



دانشکده مهندسی کامپیوتر

نمونه‌سازی تقابلی با استفاده از شبکه‌های مولد تقابلی

پروژه کارشناسی مهندسی کامپیوتر گرایش هوش مصنوعی

یگانه مرشدزاده

استاد راهنما

دکتر ناصر مزینی

تابستان ۱۴۰۰

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تأییدیه‌ی هیأت داوران جلسه‌ی دفاع از گزارش پروژه پایانی

نام دانشکده: دانشکده مهندسی کامپیوتر

نام دانشجو: یگانه مرشدزاده

عنوان گزارش پروژه پایانی: نمونه‌سازی تقابلی با استفاده از شبکه‌های مولد تقابلی

تاریخ دفاع: تابستان ۱۴۰۰

رشته: مهندسی کامپیوتر

گرایش: هوش مصنوعی

ردیف	سمت	نام و نام خانوادگی	مرتبه دانشگاهی	دانشگاه یا مؤسسه	امضا
۱	استاد راهنما	دکتر ناصر مزینی	دانشیار	دانشگاه علم و صنعت ایران	

چکیده

مدل‌های از قبل آموزش دیده (در مقیاس بزرگ)، مانند یادگیری عمیق، هم‌اکنون قلب و مرکز اصلی پیشرفت هوش مصنوعی است. با وجود این که شبکه‌های مصنوعی پیشرفت و موفقیت چشم‌گیری، در بیشتر وقت‌ها فرای توانایی انسان‌ها، از خود در حل مسئله‌های پیچیده نشان داده است، پژوهش‌های اکنون نشان داده‌اند که این شبکه‌ها نسبت به حمله‌های تقابلی در حالتی که تنها دستکاری‌های کوچکی اعمال شود، به طور کامل شبکه را فریب داده و در نتیجه، این شبکه‌ها بسیار آسیب‌پذیر و حساس هستند. در این گزارش پس از بیان اهمیت مسئله، یک دسته‌بندی از انواع حملات به شبکه‌های عصبی و همچنین مختصری از معمار و نحوه کار کردن شبکه مولد تقابلی بیان شده است. در ادامه به بررسی تاثیر نمونه تقابلی تولید شده توسط روش حمله Adv-GAN، که الهام گرفته از شبکه‌های مولد تقابلی است، در فریب شبکه هدف پرداخته شده است. در انتها نتایج بدست آمده در قالب انواع آمارها و نمودارها آورده شده است که همه نشان دهنده این هستند شبکه هدفی که قبل از حمله دقت $99/3\%$ را داشته است، پس از حمله بسیار موفق توسط بخش مولد شبکه Adv-GAN، به دقت $0/43\%$ رسیده است که نشان از میزان موفقیت $99/57\%$ حمله^۱ دارد.

واژگان کلیدی: فریب شبکه عصبی مصنوعی، روش نشانه‌ی گرادیان سریع، نمونه تقابلی، حمله جعبه سفید، حمله جعبه سیاه، یادگیری عمیق، شبکه مولد تقابلی

¹ Attack Success Rate

فهرست مطالب

ج	فهرست شکل‌ها
۱	فصل ۱: مقدمه
۱-۱	مقدمه
۲-۱	یک مثال از اهمیت کاربرد نمونه تقابلی
۴	فصل ۲: پیش‌زمینه
۴-۲	نمونه تقابلی و آموزش تقابلی
۵-۲	دسته‌بندی انواع حملات تقابلی
۵-۲-۲	حمله جعبه سفید
۵-۲-۲	حمله جعبه سیاه
۵-۲-۲	حمله جعبه نیمه‌سفید
۶-۲	شبکه مولد تقابلی
۶-۳-۲	مولد
۷-۳-۲	تمیزدهنده
۷-۳-۲	نحوه عملکرد شبکه
۹	فصل ۳: مروری بر کارهای مرتبط
۱۰-۳	الگوریتم علامت‌گذاری سریع
۱۰-۱-۳	معادله
۱۱-۱-۳	عملکرد الگوریتم

۱۱	۳-۱-۳ یک مثال از نمونه تقابلی
۱۲	فصل ۴: روش حل مسئله
۱۲	۱-۴ شرح مسئله
۱۳	۲-۴ شبکه Adv-GAN
۱۳	۱-۲-۴ معماری شبکه
۱۴	۲-۲-۴ معادله‌های شبکه
۱۶	فصل ۵: آزمایش‌ها و نتیجه‌ها
۱۶	۱-۵ مشخصات محیط اجرای کد
۱۶	۱-۱-۵ سخت افزار و سیستم عامل
۱۶	۲-۱-۵ نرم افزار و کتابخانه
۱۷	۲-۵ مجموعه داده
۱۷	۳-۵ بخش‌های کد و نحوه اجرایشان
۱۷	۱-۳-۵ آموزش شبکه هدف
۱۷	۲-۳-۵ آموزش شبکه Adv-GAN
۱۷	۳-۳-۵ ارزیابی نمونه‌های تقابلی تولید شده
۱۸	۴-۵ معماری و تنظیمات شبکه‌های عصبی
۱۸	۱-۴-۵ مدل هدف
۱۸	۲-۴-۵ مدل Adv-GAN
۱۹	۱-۲-۴-۵ شبکه مولد
۲۱	۲-۲-۴-۵ شبکه تمیزدهنده
۲۲	۵-۵ نتایج عملکرد شبکه هدف قبل از حمله
۲۴	۶-۵ آموزش شبکه Adv-GAN
۲۸	۷-۵ نتایج عملکرد شبکه هدف پس از حمله
۲۹	۸-۵ نمونه‌های تقابلی و مدل هدف

فصل ۶: نتیجه‌گیری و پیشنهادات ۳۳

۱-۶ نتیجه‌گیری ۳۳

۲-۶ پیشنهادات و کارهای آینده ۳۴

مراجع ۳۵

فهرست شکل‌ها

- ۱-۱ نمونه‌ای از خطا شبکه‌های عصبی در تشخیص نشانه‌های راهنمایی و رانندگی [۴] ۳
- ۱-۲ ساختار یک شبکه مولد تقابلی منبع ۷
- ۲-۲ تصویرسازی روند عملکرد یک شبکه مولد تقابلی ۸
- ۱-۳ نمونه‌ای از اعمال الگوریتم علامت گرادیان سریع [۷] ۱۱
- ۱-۴ نمای کلی شبکه Adv-GAN [۲۵] ۱۴
- ۱-۵ معماری شبکه هدف ۱۸
- ۲-۵ معماری بخش کدگذار شبکه مولد ۱۹
- ۳-۵ معماری یک بلوک Resnet در بخش گلوگاه شبکه مولد ۲۰
- ۴-۵ معماری بخش کدگشا شبکه مولد ۲۰
- ۵-۵ معماری شبکه تمیزدهنده ۲۱
- ۶-۵ نمودار دقت شبکه هدف قبل از حمله ۲۲
- ۷-۵ نمودار ضرر شبکه هدف قبل از حمله ۲۳
- ۸-۵ عملکرد شبکه هدف بر روی داده تست قبل از حمله ۲۳
- ۹-۵ نمودار ضرر مولد و تمیز دهنده شبکه Adv-GAN ۲۴
- ۱۰-۵ نمودار ضرر دستکاری شبکه Adv-GAN ۲۵
- ۱۱-۵ نمودار ضرر تقابلی شبکه Adv-GAN ۲۶
- ۱۲-۵ نمودار همه ضررهای شبکه Adv-GAN ۲۷
- ۱۳-۵ عملکرد شبکه هدف بر روی داده آموزش پس از حمله ۲۸

- ۱۴-۵ عملکرد شبکه هدف بر روی داده تست پس از حمله ۲۸
- ۱۵-۵ نمونه‌های تقابلی تولید شده از تصویرهای داده آموزش به همراه کلاس اصلی و کلاس
پیش‌بینی شده ۲۹
- ۱۶-۵ نمونه‌های تقابلی تولید شده از تصویرهای داده تست به همراه کلاس اصلی و کلاس
پیش‌بینی شده ۳۰
- ۱۷-۵ ماتریس درهم‌ریختگی برای نمونه‌های تقابلی تولید شده از تصویرهای داده آموزش . . . ۳۱
- ۱۸-۵ ماتریس درهم‌ریختگی برای نمونه‌های تقابلی تولید شده از تصویرهای داده تست ۳۲

فصل ۱

مقدمه

۱-۱ مقدمه

در حال حاضر، یادگیری عمیق^۱ به همراه سایر روش‌های یادگیری ماشین^۲ و هوش مصنوعی^۳ پیشرفت‌های چشمگیری در راه‌حل‌های سوال‌ها ارائه داده است به صورتی که برای کدگشایی و حل مسئله‌های سخت علمی با مقیاس بزرگ به طور مثال در بازسازی مدارهای مغزی^۴ [۱۱]، آنالیز و بررسی جهش‌های DNA^۵ [۲۶] از این روش‌ها استفاده می‌شود. علاوه بر این، شبکه‌های عصبی عمیق^۶ گزینه‌ی مورد علاقه پژوهشگران در هنگام حل بسیاری از مسئله‌های چالش برانگیز در تشخیص صدا و صحبت^۷ [۱۲]، متوجه شدن زبان طبیعی^۸ [۲۲] و بینایی ماشین^۹ هستند.

مواردی به مانند پیشرفت مداوم مدل‌های شبکه‌های عصبی عمیق [۲۳]، [۹]، دسترسی باز^{۱۰} [۲۴]، [۱۳]، [۱] به کتابخانه‌های نرم‌افزاری یادگیری عمیق و دسترسی آسان به سخت‌افزارهای مورد نیاز برای آموزش مدل‌های پیچیده کمک کرده‌اند تا یادگیری عمیق به سرعت به حدی از پختگی برای ورود به کاربردهای

¹Deep Learning

²Machine Learning

³Artificial Intelligence

⁴Reconstruction of Brain Circuits

⁵Analysis of Mutations in DNA

⁶Deep Neural Networks

⁷Speech Recognition

⁸Natural Language Understanding

⁹Computer Vision

¹⁰Open Access

حساس و مهم امنیت و ایمنی مانند ماشین‌های خودران^{۱۱}، سیستم‌های دیدبانی و مراقبت^{۱۲} [۱۹]، شناسایی بدافزارها^{۱۳} [۲۰]، [۸]، ربات‌ها و هواپیماهای بدون سرنشین^{۱۴} [۱۷]، [۵]، تشخیص دستوره‌های صوتی^{۱۵} [۱۲] و امنیت شناسایی چهره^{۱۶} بر روی گوشی همراه برسد.

