجوی دودویی ..... Binary Search

جعبه سیاه ..... Black Box

محدودیت های جعبه‌ای ..... Box Constraints

C

آبشاری ..... Cascade

مشروط بر کلاس ..... Class Conditional

دسته بند ها ..... Classifiers

اطمینان ..... Confidence

تقلیل اطمینان ..... Confidence Reduction

حملات سایبری ..... Cyber Attacks

D

یادگیری عمیق ..... Deep Learning

تقطیر دفاعی ..... Defensive Distillation

تشخیص ..... Detection

مدل انتشاری ..... Diffusion Model

ممیز ..... Discriminator

اعوجاج ..... $\text{Distortion}(x, \hat{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$ where $x, \hat{x} \in \mathbb{R}^n$

E

حمله گریزانه ..... Evasion Attack

حمله اکتشافی ..... Exploratory Attack

F

تطبیق ویژگی ..... Feature Matching

فضای ویژگی ..... Feature Space

انتشار پیش‌رو ..... Forward Diffusion

کاملاً متصل ..... Fully Connected

G

شبکه مولد تخاصمی ..... Generative Adversarial Network (GAN)

مدل های مولد ..... Generative Models

نزول گرادیان ..... Gradient Descent

H

لایه‌های پنهان ..... Hidden Layers

K

تخمین چگالی هسته‌ای ..... Kernel Density Estimation

L

نهفته ..... Latent

نرخ یادگیری ..... Learning Rate

ضرر ..... Loss

M

خمینه ..... Manifold

سنجه ..... Metric

کمینه‌ـبیشینه ..... Min-max

N

نویز ..... Noise

نرم ..... Norm

P

خط لوله ..... Pipeline

پیکسل ..... Pixel

حمله مسموم‌کننده ..... Poisoning Attack

پسین ..... Posterior

پیشین ..... Prior

نزول گرادیان افکنده ..... Projected Gradient Descent

پاکسازی ..... Purification

R

آغاز تصادفی ..... Random Initialization

خطای بازسازی ..... Reconstruction Error

بهساز ..... Reformer

منظم‌سازی ..... Regularization

انتشار معکوس ..... Reverse Diffusion

S

ماشین خودران ..... Self-driving Car

تحلیل احساسات ..... Sentiment Analysis

متغیر لَنگی ..... Slack Variable

T

دسته‌بندی اشتباه هدفمند ..... Targeted Misclassification

تنسور ..... Tensor

مدل‌های تهدید ..... Threat Models

تلورانس ..... Tolerance

U

دسته‌بندی اشتباه غیر هدفمند ..... Untargeted Misclassification

V

قربانی ..... Victim

W

جعبه سفید ..... White Box

واژه‌نامه فارسی به انگلیسی

ا

آبشاری ..... Cascade

آغاز تصادفی ..... Random Initialization

آموزش تخاصمی ..... Adversarial Training

اختلال تخاصمی ..... Adversarial Perturbation

اطمینان ..... Confidence

اعوجاج ..... Distortion$(x, \hat{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2$ where $x, \hat{x} \in \mathbb{R}^n$

انتشار پیش‌رو ..... Forward Diffusion

انتشار معکوس ..... Reverse Diffusion

ب

بهساز ..... Reformer

پ

پاکسازی ..... Purification

پاکسازی تخاصمی ..... Adversarial Purification

پس‌انتشار ..... Backpropagation

پسین ..... Posterior

پیشین ..... Prior

پیکسل ..... Pixel

ت

تاجر خودکار ..... Auto Trader

تحلیل احساسات ..... Sentiment Analysis

تخمین چگالی هسته‌ای ..... Kernel Density Estimation

تشخیص ..... Detection

تطبیق ویژگی ..... Feature Matching

تقطیر دفاعی ..... Defensive Distillation

تقلیل اطمینان ..... Confidence Reduction

تلورانس ..... Tolerance

تنسور ..... Tensor

ج

جستجوی دودویی ..... Binary Search

جعبه سفید ..... White Box

جعبه سیاه ..... Black Box

ح

حملات تخاصمی ..... Adversarial Attacks

حملات سایبری ..... Cyber Attacks

حمله اکتشافی ..... Exploratory Attack

حمله گریزانه ..... Evasion Attack

حمله مسموم‌کننده ..... Poisoning Attack

خ

خط لوله ..... Pipeline

خطای بازسازی ..... Reconstruction Error

خمینه ..... Manifold

خودرمزگذار ..... Autoencoder

د

دسته بند ها ..... Classifiers

دسته‌بند اضافی ..... Auxiliary Classifer

دسته‌بندی اشتباه غیر هدفمند ..... Untargeted Misclassification

دسته‌بندی اشتباه هدفمند ..... Targeted Misclassification

ر

رویه‌ی حمله ..... Attack Surface

س

سنجه ..... Metric

ش

شبکه مولد تخاصمی ..... Generative Adversarial Network (GAN)

