

# شبکه‌های عصبی و یادگیری عمیق

## دکتر صفا بخش



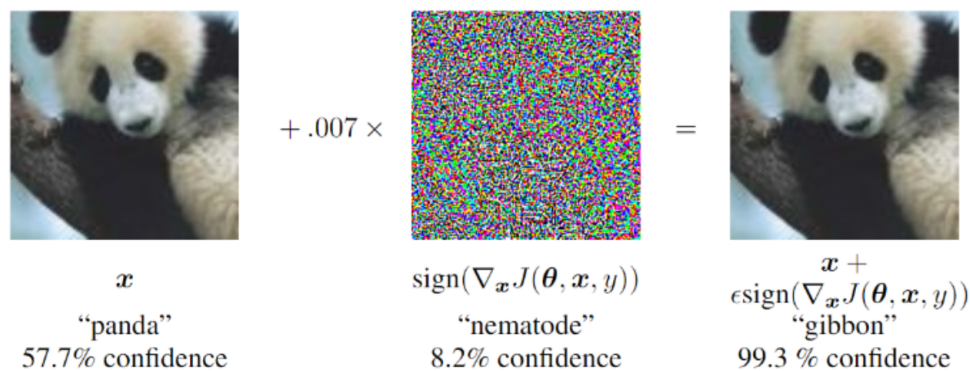
**دانشگاه صنعتی امیرکبیر**  
( پلی تکنیک تهران )  
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین هشتم  
ساختارهای Encoder و Decoder

۲۵ تیر ۱۴۰۳

حملات خصمانه<sup>۱</sup> نوعی از حملات بر روی مدل‌های یادگیری ماشین به منظور فریب دادن مدل با استفاده از ورودی‌های دستکاری شده است. هدف اصلی این حملات تغییر خروجی مدل به صورت اشتباه است. به سوالات زیر پاسخ دهید و به منبع یا منابعی که استفاده کردید ارجاع دهید.



شکل ۱: تغییر نمونه ورودی

## سوال اول - تئوری

یکی از اولین و ساده‌ترین روش‌های حمله خصمانه، FGSM است که توسط یان گودفلو و همکارانش<sup>۲</sup> معرفی شد. هدف این روش، ایجاد یک نمونه خصمانه است که تفاوت بسیار کمی با ورودی اصلی داشته باشد اما مدل را به اشتباه بیندازد. PGD یک روش قوی‌تر و بهبود یافته نسبت به FGSM است که توسط Madry و همکارانش<sup>۳</sup> معرفی شده. این روش به جای انجام یک مرحله، بروز رسانی‌های متعددی را انجام می‌دهد و در هر مرحله تغییرات را در محدوده مشخصی پروجکت می‌کند تا اطمینان حاصل شود که نمونه خصمانه بیش از حد از ورودی اصلی فاصله نگیرد. این دو روش را مطالعه و خلاصه‌ای از آن‌ها بنویسید.

پاسخ

### • روش FGSM:

این روش در سال ۲۰۱۵ در مقاله *Explaining and Harnessing Adversarial Examples* معرفی شد که یکی از ساده‌ترین روش‌های حملات خصمانه است. همانطور که در صورت سوال نیز توضیح داده شد، هدف اصلی FGSM ایجاد نمونه‌های خصمانه‌ای است که از نظر بصری تفاوت زیادی با ورودی اصلی نداشته باشند ولی باعث شوند مدل یادگیری ماشین خطا کند.