این کاربردها در برخی موارد به حدی اهمیت دارند که جان و مال فرد به آن وابسته می‌شود [۱۶] و به طور مثال یک حمله‌کننده می‌تواند یک ماشین با راننده اتوماتیک^{۱۷} را گمراه کند یا کنترل عامل‌های هوشمندی که با صدا کنترل می‌شوند^{۱۸} را به دست بگیرد. پژوهش‌هایی صورت گرفته که نشان می‌دهد الگوریتم‌های یادگیری ماشین نسبت به نمونه‌های تقابلی^{۱۹} آسیب‌پذیر هستند. این نمونه‌ها، که با دستکاری‌های جزئی و نامحسوس در نمونه‌های مجموعه داده بدست می‌آیند، می‌توانند مدل دسته‌بندی‌کننده^{۲۱} را به اشتباه بیندازند.

[۲]

با توجه به اینکه در بیشتر پژوهش‌ها هدف افزایش دقت شبکه بوده و به امنیت و انعطاف‌پذیری مدل توجهی نشده است یا توجه کمی شده است. بنابراین حتی برای قدرتمندترین شبکه‌های آموزش دیده‌شده که دقت بسیار بالایی در دسته‌بندی تصاویر گزارش می‌دهند، می‌توان به نمونه‌های تقابلی‌ای دست یافت که شبکه را به سمت پیش‌بینی‌ای با اطمینان سوق دهد و گمراه کند.

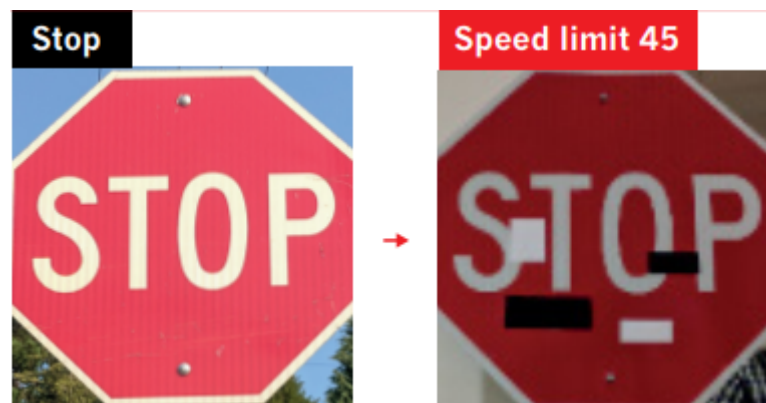
۲-۱ یک مثال از اهمیت کاربرد نمونه تقابلی

فرض کنید یک ماشین خودران در حال نزدیک شدن به یک تقاطع است و به یک تابلو ایست نزدیک می‌شود و به جای کاهش سرعت، سرعت خود را افزایش می‌دهد و در نتیجه تصادفی رخ می‌دهد. بعدها یک گزارشگر حادثه افشا می‌کند که به تابلو ایست چهار مستطیل چسبیده بودند و باعث شدند که هوش مصنوعی ماشین آن را به اشتباه محدودیت سرعت ۴۵ تشخیص دهد. این مثال واقعی توسط دانشمندان ارائه شده است و آن‌ها در

¹¹Self Driving Cars¹²Surveillance¹³Male-ware Detection¹⁴Robotics and Drones¹⁵Voice Command Recognition¹⁶Face ID Security¹⁷Autonomous Driving Vehicles¹⁸Voice Controlled Intelligent Agents¹⁹Adversarial Examples²⁰Perturbations²¹Classifiers

عمل توانستند با قرار دادن برجسب‌هایی بر روی تابلو ایست، سیستم هوش مصنوعی را به اشتباه و بدخوانی بیندازند [۴].

در شکل ۱-۱ تصویر سمت چپ با قرار دادن برجسب‌هایی به تصویر سمت راست تبدیل می‌شود. با وجود اینکه این تصاویر به راحتی توسط انسان قابل شناسایی هستند ولی توسط سیستم‌های هوش مصنوعی به جای تابلو ایست، تابلو محدودیت سرعت ۴۵ تشخیص داده می‌شود که این بسیار خطرناک است. [۱۰]



شکل ۱-۱: نمونه‌ای از خطا شبکه‌های عصبی در تشخیص نشانه‌های راهنمایی و رانندگی [۴]

فصل ۲

پیش زمینه

۱-۲ نمونه تقابلی و آموزش تقابلی

نمونه تقابلی^۱ یک نسخه دستکاری شده است از یک نمونه، برای مثال تصویر است که عمداً و آگاهانه آشفته و دستکاری شده است تا شبکه را به اشتباه بیندازد. این نمونه‌ها با تغییرهای جزئی تقابلی^۲ در تصویر اصلی به وجود می‌آیند و در حالی که برای انسان‌ها قابل تشخیص نیستند و مشابه^۳ به نظر می‌رسند، برای شبکه‌ها به شکلی کاملاً متفاوت به نظر می‌رسد. آموزش تقابلی^۴ نیز به معنی استفاده از نمونه‌های تقابلی در کنار نمونه‌های تمیز و دست نخورده برای آموزش مدل‌های یادگیری عمیق است.

¹Adversarial Example/Image

²Adversarial Perturbation

³Quasi-imperceptible/Imperceptible

⁴Adversarial Training

۲-۲ دسته بندی انواع حملات تقابلی

حمله های تقابلی^۵ در دسته بندی های مختلفی مانند حمله هدفمند^۶ و بدون هدف^۷، حمله جامع^۸ و وابسته به داده^۹، حمله دستکاری^{۱۰} و جایگذاری^{۱۱}، حمله جعبه سیاه^{۱۲}، حمله جعبه نیمه سفید^{۱۳} و جعبه سفید^{۱۴} طبقه بندی می شوند.

۱-۲-۲ حمله جعبه سفید

در این نوع حمله ها این طور در نظر گرفته می شود که دانش و اطلاع کامل بر مدل هدف، مقدارهای پارامترهای مدل، معماری، روش آموزش و در برخی موارد داده های آموزش در اختیار است.

۲-۲-۲ حمله جعبه سیاه

در حمله های جعبه سیاه به یک مدل هدف در مرحله آزمایش، نمونه های تقابلی ای که بدون دانش از مدل ساخته شده اند، داده می شوند. در برخی موارد این طور فرض می شود که حمله کننده دانش کم و محدودی از مدل به طور مثال روند آموزش و یا معماری آن را دارد ولی قطعاً درباره پارامترهای مدل چیزی نمی داند.

۳-۲-۲ حمله جعبه نیمه سفید

در این نوع حمله که بسیار شبیه به حمله های جعبه سیاه و همچنین جعبه سفید است، با این فرق که هر دانش و اطلاعات دیگری درباره مدل هدف در حمله استفاده می شود. به طور مثال می تواند احتمال هایی که از پیش بینی شبکه بدست می آید را داشته باشد در حالی که در حمله جعبه سیاه ممکن از تنها پیش بینی نهایی یا به عبارت دیگر کلاس پیش بینی شده را بداند.

⁵ Adversarial Attacks

⁶ Targeted Attack

⁷ Untargeted Attack

⁸ Universal Attack

⁹ Data Dependent Attack

¹⁰ Perturbation

¹¹ Replacement

¹² Black-Box Attack

¹³ Semi-White-Box Attack

¹⁴ White-Box Attack

لازم به ذکر است که حمله‌های جعبه سفید بسیار قوی‌تر هستند و بنابراین اگر مدلی بتواند در مقابل این نوع حمله‌ها مقاوم باشد، به طور مشابه می‌تواند در برابر حمله‌های جعبه سیاه و جعبه نیمه سفید مقاوم باشد.

۲-۳ شبکه مولد تقابلی

شبکه مولد تقابلی یا شبکه زایای دشمن گونه ^{۱۵} [۶]، که به اختصار GAN نیز گفته می‌شود، دارای دو بخش اصلی است به نام‌های مولد ^{۱۶} و تمیزدهنده ^{۱۷} است. این دو شبکه عصبی عملکردی برضد و برخلاف یکدیگر دارند. این دو شبکه در یک بازی مجموع-صفر ^{۱۸} با هم به رقابت می‌پردازند.

این شبکه در اصل به عنوان یک مدل مولد ^{۱۹} برای یادگیری بی نظارت ^{۲۰} ارائه شده بود ولی شبکه‌های مولد تقابلی برای یادگیری نیمه نظارتی ^{۲۱}، یادگیری نظارت شده ^{۲۲} و یادگیری تقویتی ^{۲۳} نیز کاربردی اثبات شده‌اند.

۲-۳-۱ مولد

مولد یا تولید کننده، یک نویزی که اغلب به صورت گوسی ^{۲۴} یا یکنواخت ^{۲۵} را به عنوان ورودی خود می‌گیرد و سپس تصویر بسیار نویزداری از دیتای ورودی را به وجود می‌آورد. هدف اصلی مولد این است که تصویرهای تولید شده تا جای ممکن به تصویرهای حقیقی و طبیعی مجموعه‌ی داده شباهت داشته باشد.

مولد از نویزهای تصادفی، تصویرهایی با همان ابعاد تصویرهای مجموعه داده به وجود می‌آورد.

¹⁵Generative Adversarial Network

¹⁶Generator

¹⁷Discriminator

¹⁸Zero-Sum Game

¹⁹Generative Model

²⁰Unsupervised Learning

²¹Semi-Supervised Learning

²²Supervised Learning

²³Reinforcement Learning

²⁴Gaussian

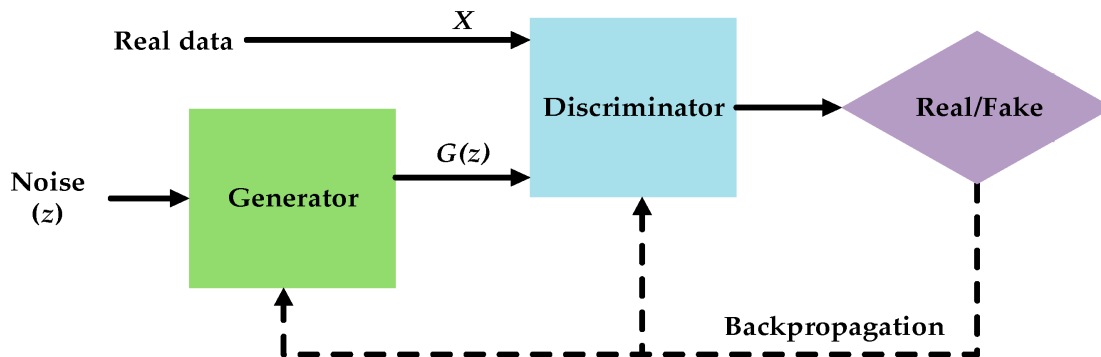
²⁵Uniform

۲-۳-۲ تمیزدهنده

تمیز دهنده، جداساز، تفکیک کننده یا تشخیص دهنده یک تصویر را به عنوان ورودی می‌گیرد و وظیفه آن تشخیص تصویرهای حقیقی (تصویرهای مجموعه داده) از تصاویر جعلی توسط مولد است. تمیزدهنده اگر تصویر را طبیعی تشخیص دهد در خروجی مقداری نزدیک به ۰ و اگر تصویر را غیرطبیعی تشخیص دهد مقداری نزدیک به ۱ را می‌دهد.