ض

ضرر ..... Loss

ف

فضای ویژگی ..... Feature Space

ق

قربانی ..... Victim

ک

کاملاً متصل ..... Fully Connected

کمینه-بیشینه ..... Min-max

ل

لایه‌های پنهان ..... Hidden Layers

م

ماشین خودران ..... Self-driving Car

متغیر لَنگی ..... Slack Variable

محدودیت های جعبه‌ای ..... Box Constraints

مدل انتشاری ..... Diffusion Model

مدل های مولد ..... Generative Models

مدل‌های تهدید ..... Threat Models

مشروط بر کلاس ..... Class Conditional

ممیز ..... Discriminator

منظم‌سازی ..... Regularization

مهاجم ..... Attacker

ن

نرخ یادگیری ..... Learning Rate

نرم ..... Norm

نزول گرادیان ..... Gradient Descent

نزول گرادیان افکنده ..... Projected Gradient Descent

نمونه تخاصمی ..... Adversarial Sample

نویز ..... Noise

نهفته ..... Latent

ی

مراجع

[1] Wu, H., Yunas, S., Rowlands, S., Ruan, W., and Wahlstrom, J., "Adversarial Driving: Attacking End-to-End Autonomous Driving", in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, June 2023.

[2] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A., "Practical Black-Box Attacks against Machine Learning", Mar. 2017.

[3] Nehemya, E., Mathov, Y., Shabtai, A., and Elovici, Y., "Taking Over the Stock Market: Adversarial Perturbations Against Algorithmic Traders", Sept. 2021.

[4] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D., "Adversarial Attacks and Defences: A Survey", Sept. 2018.

[5] Costa, J. C., Roxo, T., Proença, H., and Inácio, P. R. M., "How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defenses", May 2023.

[6] Khamaiseh, S. Y., Bagagem, D., Al-Alaj, A., Mancino, M., and Alomari, H. W., "Adversarial Deep Learning: A Survey on Adversarial Attacks and Defense Mechanisms on Image Classification", *IEEE Access*, Vol. 10, pp. 102266–102291, 2022.

[7] Ren, K., Zheng, T., Qin, Z., and Liu, X., "Adversarial Attacks and Defenses in Deep Learning", *Engineering*, Vol. 6, pp. 346–360, Mar. 2020.

[8] Sun, L., Tan, M., and Zhou, Z., "A survey of practical adversarial example attacks", *Cybersecurity*, Vol. 1, p. 9, Dec. 2018.

[9] Goodfellow, I. J., Shlens, J., and Szegedy, C., "Explaining and Harnessing Adversarial Examples", Mar. 2015.

[10] Li, Y., Cheng, M., Hsieh, C.-J., and Lee, T. C. M., "A Review of Adversarial Attack and Defense for Classification Methods", *The American Statistician*, Vol. 76, pp. 329–345, Oct. 2022.

[11] Qiu, H., Custode, L. L., and Iacca, G., "Black-box adversarial attacks using Evolution Strategies", in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1827–1833, July 2021.

[12] Zhou, M., Gao, X., Wu, J., Liu, K., Sun, H., and Li, L., "Investigating White-Box Attacks for On-Device Models", Mar. 2024.

[13] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R., "Intriguing properties of neural networks", Feb. 2014.

[14] Carlini, N. and Wagner, D., "Towards Evaluating the Robustness of Neural Networks", Mar. 2017.

[15] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A., "Towards Deep Learning Models Resistant to Adversarial Attacks", Sept. 2019.

[16] Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization", Jan. 2017.

[17] Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T., "Adversarial Training for Free!", Nov. 2019.

[18] Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B., "You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle", Nov. 2019.

[19] Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M., "Geometry-aware Instance-reweighted Adversarial Training", May 2021.

[20] Wong, E., Rice, L., and Kolter, J. Z., "Fast is better than free: Revisiting adversarial training", Jan. 2020.

[21] Gu, S. and Rigazio, L., "Towards Deep Neural Network Architectures Robust to Adversarial Examples", Apr. 2015.

[22] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A., "Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks", Mar. 2016.

[23] Zi, B., Zhao, S., Ma, X., and Jiang, Y.-G., "Revisiting Adversarial Robustness Distillation: Robust Soft Labels Make Student Better", Aug. 2021.

[24] Cui, J., Liu, S., Wang, L., and Jia, J., "Learnable Boundary Guided Adversarial Training", Aug. 2021.

[25] Chen, E.-C. and Lee, C.-R., "LTD: Low Temperature Distillation for Robust Adversarial Training", June 2023.

[26] Ross, A. and Doshi-Velez, F., "Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, Apr. 2018.

[27] Gao, J., Wang, B., Lin, Z., Xu, W., and Qi, Y., "DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples", Apr. 2017.

[28] Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J., "Towards Robust Neural Networks via Random Self-ensemble", July 2018.

[29] Meng, D. and Chen, H., "MagNet: A Two-Pronged Defense against Adversarial Examples", in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, (Dallas Texas USA), pp. 135–147, ACM, Oct. 2017.

[30] Samangouei, P., Kabkab, M., and Chellappa, R., "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models", May 2018.