<sup>۱</sup> Adversarial Attack

<sup>۲</sup> Examples Adversarial Harnessing and Explaining

<sup>۳</sup> Attacks Adversarial to Resistant Models Learning Deep Towards

## پاسخ

در ادامه، روش انجام این الگوریتم را به صورت خلاصه توضیح می‌دهیم:

- FGSM با اضافه کردن یک اختلال به داده‌های ورودی اصلی کار می‌کند.
- این اختلال با استفاده از گرادیان تابع هزینه نسبت به داده‌های ورودی محاسبه می‌شود.
- به طور خاص، می‌توان فرمول این اختلال را به صورت زیر بیان نمود:

$$\eta = \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

که در آن:

- $\epsilon$  یک مقدار عددی کوچک است که میزان اختلال را کنترل می‌کند.
- $\nabla_x J(\theta, x, y)$  گرادیان تابع هزینه  $J$  نسبت به ورودی  $x$  است.
- $\text{sign}$  تابع علامت است که علامت گرادیان را استخراج می‌کند.
- سپس نمونه‌ی خصمانه با اضافه کردن این اختلال به ورودی اصلی ایجاد می‌شود:

$$x' = x + \eta$$

از مزایا و معایب FGSM می‌توان به موارد زیر اشاره کرد:  
مزایا: از نظر محاسباتی کارآمد و ساده است و به همین دلیل انتخاب محبوبی برای مطالعات اولیه در مورد حملات خصمانه است.  
معایب: سادگی FGSM می‌تواند به عنوان یک نقطه ضعف نیز عمل کند، زیرا اختلالات ایجاد شده ممکن است در مقابل مدل‌های مقاوم‌تر به اندازه‌ی کافی مؤثر نباشند.

### • روش PGD:

همانطور که در صورت سوال نیز گفته شد، روش PGD یک روش بهبود یافته نسبت به FGSM است که در سال ۲۰۱۷ در مقاله‌ی *Towards Deep Learning Models Resistant to Adversarial Attacks* معرفی شد. هدف اصلی این روش، ایجاد نمونه‌های خصمانه قوی‌تر با انجام چندین به‌روزرسانی گرادیانی و اطمینان از ماندن اختلالات در یک محدوده‌ی مشخص است.  
در ادامه، روش انجام این الگوریتم را توضیح می‌دهیم:

- PGD به طور مکرر نمونه‌ی خصمانه را با گرفتن چندین گام گرادیانی اصلاح می‌کند.
- هر گام شامل مراحل زیر است:

۱. محاسبه‌ی گرادیان: گرادیان تابع هزینه نسبت به نمونه‌ی خصمانه‌ی فعلی محاسبه می‌شود.
۲. گام به‌روزرسانی: نمونه‌ی خصمانه با حرکت در جهت گرادیان به‌روزرسانی می‌شود:

$$x_{t+1} = x_t + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_t, y))$$

که  $\alpha$  اندازه‌ی گام است.

۳. پروژه‌سازی: اطمینان حاصل می‌شود که نمونه‌ی خصمانه‌ی به‌روز شده در یک کره‌ی  $\epsilon$  حول ورودی اصلی باقی می‌ماند:

$$x_{t+1} = \text{clip}(x_{t+1}, x - \epsilon, x + \epsilon)$$

این مرحله اطمینان می‌دهد که اختلال از محدوده‌ی مجاز تجاوز نمی‌کند.

پاسخ

این فرآیند برای تعداد معینی از تکرارها یا تا همگرایی تکرار می‌شود.  
از مزایا و معایب PGD می‌توان به موارد زیر اشاره کرد:

۱. مزایا: PGD به دلیل اینکه فضای اختلالات ممکن را با تکرارهای متعدد به طور جامع‌تری کاوش می‌کند، به عنوان یک روش حمله‌ی قوی‌تر نسبت به FGSM شناخته می‌شود.
۲. معایب: طبیعت تکراری PGD باعث می‌شود که از نظر محاسباتی پرهزینه‌تر از FGSM باشد.

\*  
—————

## References

- [1] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR) [\[Link\]](#)
- [2] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In Proceedings of the International Conference on Learning Representations (ICLR) [\[Link\]](#)

## سوال دوم - تئوری

چگونه آموزش خصمانه<sup>۴</sup> می‌تواند بر تعمیم‌پذیری مدل به داده‌های دیده نشده تاثیر بگذارد؟ آیا همیشه بهبود در مقاومت شدن در برابر حملات، بهبود صحت بر روی داده‌های دیده نشده را تضمین می‌کند؟ نشان دهید.