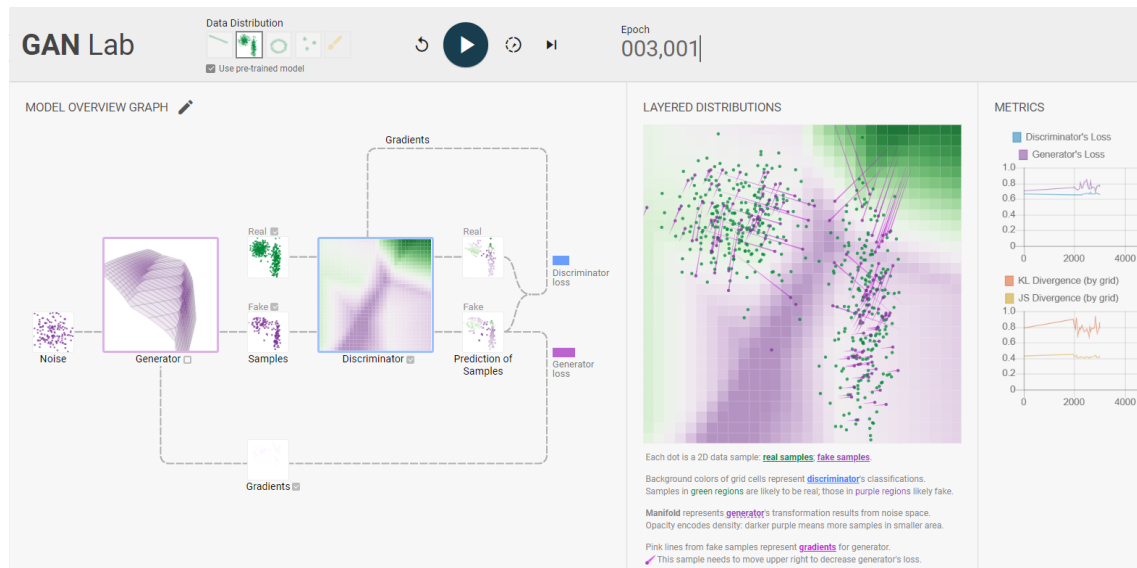
۲-۳-۳ نحوه عملکرد شبکه

در شبکه مولد تقابلی، هدف این است که مولد بتواند تصویری را به وجود بیاورد که به اندازه‌ی تصویرهای حقیقی، طبیعی جلوه کنند. به طوری که این تصویرها هم کامپیوتر و هم انسان را بتوانند فریب دهند. همان طور که در شکل ۱-۲ دیده می‌شود، تصویری‌های تولید شده توسط مولد و تصویرهای مجموعه داده به تمیزدهنده‌ی داده می‌شوند. مولد نیز آموزش می‌بیند که تصویرهایی طبیعی‌تر و فریبنده‌تر تولید کند و سپس پس از محاسبه گرادیان‌ها، از این مقادیر برای به روزرسانی پارامترهای هر شبکه استفاده می‌شود.



شکل ۱-۲: ساختار یک شبکه مولد تقابلی منبع

در شکل ۲-۲ نیز تصویرسازی‌ای از روند تعامل و آموزش شبکه مولد تقابلی، که در این لینک قابل دسترسی است، آورده شده است.



شکل ۲-۲: تصویرسازی روند عملکرد یک شبکه مولد تقابلی

فصل ۳

مروری بر کارهای مرتبط

برای حمله به شبکه‌های عصبی، می‌توان شبکه‌های عصبی عمیق در معرض نمونه‌های تقابلی قرار داد و با آموزش تقابلی، مقاومت^۱ و نیرومندی شبکه را افزایش داد. از روش‌های ارائه شده برای حمله می‌توان به روش‌های زیر اشاره کرد:

- علامت‌ی گرادیان سریع^۲ [۷]، [۱۴]
- روش کاهش گرادیان تصویر شده^۳ [۱۵]
- حمله کارلینی و وگنر^۴ [۳]
- حمله تک پیکسل^۵ [۲۱]
- روش تکرار شونده پایه^۶ [۱۴]
- روش فریب عمیق^۷ [۱۸]

در ادامه به توضیح الگوریتم علامت گرادیان سریع پرداخته می‌شود.

¹Robustness

²Fast Gradient Sign Method (FGSM)

³Projected Gradient Descent (PGD)

⁴Carlini and Wagner Attack (C&W)

⁵One pixel attack

⁶Basic Iterative Method (BIM)

⁷DeepFool

۳-۱ الگوریتم علامت گرادیان سریع

یکی از روش‌هایی که برای آموزش تقابلی موثر ارائه شده است، روش علامت گرادیان سریع است [۷] که به صورت بهینه‌ای یک دستکاری تقابلی برای یک تصویر را محاسبه می‌کند. این روش جزو حمله‌های جعبه سفید است زیرا حمله‌کننده^۸ نیاز دارد که به معماری و پارامترهای مدل در تمام زمان دسترسی داشته باشد.

۳-۱-۱ معادله

معادله الگوریتم علامت گرادیان سریع به صورت زیر است:

$$X^{adv} = X + \epsilon \cdot \text{sign}(\nabla_X J(X, Y_{true})) \quad (3-1)$$

در معادله ۳-۱ داریم:

- X ، تصویر اصلی و دست‌نخورده
- X^{adv} ، نمونه تقابلی که پس از اعمال دستکاری و اجرای روش نشانه‌ی گرادیان سریع بر روی تصویر اصلی بدست می‌آید
- ϵ ، یک ثابت برای تعیین شدت حمله (بزرگی آشتفگی تقابلی)
- J ، تابع زیان
- Y_{true} ، دسته و برچسب صحیح تصویر اصلی
- ∇_X ، شیب تابع هزینه

در این معادله (۳-۱) ، هر چه مقدار ϵ بیشتر باشد، نمونه تقابلی نسبت به تصویر اصلی غیر قابل تشخیص‌تر

می‌شود.

⁸Adversary




۳-۱-۲ عملکرد الگوریتم

در این روش هدف این است که یک نویز محاسبه شده که تصادفی نیست و در راستای شیب تابع هزینه، در تصویر اصلی وجود دارد، را به تصویر اولیه اضافه کند. در این روش حمله کننده دقیقاً از روشی که برای آموزش شبکه در تشخیص مرزهای دسته بندی استفاده می شود، بهره می برد به این طریق که تصویر دستکاری شده را طوری تنظیم می کند که تابع زیان به سمتی هدایت شود که با تصویر دیگری اشتباه گرفته شود.

روش الگوریتم علامت گرادیان سریع برای مدل های خطی بهترین حمله طبق قاعده L^∞ است و از آنجایی که شبکه های عصبی مدل هایی غیر خطی هستند، به این نتیجه می توان رسید که در مقابل شبکه های عصبی چندان خوب نمی تواند عمل کند.

۳-۱-۳ یک مثال از نمونه تقابلی

همان طور که در شکل ۳-۱ مشاهده می شود، یک نمونه از تصویرهای مجموعه داده ی ImageNet که بر روی شبکه GoogleNet آموزش دیده است، آورده شده است که شبکه با اطمینان ۵۷.۷٪ تصویر را پاندا تشخیص می دهد. سپس با اضافه کردن بردار بسیار کوچکی به تصویر می توان طبقه بندی شبکه را به خطا انداخت. مشاهده می شود که مدل، تصویر دستکاری شده را یک میمون دراز دست با اطمینان ۹۹.۳٪ تشخیص داده است.

	$+ .007 \times$		$=$	
x		$\text{sign}(\nabla_x J(\theta, x, y))$		$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$
“panda”		“nematode”		“gibbon”
57.7% confidence		8.2% confidence		99.3 % confidence

شکل ۳-۱: نمونه ای از اعمال الگوریتم علامت گرادیان سریع [۷]

فصل ۴

روش حل مسئله

برای این که نمونه‌های تقابلی ساخته شده هم شبکه را بیشتر به خطا بیندازند و هم از نظر انسان‌ها واقعی‌تر باشند، از شبکه‌ای به عنوان Adv-GAN استفاده می‌شود. [۲۵]

۴-۱ شرح مسئله

اگر موارد زیر را در نظر بگیریم:

• فضای ویژگی^۱: $X \subseteq R^n$

— تعداد ویژگی: n

• نمونه i ام در داخل مجموعه داده‌های آموزش که تشکیل شده است از بردارهای ویژگی^۲: (x_i, y_i)

— $x_i \in X$: با توجه به یک توزیع ناشناخته^۳ $x_i \sim P_{data}$ ساخته می‌شوند.

— $y_i \in Y$: برچسب کلاس حقیقی و درست

هدف سیستم یادگیرنده^۴ این است که یک طبقه‌بند^۵ $f: X \rightarrow Y$ را از دامنه X به مجموعه‌ی خروجی‌های

¹Feature Space

²Feature Vectors

³Unknown Distribution

⁴Learning System

⁵Classifier

دسته‌بندی Y یادبگیرد.

با دادن یک نمونه x هدف حمله‌کننده این است که یک نمونه متقابلی x_A را تولید کند که به صورت $f(x_A) \neq y$ (در حمله بدون هدف)، و $f(x_A) = t$ (در حمله هدفمند) که t کلاس هدف است، طبقه‌بندی شود. همچنین x_A باید از نظر فاصله‌ی L^2 یا سایر روش‌های سنجش فاصله به نمونه اصلی x نزدیک باشد.

۴-۲ شبکه Adv-GAN

این شبکه به این صورت است که از یک شبکه عصبی پیش‌خور^۶ برای تولید تغییرات جزئی و از یک شبکه تمیزدهنده برای اطمینان پیدا کردن از نزدیک به واقعیت بودن نمونه‌های تولید شده استفاده می‌شود. در مقایسه با روش علامت‌گرایان سریع، در این روش، پس از آموزش شبکه پیش‌خور، بی‌درنگ و فوراً می‌توان از آن برای تولید دستکاری‌های تقابلی برای هر نمونه‌ی ورودی، بدون نیاز به دسترسی به خود مدل (حمله جعبه نیمه‌سفید)، استفاده کرد. شبکه Adv-GAN این قابلیت را دارد که هم در حمله‌های جعبه نیمه‌سفید و جعبه سیاه و هم در حمله هدفمند و بدون هدف استفاده شود.