[31] Jin, G., Shen, S., Zhang, D., Dai, F., and Zhang, Y., "APE-GAN: Adversarial Perturbation Elimination with GAN", in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, United Kingdom), pp. 3842–3846, IEEE, May 2019.

[32] Li, Y., Min, M. R., Lee, T., Yu, W., Kruus, E., Wang, W., and Hsieh, C.-J., "Towards Robustness of Deep Neural Networks via Regularization", in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Montreal, QC, Canada), pp. 7476–7485, IEEE, Oct. 2021.

[33] Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A., "Diffusion Models for Adversarial Purification", May 2023.

[34] Gong, Z., Wang, W., and Ku, W.-S., "Adversarial and Clean Data Are Not Twins", Apr. 2017.

[35] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A., "The Limitations of Deep Learning in Adversarial Settings", Nov. 2015.

[36] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B., "On Detecting Adversarial Perturbations", Feb. 2017.

[37] Li, X. and Li, F., "Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics", Oct. 2017.

[38] Saberian, M. J. and Vasconcelos, N., "Boosting Classifier Cascades", *Advances in Neural Information Processing Systems*, Vol. 23, pp. 2047–2055, 2010.

[39] Zheng, Z. and Hong, P., "Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks", in *Advances in Neural Information Processing Systems* (Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., eds.), Vol. 31, Curran Associates, Inc., 2018.

[40] Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., and Jordan, M. I., "ML-LOO: Detecting Adversarial Examples with Feature Attribution", June 2019.

[41] Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., and Schmidt-Erfurth, U., "F-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks", *Medical Image Analysis*, Vol. 54, pp. 30–44, May 2019.

[42] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., "Improved Techniques for Training GANs", June 2016.

[43] Wang, H., Miller, D. J., and Kesidis, G., "Anomaly Detection of Adversarial Examples using Class-conditional Generative Adversarial Networks", May 2022.

[44] Odena, A., Olah, C., and Shlens, J., "Conditional Image Synthesis With Auxiliary Classifier GANs", July 2017.

[45] Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B., "Detecting Adversarial Samples from Artifacts", Nov. 2017.

[46] Miller, D. J., Wang, Y., and Kesidis, G., "When Not to Classify: Anomaly Detection of Attacks (ADA) on DNN Classifiers at Test Time", June 2018.

[47] Kullback, S. and Leibler, R. A., "On Information and Sufficiency", *The Annals of Mathematical Statistics*, Vol. 22, pp. 79–86, Mar. 1951.

[48] Lee, K., Lee, K., Lee, H., and Shin, J., "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks", Oct. 2018.

[49] Roth, K., Kilcher, Y., and Hofmann, T., "The Odds are Odd: A Statistical Test for Detecting Adversarial Examples", May 2019.

[50] Tian, J., Zhou, J., Li, Y., and Duan, J., "Detecting Adversarial Examples from Sensitivity Inconsistency of Spatial-Transform Domain", Mar. 2021.

[51] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative Adversarial Networks", June 2014.

[52] Saad, M. M., O'Reilly, R., and Rehmani, M. H., "A survey on training challenges in generative adversarial networks for biomedical image analysis", *Artificial Intelligence Review*, Vol. 57, p. 19, Jan. 2024.

[53] Saxena, D. and Cao, J., "Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions", *ACM Computing Surveys*, Vol. 54, pp. 1–42, Apr. 2022.

[54] Mohebbi Moghaddam, M., Boroomand, B., Jalali, M., Zareian, A., Daeijavad, A., Manshaei, M. H., and Krunz, M., "Games of GANs: Game-theoretical models for generative adversarial networks", *Artificial Intelligence Review*, Vol. 56, pp. 9771–9807, Sept. 2023.

[55] Mounjid, O. and Guo, X., "Convergence of GANs Training: A Game and Stochastic Control Methodology", Dec. 2021.

[56] Mirza, M. and Osindero, S., "Conditional Generative Adversarial Nets", Nov. 2014.

Athena: A Framework for Adversarial Robustness using Class-conditional Generative Models

Arian Tashakkor
tashakkor.a@ec.iut.ac.ir

Feb 19, 2024

Department of Electrical and Computer Engineering
Isfahan University of Technology, Isfahan 84156-83111, Iran
Degree: M.Sc. Language: Farsi

Supervisor: Prof. Mohammad Hossein Manshaei (manshaei@ece.iut.ac.ir)

Abstract

Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs.

Key Words: Deep Learning, Machine Learning, Adversarial Robustness, Adversarial Attacks

Isfahan University of Technology

Department of Electrical and Computer Engineering

Athena: A Framework for Adversarial Robustness using Class-conditional Generative Models

A Thesis

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

by

Arian Tashakkor

Evaluated and Approved by the Thesis Committee, on Jan 01, 2022

1. Mohammad Hossein Manshaei, Prof. (Supervisor)
2. XYZ, Prof. (Examiner)
3. XYZ, Assist. Prof (Examiner)

Dr. Behzad Nazari, Department Graduate Coordinator