پاسخ

## ۱. توازن بین مقاومت و دقت:

- بهبود مقاومت: آموزش خصمانه به طور عمده مقاومت مدل در برابر حملات خصمانه را بهبود می‌بخشد. با معرفی نمونه‌های خصمانه به مدل در طول آموزش، مدل یاد می‌گیرد که حتی در مواجهه با ورودی‌های دستکاری شده نیز پیش‌بینی‌های دقیقی انجام دهد.
- دقت در داده‌های تمیز: در حالی که آموزش خصمانه می‌تواند مقاومت را به طور قابل توجهی افزایش دهد، اغلب با کاهش دقت در داده‌های تمیز (غیر دستکاری شده) همراه است. این به این دلیل است که مدل ممکن است بیش از حد بر روی نمونه‌های خصمانه تمرکز کند و توانایی تعمیم‌دهی به نمونه‌های عادی و غیر خصمانه کاهش یابد.

## ۲. تعمیم‌پذیری به داده‌های دیده نشده:

- توانایی تعمیم‌دهی: در برخی موارد، مقاومتی که توسط آموزش خصمانه ایجاد می‌شود، می‌تواند به مدل کمک کند تا به داده‌های دیده نشده خاصی بهتر تعمیم دهد، به ویژه اگر داده‌های دیده نشده شامل نویز یا تغییرات مشابه نمونه‌های خصمانه باشد. با این حال، اگر داده‌های دیده نشده تمیز باشند، ممکن است مدل نسبت به مدلی که بدون نمونه‌های خصمانه آموزش دیده، عملکرد ضعیف‌تری داشته باشد.
- تنوع توزیع داده‌ها: قابلیت تعمیم‌دهی مدل همچنین به این بستگی دارد که نمونه‌های خصمانه چقدر نماینده تغییرات احتمالی در داده‌های دیده نشده هستند. اگر نمونه‌های خصمانه به خوبی طراحی شده باشند و طیف وسیعی از تغییرات ممکن را پوشش دهند، می‌توانند به مدل کمک کنند تا بهتر تعمیم دهد. در مقابل، اگر نمونه‌های خصمانه بیش از حد خاص باشند، ممکن است مدل با توزیع داده‌های جدیدی که به خوبی نمایندگی نشده‌اند، مشکل داشته باشد.

بهبود مقاومت در برابر حملات خصمانه در مقابل دقت بر روی داده‌های دیده نشده همیشه تضمین نمی‌شود، بهبود مقاومت در برابر حملات خصمانه همیشه دقت بهتر بر روی داده‌های دیده نشده را تضمین نمی‌کند. فرآیند آموزش خصمانه می‌تواند گاهی منجر به overfitting به نمونه‌های خصمانه شود که ممکن است به خوبی به نمونه‌های جدید و دیده نشده تعمیم ندهد. دستیابی به تعادل بین مقاومت و دقت نیاز به توجه دقیق به داده‌های آموزشی و انواع نمونه‌های خصمانه مورد استفاده دارد. تکنیک‌های regularization، data augmentation و سایر روش‌ها می‌توانند به کاهش تأثیر منفی بر دقت کمک کنند در حالی که مقاومت را نیز بهبود می‌بخشند.