۴-۲-۱ معماری شبکه

نمای کلی معماری شبکه Adv-GAN در شکل ۴-۱ به تصویر کشیده شده است. این شبکه دارای سه شبکه عصبی است:

۱. مولد G

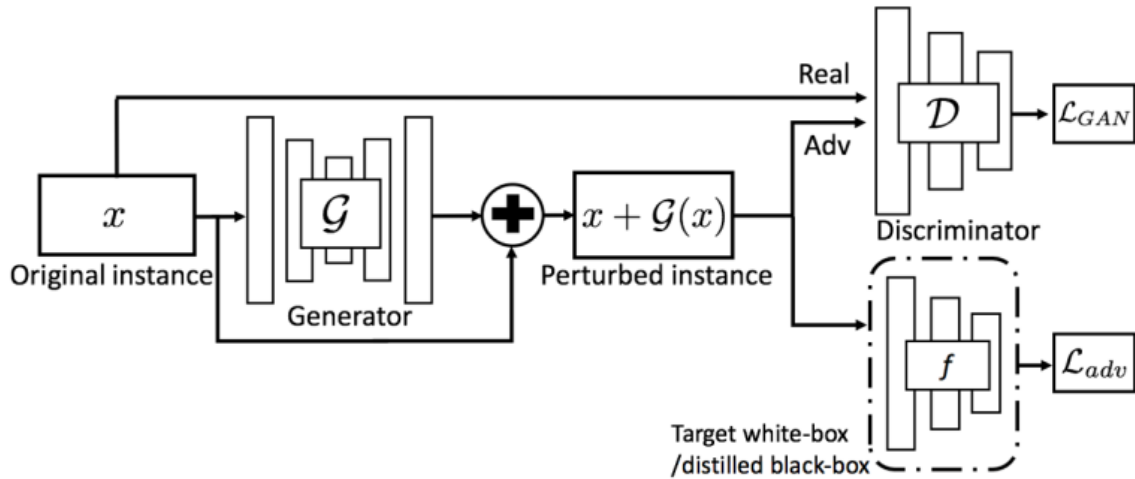
۲. تمیزدهنده D

۳. شبکه عصبی هدف f

مولد نمونه اصلی x را به عنوان ورودی می‌گیرد و یک دستکاری $G(x)$ را تولید می‌کند. سپس $x + G(x)$ به تمیزدهنده D فرستاده می‌شوند که برای تمایز دادن بین نمونه تولید شده و نمونه اصلی استفاده می‌شود. در

^۶Feed-Forward Neural Network

این جا نقش D به عنوان مشوق این است که نمونه‌های تولید شده از داده‌های کلاس اصلی غیر قابل تشخیص باشند. در ادامه $x + G(x)$ به عنوان ورودی به f داده می‌شود و ضرر L_{adv} خود را مطابق فرمول ۴-۲ به عنوان خروجی می‌دهد. [۲۵]



شکل ۴-۱: نمای کلی شبکه Adv-GAN [۲۵]

۴-۲-۲ معادله‌های شبکه

در فرمول ۴-۱ ضرر تقابلی^۷ آورده شده است:

$$L_{GAN} = \mathbb{E}_x \log D(x) + \mathbb{E}_x \log(1 - D(x + G(x))) \quad (۴-۱)$$

فرمول ۴-۲ برای محاسبه ضرر برای فریب دادن مدل هدف f در یک حمله هدفمند است:

$$L_{adv}^f = \mathbb{E}_x l_f(x + G(x), t) \quad (۴-۲)$$

⁷Adversarial Loss

لازم به ذکر است که ضرر L_{adv}^f تصویر دستکاری شده را تشویق می‌کند که به اشتباه در کلاس هدف t دسته‌بندی شود.

برای محدود کردن اندازه دستکاری‌ها از یک ضرر Soft Hinge بر روی قاعده L_2 در فرمول ۴-۳ استفاده می‌شود که در آن c نشان دهنده‌ی محدودیتی است که توسط کاربر تعیین می‌شود^۸:

$$L_{hinge} = \mathbb{E}_x \max(0, \|G(x)\|_2 - c) \quad (۴-۳)$$

بنابراین در مجموع هدف نهایی در فرمول ۴-۴ آورده شده است:

$$L = L_{adv}^f + \alpha L_{GAN} + \beta L_{hinge} \quad (۴-۴)$$

پارامترهای α و β متناسب با اهمیت هر یک از اهداف هستند.

در نهایت D و G با حل شدن یک بازی کمین بیش^۹ که در فرمول ۴-۵ آورده شده است، بدست می‌آیند.

$$\arg \min_G \max_D L \quad (۵-۴)$$

^۸User-Specified Bound

^۹MinMax Game

فصل ۵

آزمایش‌ها و نتیجه‌ها

تمامی کدهای پروژه در [این لینک](#) تحت لیسانس MIT قابل دسترس است. لازم به ذکر است که نمودارها در پوشه **results** و مدل‌های آموزش دیده در پوشه **models** قرار داده شده‌اند.

۵-۱ مشخصات محیط اجرای کد

۵-۱-۱ سخت افزار و سیستم عامل

GPU.1080Ti.xlarge با رم 31.3 GB که دارای ۶ vCPU (virtual CPU) است و سیستم عامل Ubuntu 18.04 استفاده شده است.

۵-۱-۲ نرم افزار و کتابخانه

کدها به زبان Python و با استفاده از چارچوب ^۱PyTorch نوشته شده‌اند. برای اجرا کدها از سخت افزار با دستور `nvidia-smi` میزان استفاده از GPU را مشاهده نمود. تمامی کتابخانه‌ها و پکیج‌های استفاده شده به همراه ورژن آن‌ها در فایل `requirements.txt` آورده شده است. برای نصب پکیج‌های مورد نیاز برای اجرا کد، می‌توان از دستور زیر استفاده کرد:

```
pip install -r requirements.txt
```

^۱Framework

۵-۲ مجموعه داده

در این پروژه از مجموعه داده MNIST استفاده می‌کنیم که این مجموعه داده مشتمل بر ۶۰۰۰۰ عکس ۲۸*۲۸ پیکسلی برای آموزش شبکه و ۱۰۰۰۰ عکس ۲۸*۲۸ پیکسلی برای تست شبکه است.

۵-۳ بخش‌های کد و نحوه اجرایشان

کدهای پروژه در سه بخش زیر قابل اجرا هستند:

۵-۳-۱ آموزش شبکه هدف

در این بخش، پس از بارگیری مجموعه داده‌های تست و آموزش، مدل هدف آموزش می‌بیند و سپس مدل آموزش دیده ذخیره می‌شود و نمودارهای مربوط به عملکرد شبکه در طول آموزش نیز رسم می‌شود. در آخرین بخش عملکرد شبکه بر روی مدل آموزش دیده نهایی اعلام می‌گردد.

```
python3 train_target_model.py
```

۵-۳-۲ آموزش شبکه Adv-GAN

در این بخش، پس از بارگیری مجموعه داده‌های تست و آموزش و مدل هدف از پیش آموزش دیده شده، مدل Adv-GAN آموزش می‌بیند و در همین حین ضررهای هر بخش از مدل نیز اعلام می‌شود. سپس مدل آموزش دیده ذخیره می‌شود و نمودارهای مربوط به عملکرد شبکه در طول آموزش نیز رسم می‌شود.

```
python3 train_advGAN_model.py
```

۵-۳-۳ ارزیابی نمونه‌های تقابلی تولید شده

در این بخش، مجموعه داده‌های تست و آموزش، ورژن نهایی مدل هدف و قسمت مولد Adv-GAN آموزش دیده شده بارگیری می‌شوند و سپس با تولید نمونه‌های تقابلی و تست کردن آن‌ها بر روی مدل هدف، عملکرد

شبکه با رسم ماتریس درهم‌ریختگی^۲ و اعلام دقت^۳ و امتیازاف-۱^۴ نشان داده می‌شود.

```
python3 test_adversarial_examples.py
```

۴-۵ معماری و تنظیمات شبکه‌های عصبی

۱-۴-۵ مدل هدف

در شکل ۱-۵ معماری شبکه هدف به عنوان دسته‌بند مجموعه داده MNIST آورده شده است. همچنین تنظیمات پارامترهای آموزش شبکه به صورت زیر است:

● Epoch: ۴۰

● Batch Size: ۲۵۶

```
MNIST_target_net(
    (conv1): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
    (conv2): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1))
    (conv3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
    (conv4): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1))
    (fc1): Linear(in_features=1024, out_features=200, bias=True)
    (fc2): Linear(in_features=200, out_features=200, bias=True)
    (logits): Linear(in_features=200, out_features=10, bias=True)
)
```

شکل ۱-۵: معماری شبکه هدف

۲-۴-۵ مدل Adv-GAN

تنظیمات پارامترهای آموزش شبکه به صورت زیر است:

^۲Confusion Matrix

^۳Accuracy

^۴F1-Score

● Epoch: ۶۰

● Batch Size: ۱۲۸

شبکه مولد ۱-۲-۴-۵

این شبکه دارای سه بخش است که به ترتیب در ادامه‌ی هم قرار دارند:

۱. کدگذار^۵

```
Generator(
  (encoder): Sequential(
    (0): Conv2d(1, 8, kernel_size=(3, 3), stride=(1, 1))
    (1): InstanceNorm2d(8, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (2): ReLU()
    (3): Conv2d(8, 16, kernel_size=(3, 3), stride=(2, 2))
    (4): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (5): ReLU()
    (6): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2))
    (7): InstanceNorm2d(32, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
    (8): ReLU()
  )
)
```

شکل ۵-۲: معماری بخش کدگذار شبکه مولد

۲. گلوگاه^۶: این بخش متشکل از ۴ بلوک Resnet یکسان می‌باشد که معماری اولین بلوک آن در شکل

۵-۳ آورده شده است.

^۵Encoder

^۶Bottleneck

```
(bottle_neck): Sequential(
  (0): ResnetBlock(
    (conv_block): Sequential(
      (0): ReflectionPad2d((1, 1, 1, 1))
      (1): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), bias=False)
      (2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (3): ReLU(inplace=True)
      (4): ReflectionPad2d((1, 1, 1, 1))
      (5): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), bias=False)
      (6): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
)
```

شکل ۵-۳: معماری یک بلوک Resnet در بخش گلوگاه شبکه مولد

۳. کدگشا^۷

```
(decoder): Sequential(
  (0): ConvTranspose2d(32, 16, kernel_size=(3, 3), stride=(2, 2), bias=False)
  (1): InstanceNorm2d(16, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  (2): ReLU()
  (3): ConvTranspose2d(16, 8, kernel_size=(3, 3), stride=(2, 2), bias=False)
  (4): InstanceNorm2d(8, eps=1e-05, momentum=0.1, affine=False, track_running_stats=False)
  (5): ReLU()
  (6): ConvTranspose2d(8, 1, kernel_size=(6, 6), stride=(1, 1), bias=False)
  (7): Tanh()
)
```

شکل ۵-۴: معماری بخش کدگشا شبکه مولد

⁷Decoder

شبکه تمیزدهنده ۲-۲-۴-۵

```

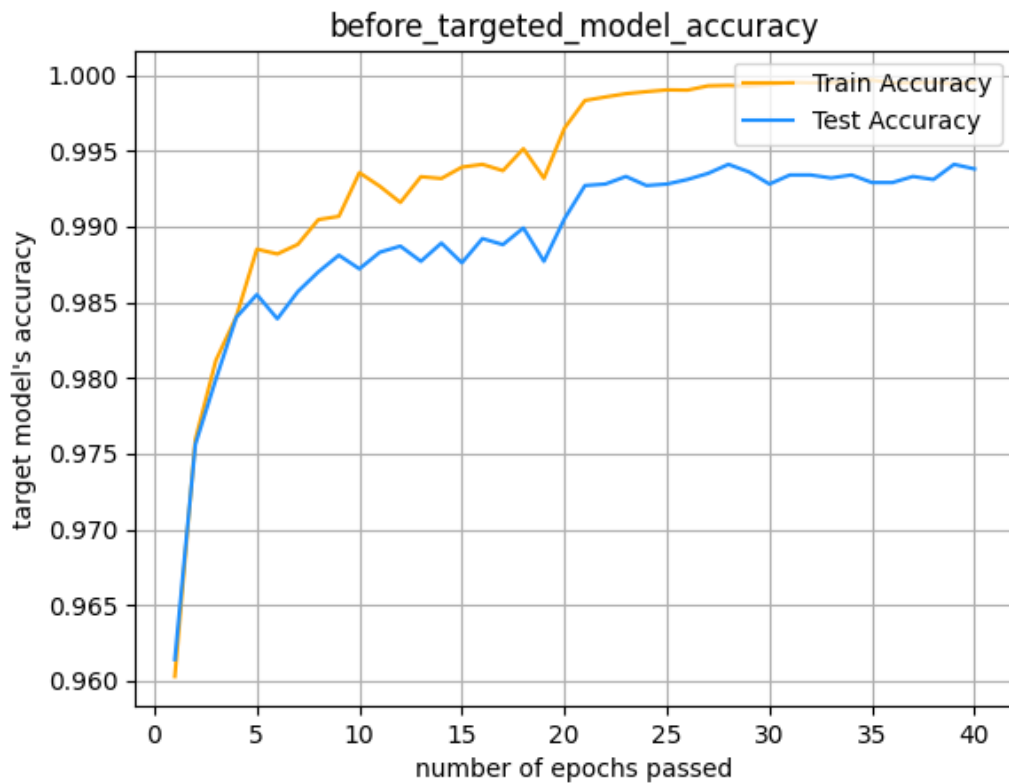
Discriminator(
  (model): Sequential(
    (0): Conv2d(1, 8, kernel_size=(4, 4), stride=(2, 2))
    (1): LeakyReLU(negative_slope=0.2)
    (2): Conv2d(8, 16, kernel_size=(4, 4), stride=(2, 2))
    (3): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (4): LeakyReLU(negative_slope=0.2)
    (5): Conv2d(16, 32, kernel_size=(4, 4), stride=(2, 2))
    (6): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (7): LeakyReLU(negative_slope=0.2)
    (8): Conv2d(32, 1, kernel_size=(1, 1), stride=(1, 1))
    (9): Sigmoid()
  )
)

```

شکل ۵-۵: معماری شبکه تمیزدهنده

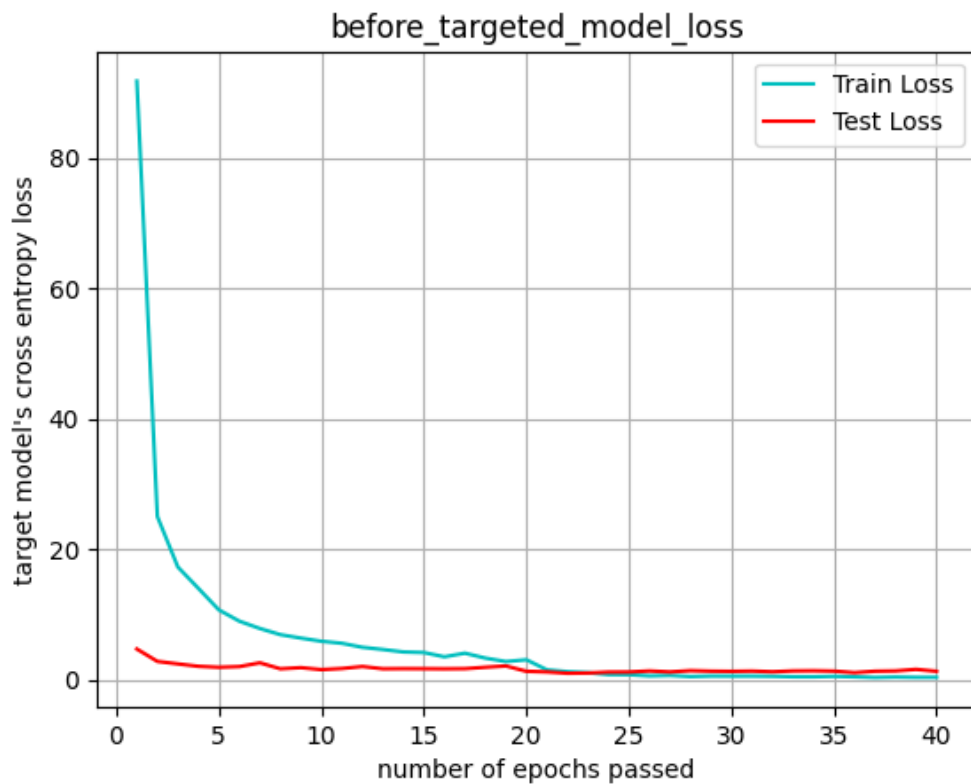
۵-۵ نتایج عملکرد شبکه هدف قبل از حمله

در شکل ۵-۶ نمودار عملکرد دقت شبکه در طول آموزش شبکه هدف بر روی داده تست و آموزش آورده شده است و همان طور که مشاهده می‌شود دقت نمونه با افزایش تعداد دورهای آموزش بهتر می‌شود.



شکل ۵-۶: نمودار دقت شبکه هدف قبل از حمله

به همین منوال در شکل ۵-۷ مشاهده می‌شود که ضرر شبکه بر روی هر دو نوع مجموعه داده نزولی است.



شکل ۵-۷: نمودار ضرر شبکه هدف قبل از حمله

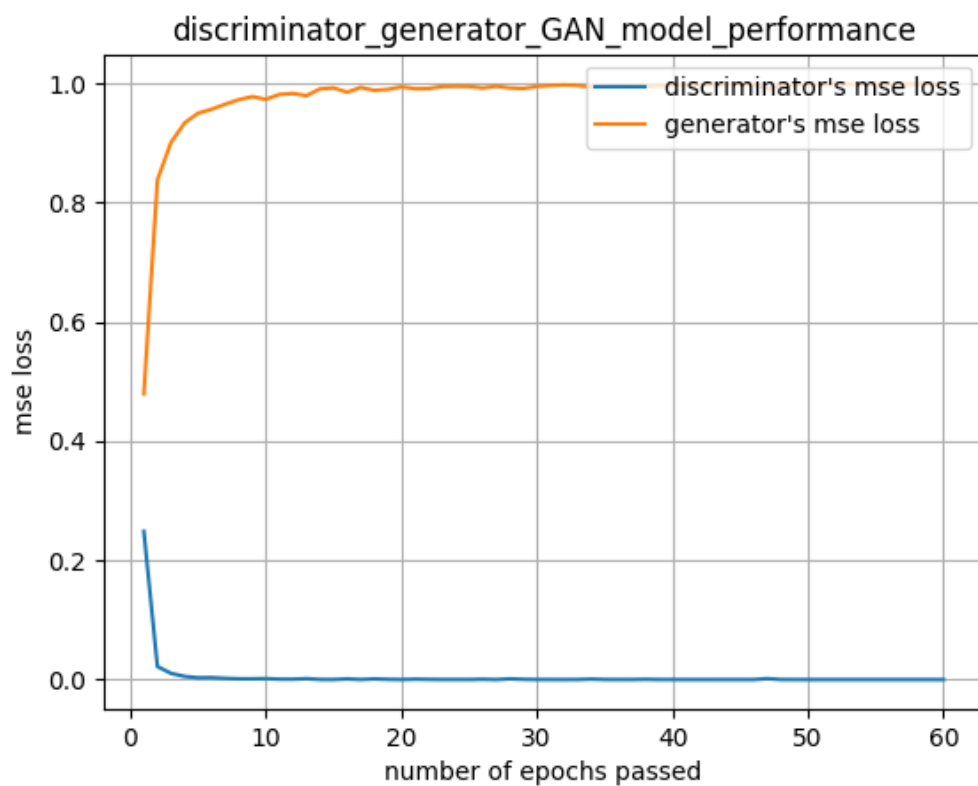
در نهایت، پس از پایان آموزش شبکه هدف، نتایج عملکرد شبکه بر روی مجموعه داده تست مطابق شکل ۵-۸ است و می‌توان گفت که شبکه از ۱۰۰۰۰ نمونه تست تنها ۷۰ مورد را به درستی دسته‌بندی نکرده است و ۹۹۳۰ مورد را کاملاً درست تشخیص داده است.

```
test_num_correct: 9930 total test data: 10000
model loss on testing set: 1.504620
model accuracy on testing set: 0.993000
```

شکل ۵-۸: عملکرد شبکه هدف بر روی داده تست قبل از حمله

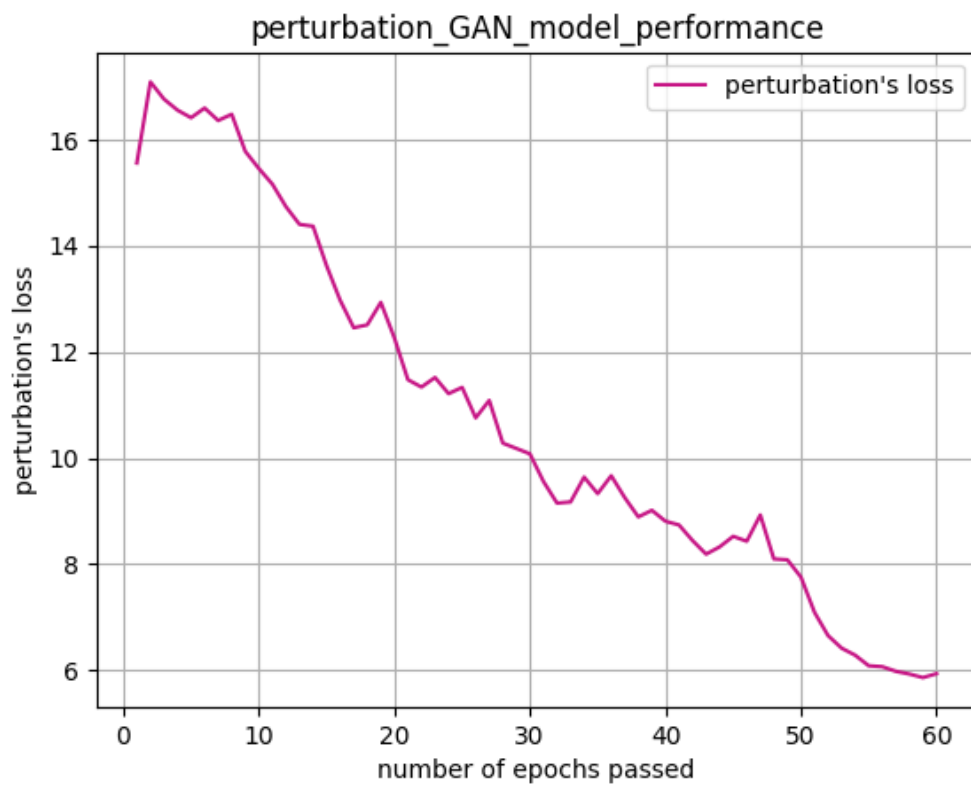
۵-۶ آموزش شبکه Adv-GAN

در شکل ۵-۹ نمودار ضرر برای مولد و تمییزدهنده شبکه Adv-GAN آورده شده است. همان طور که انتظار می‌رود هر چه تعداد دورهای بیشتری شبکه آموزش می‌بیند، مولد در تولید نمونه‌های طبیعی‌تر (ضرر نزدیک به ۱) بهتر می‌شود و همین‌طور تمییزدهنده در تشخیص نمونه تقلبی و تولید شده از نمونه اصلی و طبیعی (ضرر نزدیک به ۰) عملکرد بهتری پیدا می‌کند.



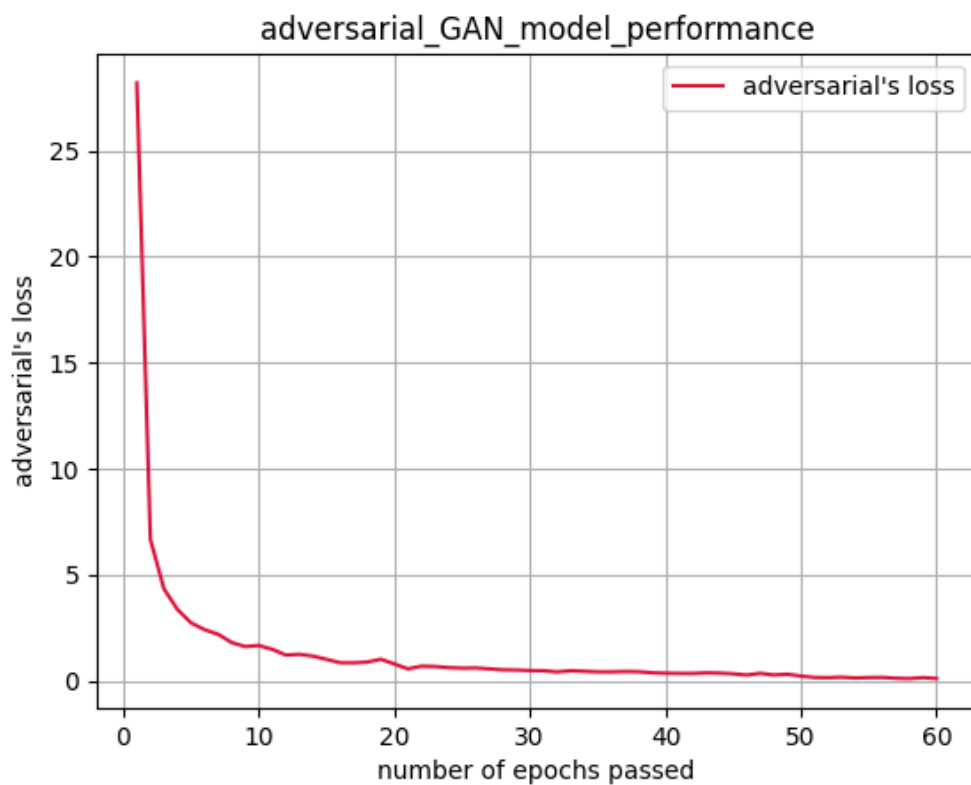
شکل ۵-۹: نمودار ضرر مولد و تمییزدهنده شبکه Adv-GAN

در شکل ۵-۱۰ نمودار ضرر دستکاری‌ای که برای تولید نمونه تقابلی بر روی تصویر اصلی اعمال می‌شود آورده شده است. طبق انتظار، هر چه تعداد دورهای بیشتری شبکه آموزش می‌بیند، شبکه *Adv-GAN* نمونه‌های تقابلی با دستکاری کمتری می‌سازد و لذا نمونه‌های تولیدی به نمونه‌های طبیعی نزدیک تر و مشابه تر (ضرر نزدیک به ۰) هستند.



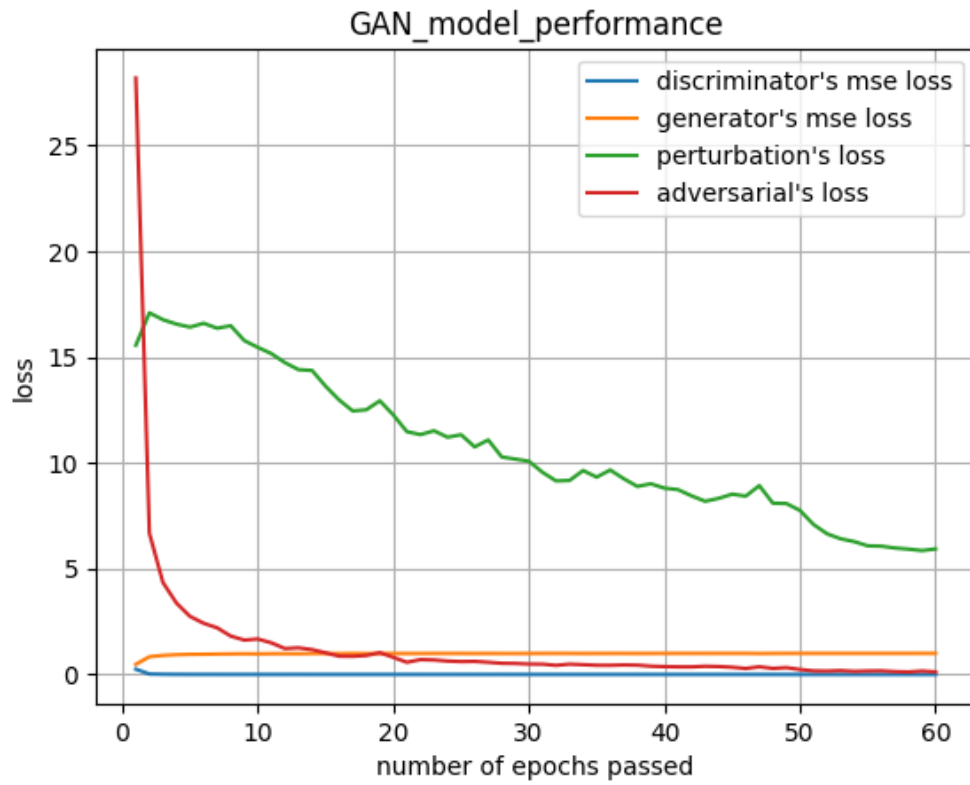
شکل ۵-۱۰: نمودار ضرر دستکاری شبکه *Adv-GAN*

در شکل ۵-۱۱ نمودار ضرر تقابلی که کمینه کردن آن هدف اصلی شبکه است آورده شده است. مشاهده می‌شود با بیشتر شدن تعداد دورهایی که شبکه آموزش دیده است، نمونه‌های تقابلی بیشتر و بیشتر می‌توانند شبکه را فریب بدهند و در دسته‌بندی متفاوت از دسته‌ی تصویر اصلی قرار بگیرند. این ضرر در اصل، عملکرد شبکه Adv-GAN در گول زدن شبکه هدف را نشان می‌دهد و هر چه این ضرر به ۰ نزدیک‌تر باشد به این معنی است که شبکه هدف در مقابل حمله ضعیف‌تر و ضعیف‌تر می‌شود.



شکل ۵-۱۱: نمودار ضرر تقابلی شبکه Adv-GAN

در شکل ۵-۱۱ نموداری از هر ۴ ضرر شبکه Adv-GAN به صورت یکجا آورده شده است.



شکل ۵-۱۲: نمودار همه ضررهای شبکه Adv-GAN

۷-۵ نتایج عملکرد شبکه هدف پس از حمله

پس از اینکه شبکه Adv-GAN آموزش دیده شد و نمونه‌های تقابلی تولید شده توسط بخش مولد این شبکه را به مدل هدف داده شد، نتایج عملکرد شبکه هدف بر روی مجموعه داده آموزش در شکل ۵-۱۳ و مجموعه داده تست در شکل ۵-۱۴ آورده شده است.

```
Training set per-class accuracy:
[(0, 0.2026000337666723), (1, 0.059329575793533075), (2, 0.35246727089627394), (3, 0.326211058554885), (4, 0.15405682985279015),
(5, 0.27670171555063644), (6, 0.08448800270361609), (7, 0.22346368715083798), (8, 0.6494616304905144), (9, 0.15128593040847202)]
Training set F1 score (micro): 0.002450
Training set F1 score (weighted): 0.002756
Training set Accuracy score: 0.245000
Training set attack success rate: 99.755000
```

شکل ۵-۱۳: عملکرد شبکه هدف بر روی داده آموزش پس از حمله

```
Testing set per-class accuracy:
[(0, 0.30612244897959184), (1, 0.1762114537444934), (2, 1.3565891472868217), (3, 0.297029702970297), (4, 0.10183299389
002037), (5, 0.336322869955157), (6, 0.10438413361169102), (7, 0.0), (8, 1.2320328542094456), (9, 0.39643211100099107)]
Testing set F1 score (micro): 0.004300
Testing set F1 score (weighted): 0.004360
Testing set Accuracy score: 0.430000
Testing set attack success rate: 99.570000
```

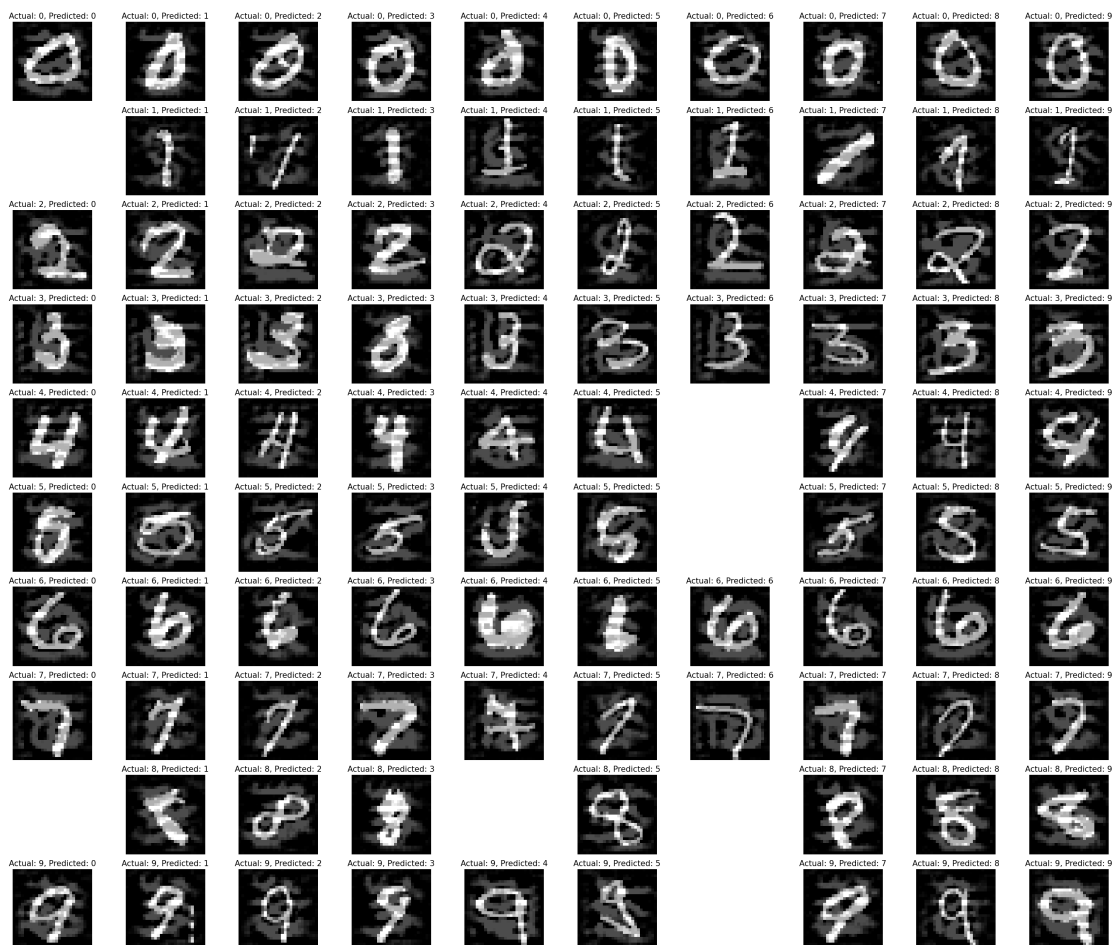
شکل ۵-۱۴: عملکرد شبکه هدف بر روی داده تست پس از حمله

همان‌طور که مشاهده می‌شود در هر دو مجموعه داده تست و آموزش میزان موفقیت حمله بالاتر از ۹۹/۵٪ است و یعنی شبکه هدف که تا قبل از این حمله تنها ۷۰ مورد از مجموعه داده تست را به درستی تشخیص نمی‌داد، الان تنها حدود ۴۳ مورد را به درستی دسته‌بندی می‌کند.

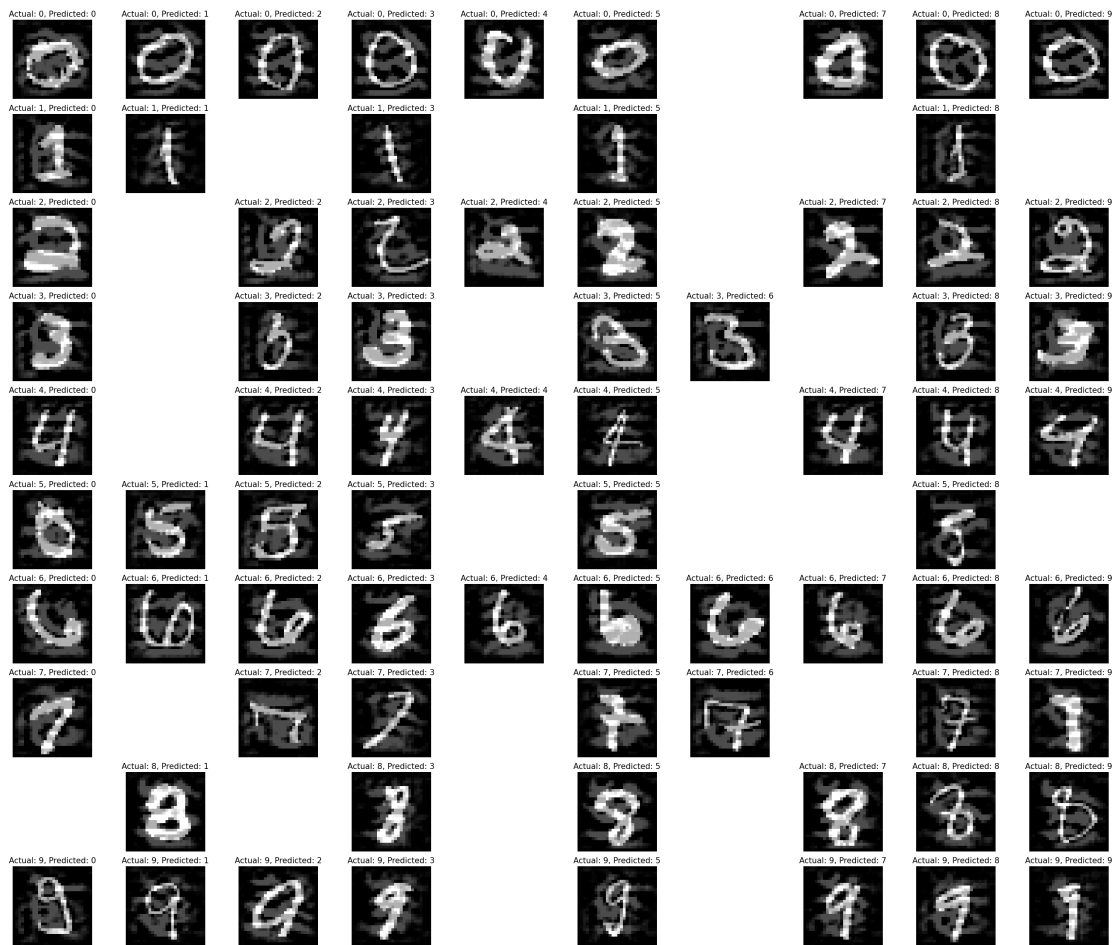
۵-۸ نمونه‌های تقابلی و مدل هدف

پس از اینکه شبکه Adv-GAN آموزش دیده شد، تنها شبکه مولد را بارگیری کرده و سپس با دادن یک نویز تصادفی نمونه تقابلی تولید می‌شود.

در شکل ۵-۱۵ و ۵-۱۶ برای حالت‌های مختلف برچسب اصلی و پیش‌بینی شده، نمونه‌های تقابلی آورده شده‌اند. در این دو شکل، ردیف‌ها نشان دهنده برچسب و کلاس اصلی و ستون‌ها نشان‌دهنده برچسب پیش‌بینی شده توسط مدل هدف هستند.



شکل ۵-۱۵: نمونه‌های تقابلی تولید شده از تصویرهای داده آموزش به همراه کلاس اصلی و کلاس پیش‌بینی شده

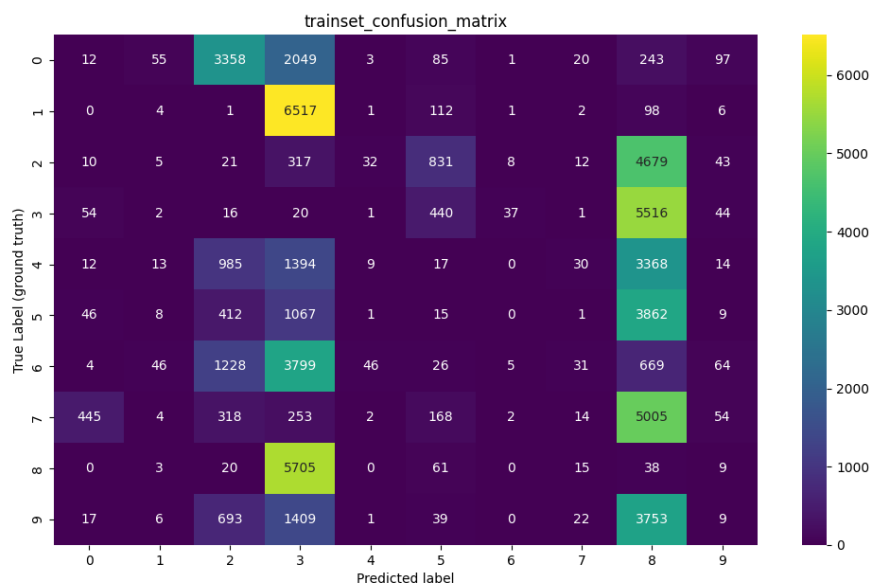


شکل ۵-۱۶: نمونه‌های تقابلی تولید شده از تصویرهای داده تست به همراه کلاس اصلی و کلاس پیش‌بینی شده

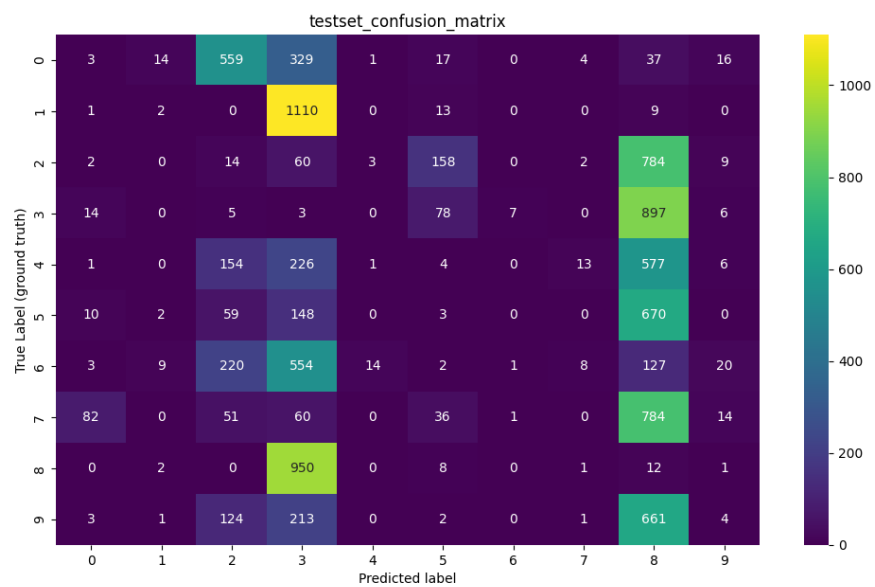
لازم به ذکر است که در شکل‌های ۵-۱۵ و ۵-۱۶ جاهای خالی به این علت وجود دارند که ممکن است مدل هدف هنگامی که یک تصویر با برچسب اصلی * به آن داده می‌شود را هیچ‌گاه شبیه به دسته‌ی ۶ طبقه بندی نکند و از این رو نمونه‌ی تقابلی‌ای برای این حالت در شکل وجود ندارد.

در شکل‌های ۵-۱۷ و ۵-۱۸ نیز ماتریس درهم‌ریختگی برای نمونه‌های تقابلی بدست آمده از داده‌های آموزش و تست آورده شده است. در این ماتریس ردیف‌ها نشان دهنده برچسب اصلی و ستون‌ها نشان دهنده برچسب پیش‌بینی شده توسط مدل هدف هستند.

قطر ماتریس نشان‌دهنده تعداد پیش‌بینی‌های صحیح برای هر برچسب و دسته (برچسب پیش‌بینی شده با برچسب اصلی یکسان است) است. همان‌طور که ملاحظه می‌شود بیشتر نمونه‌های تقابلی در دسته‌ای به غیر از دسته درست خود قرار گرفتند. همچنین مشاهده می‌شود که به‌طور مثال بیشتر نمونه‌های تقابلی‌ای که از تصویرهای کلاس ۱ آموزشی تولید شده‌اند، از نظر شبکه هدف بیشتر به کلاس ۳ شباهت داشته‌اند.



شکل ۵-۱۷: ماتریس درهم‌ریختگی برای نمونه‌های تقابلی تولید شده از تصویرهای داده آموزش



شکل ۵-۱۸: ماتریس درهم‌ریختگی برای نمونه‌های تقابلی تولید شده از تصویرهای داده تست

فصل ۶

نتیجه گیری و پیشنهادات

۶-۱ نتیجه گیری

در این پروژه روش Adv-GAN به عنوان یک روش حمله نوین و قدرتمند به شبکه‌های عصبی عمیق بررسی شد. ایده‌ی اصلی این شبکه برگرفته از شبکه‌های مولد تقابلی است. از این روش می‌توان در حمله‌های جعبه نیمه‌سفید و جعبه سیاه با درصد موفقیت حمله‌ی بالا استفاده کرد زیرا هنگامی که بخش مولد شبکه Adv-GAN آموزش دیده است، می‌تواند به صورت مستقل، دستکاری‌های تقابلی‌ای را به صورت بهینه تولید کند. نمونه‌های تقابلی تولید شده توسط این روش دارای کیفیت واقعی بسیار بالایی هستند و بنابراین این روش یک کاندید بسیار قوی در هنگام بررسی و ارزیابی شبکه‌های عصبی عمیق در برابر نمونه‌های تقابلی است. شبکه آموزش داده شده توانست عملکرد بسیار عالی شبکه هدف را از ۹۹/۳٪ به عملکرد کاملاً اشتباه ۰/۴۳٪ تنزل بدهد.

شایان توجه است که وجود این نمونه‌های تقابلی نشان می‌دهد، توانایی خوب در توضیح مجموعه داده یا حتی برچسب‌زنی دقت بالا این داده‌ها، لزوماً به این معنی نیست که مدل به درستی متوجه وظیفه‌ای که بر عهده‌اش است، شده است و در مقابل نمونه‌هایی که عمداً دستکاری شده‌اند مقاوم است.

۲-۶ پیشنهادات و کارهای آینده

آسیب‌پذیری شبکه‌های عصبی به نمونه‌های تقابلی باعث شده است که پژوهش‌ها در زمینه حمله‌های تقابلی و راه‌های دفاع آن در حال حاضر بسیار فعال شود. در برخی از پژوهش‌ها تکنیک‌هایی برای دفاع از شبکه‌های عصبی در مقابل روش‌های حمله شناخته شده، ارائه می‌شود و در سمتی دیگر روش‌هایی برای حمله‌های قدرتمندتر در حال طراحی و ارائه است. در برخی از پژوهش‌ها از شبکه‌های مولد تقابلی برای هر دو منظور استفاده شده است. امید است که این فعالیت و کوشش فراوان باعث شود تا روش‌های یادگیری عمیق بسیار مقاوم‌تر در حوزه کاربردهای حساس امنیتی و ایمنی در جهان واقعی شود و حمله‌کننده‌ها توانایی اعمال دخالت و خراب‌کاری در شبکه را نداشته باشند.

از این رو، بررسی کردن عملکرد این روش‌های حمله بر روی مجموعه داده‌های پیچیده و پردسته‌تر مانند CIFAR-100 و ImageNet و همچنین اعمال این روش‌های حمله بر روی شبکه‌های Recurrent و مجموعه داده‌های متنی و غیر تصویری قدمی دیگر به سوی شبکه‌های قابل اطمینان تر است.

مراجع

- [1] ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JÓZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVENBERG, J., MANÉ, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TALWAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIÉGAS, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., AND ZHENG, X. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *ArXiv abs/1603.04467* (2016).
- [2] AKHTAR, N., AND MIAN, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6 (2018), 14410–14430.
- [3] CARLINI, N., AND WAGNER, D. A. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)* (2017), 39–57.
- [4] EVTIMOV, I., EYKHOLT, K., FERNANDES, E., KOHNO, T., LI, B., PRAKASH, A., RAHMATI, A., AND SONG, D. Robust physical-world attacks on deep learning models. *arXiv: Cryptography and Security* (2017).
- [5] GIUSTI, A., GUZZI, J., CIRESAN, D., HE, F., RODRÍGUEZ, J. P., FONTANA, F., FAESSLER, M., FORSTER, C., SCHMIDHUBER, J., CARO, G. D., SCARAMUZZA, D., AND GAMBARDELLA, L. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters* 1 (2016), 661–667.
- [6] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. C., AND BENGIO, Y. Generative adversarial nets. in *NIPS* (2014).
- [7] GOODFELLOW, I., SHLENS, J., AND SZEGEDY, C. Explaining and harnessing adversarial examples. *CoRR abs/1412.6572* (2015).

- [8] GROSSE, K., PAPERNOT, N., MANOHARAN, P., BACKES, M., AND MCDANIEL, P. Adversarial examples for malware detection. in *ESORICS* (2017).
- [9] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.
- [10] HEAVEN, D. Why deep-learning aIs are so easy to fool.
- [11] HELMSTAEDTER, M., BRIGGMAN, K., TURAGA, S. C., JAIN, V., SEUNG, H., AND DENK, W. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature* 500 (2013), 168–174.
- [12] HINTON, G. E., DENG, L., YU, D., DAHL, G. E., RAHMAN MOHAMED, A., JAITLY, N., SENIOR, A., VANHOUCKE, V., NGUYEN, P., SAINATH, T., AND KINGSBURY, B. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine* 29 (2012), 82.
- [13] JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R. B., GUADARRAMA, S., AND DARRELL, T. Caffe: Convolutional architecture for fast feature embedding. *Proceedings of the 22nd ACM international conference on Multimedia* (2014).
- [14] KURAKIN, A., GOODFELLOW, I., AND BENGIO, S. Adversarial examples in the physical world. *ArXiv abs/1607.02533* (2017).
- [15] KURAKIN, A., GOODFELLOW, I., AND BENGIO, S. Adversarial machine learning at scale. *ArXiv abs/1611.01236* (2017).
- [16] MADRY, A., MAKELOV, A., SCHMIDT, L., TSIPRAS, D., AND VLADU, A. Towards deep learning models resistant to adversarial attacks. *ArXiv abs/1706.06083* (2018).
- [17] MNIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A. A., VENESS, J., BELLEMARE, M. G., GRAVES, A., RIEDMILLER, M. A., FIDJELAND, A., OSTROVSKI, G., PETERSEN, S., BEATTIE, C., SADIK, A., ANTONOGLOU, I., KING, H., KUMARAN, D., WIERSTRA, D., LEGG, S., AND HASSABIS, D. Human-level control through deep reinforcement learning. *Nature* 518 (2015), 529–533.
- [18] MOOSAVI-DEZFOOLI, S.-M., FAWZI, A., AND FROSSARD, P. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 2574–2582.

- [19] NAJAFABADI, M. M., VILLANUSTRE, F., KHOSHGOFTAAR, T., SELIYA, N., WALD, R., AND MUHAREMAGIC, E. A. Deep learning applications and challenges in big data analytics. *Journal of Big Data 2* (2014), 1–21.
- [20] PAPERNOT, N., MCDANIEL, P., SINHA, A., AND WELLMAN, M. P. Towards the science of security and privacy in machine learning. *ArXiv abs/1611.03814* (2016).
- [21] SU, J., VARGAS, D. V., AND SAKURAI, K. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation 23* (2019), 828–841.
- [22] SUTSKEVER, I., VINYALS, O., AND LE, Q. V. Sequence to sequence learning with neural networks. in *NIPS* (2014).
- [23] SZEGEDY, C., VANHOUCHE, V., IOFFE, S., SHLENS, J., AND WOJNA, Z. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 2818–2826.
- [24] VEDALDI, A., AND LENC, K. Matconvnet: Convolutional neural networks for matlab. *Proceedings of the 23rd ACM international conference on Multimedia* (2015).
- [25] XIAO, C., LI, B., ZHU, J.-Y., HE, W., LIU, M., AND SONG, D. Generating adversarial examples with adversarial networks. *ArXiv abs/1801.02610* (2018).
- [26] XIONG, H. Y., ALIPANAHI, B., LEE, L. J., BRETSCHNEIDER, H., MERICO, D., YUEN, R., HUA, Y., GUEROUSSOV, S., NAJAFABADI, H., HUGHES, T., MORRIS, Q., BARASH, Y., KRAINER, A., JOJIC, N., SCHERER, S., BLENCOWE, B., AND FREY, B. The human splicing code reveals new insights into the genetic determinants of disease. *Science 347* (2015).

Abstract:

Since many researches have been made on the vulnerability of Deep neural networks (DNNs), in this project, we tried to generate a more realistic and efficiently adversarial examples.

More precisely, we used Adv-GAN framework to learn and approximate the distribution of original instances. Afterwards, once the Adv-GAN is trained, we used the generator to produce high perceptual quality adversarial examples.

The generator can be efficiently and independently, used to generate adversarial perturbations for any given instance. Therefore, this attack can be used in both semi-whitebox and black-box attacks. Using this method we could perform a 99.57% attack success rate on the target model, which had a 99.3% accuracy on MNIST dataset before the attack.

Keywords: Generative Adversarial Network, Adversarial Example, Fast Gradient Sign Method, Adversarial Perturbations, Deep neural networks



Iran University of Science and Technology
Computer Engineering Department

Generating adversarial examples using Generative Adversarial Networks (GANs)

Bachelor of Science Thesis in Computer Engineering

By:

Yeganeh Morshedzadeh

Supervisor:

Dr. Nasser Mozayani

Summer 2021