<sup>۴</sup>Adversarial Training

## سوال سوم - تئوری

چرا و چگونه نمونه‌های خصمانه‌ای ایجاد شده برای یک مدل می‌توانند مدل‌های دیگر را نیز فریب دهند؟ این خاصیت انتقال‌پذیری چگونه می‌تواند در حملات جعبه سیاه استفاده شود؟

پاسخ

یک جنبه جالب و مهم از نمونه‌های خصمانه خاصیت انتقال‌پذیری آنهاست، یک نمونه خصمانه که برای فریب یک مدل طراحی شده است، می‌تواند مدل‌های دیگر را نیز فریب دهد. از این پدیده در حملات جعبه سیاه استفاده می‌شود، جایی که مهاجم دسترسی مستقیم به مدل هدف ندارد. این موضوع می‌تواند به دلایل زیر باشد:

۱. شباهت در ویژگی‌های آموخته شده

۲. داده‌های آموزشی مشترک

۳. آسیب‌پذیری‌های مشترک:

مدل‌های مختلف، به ویژه آنهایی که بر روی همان مجموعه داده یا با معماری‌های مشابه آموزش دیده‌اند، تمایل به یادگیری ویژگی‌های مشابه دارند. مثال‌های خصمانه از ضعف‌های این ویژگی‌های آموخته شده سوءاستفاده می‌کنند، که باعث می‌شود به احتمال زیاد بین مدل‌ها انتقال یابند.

۴. ماهیت خطی مدل‌ها:

یساری از مدل‌های یادگیری ماشین، به ویژه شبکه‌های عصبی، طبیعتی خطی دارند. تغییرات خصمانه از این خطی بودن سوءاستفاده می‌کنند و از آنجا که این ویژگی بین مدل‌ها مشترک است، مثال‌های خصمانه می‌توانند انتقال پیدا کنند.

۵. همپوشانی در مرزهای تصمیم‌گیری:

مدل‌هایی که بر روی داده‌های مشابه آموزش دیده‌اند، تمایل به داشتن مرزهای تصمیم‌گیری همپوشان دارند. مثال‌های خصمانه‌ای که برای عبور از مرز تصمیم‌گیری یک مدل ساخته شده‌اند، به احتمال زیاد از مرزهای تصمیم‌گیری مدل‌های دیگر که بر روی داده‌های مشابه آموزش دیده‌اند نیز عبور می‌کنند.

در حملات جعبه سیاه، مهاجم دسترسی مستقیم به پارامترهای مدل هدف یا داده‌های آموزشی آن ندارد. در عوض، آنها از خاصیت انتقال‌پذیری مثال‌های خصمانه بهره می‌برند. به همین دلیل می‌توان مدلی جایگزین را آموزش داد که مدل هدف را تقریب بزند. این کار می‌تواند با استفاده از داده‌های مشابه یا تقلید از رفتار مدل هدف انجام شود.

هنگامی که مدل جایگزین آموزش دید، نمونه‌های خصمانه با استفاده از این مدل ساخته می‌شوند. سپس این نمونه‌های ساخته شده، به مدل هدف اعمال می‌شوند. به دلیل خاصیت انتقال‌پذیری، این نمونه‌ها احتمال بالایی برای فریب مدل هدف دارند.

## سوال چهارم - تئوری

چگونه می‌توان حملات خصمانه را در حوزه‌هایی مانند پردازش زبان طبیعی پیاده‌سازی کرد؟ چه چالش‌های خاصی در این حوزه وجود دارد؟

پاسخ

حملات خصمانه در حوزه پردازش زبان طبیعی شامل ایجاد ورودی‌های متنی است که به طور عمدی باعث می‌شود مدل‌های NLP خروجی‌های نادرست یا نامطلوب تولید کنند. این حملات می‌توانند طیف وسیعی از کاربردهای NLP مانند تحلیل احساسات، ترجمه ماشینی و طبقه‌بندی متن را هدف قرار دهند. پیاده‌سازی حملات خصمانه در NLP نسبت به حوزه‌های دیگر مانند پردازش تصویر، چالش‌های منحصر به فردی دارد. در ادامه، به نحوه پیاده‌سازی این حملات و چالش‌های خاص آنها می‌پردازیم:

### ۱. پیاده‌سازی حملات خصمانه در NLP

#### (آ) ایجاد اختلال در متن:

- جایگزینی مترادف‌ها: جایگزینی کلمات با مترادف‌هایشان برای ایجاد تغییرات جزئی در ورودی بدون تغییر کلی معنای آن. به عنوان مثال، جایگزینی خوشحال با شاد.
- ایجاد اختلال در سطح کاراکتر: وارد کردن اشتباهات کوچک یا جابجایی کاراکترها، مانند تغییر سلام به سللام.
- افزودن/حذف کلمات: افزودن یا حذف کلمات برای تغییرات جزئی در ورودی. برای مثال، اضافه کردن کلمه نه برای تغییر احساس جمله.
- بازنویسی: بازنویسی جمله با حفظ معنای اصلی آن اما با تغییر ساختار و انتخاب کلمات.

#### (ب) حملات مبتنی بر گرادیان:

- روش HotFlip: روشی که کاراکترها را در متن بر اساس گرادیان‌های مدل تغییر می‌دهد تا حداقل تغییراتی که تأثیر زیادی بر خروجی مدل دارد را پیدا کند.
- روش TextFooler: تکنیکی که با استفاده از گرادیان‌ها، مهم‌ترین کلمات در متن را شناسایی و آنها را با مترادف‌ها یا کلمات مشابه معنایی جایگزین می‌کند.

#### (ج) حملات جعبه سیاه:

- الگوریتم‌های ژنتیک: استفاده از الگوریتم‌های تکاملی برای تغییر تدریجی متن و ایجاد مثال‌های خصمانه بدون نیاز به دسترسی به گرادیان‌های مدل.
- حملات مبتنی بر پرس‌وجو: ارسال پرس‌و‌جوه‌های متعدد به مدل و مشاهده خروجی‌ها برای ساختن مثال‌های خصمانه، با استفاده از قابلیت انتقال‌پذیری حملات از مدل‌های جایگزین.

### ۲. چالش‌های خاص در حملات خصمانه NLP

- (آ) ماهیت گسسته متن: برخلاف تصاویر که می‌توان مقادیر پیکسلی را به طور پیوسته تغییر داد، متن گسسته است و هر تغییری باید منجر به جملات معتبر و معنادار شود. این امر ایجاد مثال‌های خصمانه را پیچیده‌تر می‌کند.
- (ب) حفظ معنا: اطمینان از اینکه تغییرات خصمانه معنای اصلی متن را تغییر نمی‌دهد بسیار مهم است. تغییرات باید برای خوانندگان انسانی غیرقابل تشخیص باشد در حالی که همچنان بر خروجی مدل تأثیر می‌گذارد.
- (ج) دستور زبان: اختلالات باید دستور زبان و ساختار نحوی صحیح را حفظ کنند تا از ایجاد جملات بی‌معنی یا نادرست از نظر گرامری جلوگیری کنند. این برای واقعی و منطقی بودن متن خصمانه مهم است.

## ۳. نمونه‌هایی از تکنیک‌های حمله خصمانه در NLP

- (آ) HotFlip: از اطلاعات گرادیان یک مدل در سطح کاراکتر استفاده می‌کند تا موثرترین تغییرات کاراکتری را که می‌تواند پیش‌بینی مدل را تغییر دهد شناسایی کند.
- (ب) TextFooler: مهم‌ترین کلمات در متن را که بر خروجی مدل تأثیر می‌گذارند شناسایی و آنها را با مترادف‌ها یا کلمات مشابه معنایی جایگزین می‌کند.
- (ج) PWWS (احتمال وزن‌دار اهمیت کلمات): کلماتی را برای جایگزینی انتخاب می‌کند که برای پیش‌بینی مدل اهمیت دارند و احتمال حفظ معنای اصلی جمله بعد از جایگزینی را دارند.

## ۴. کاربردهای عملی و استراتژی‌های مقابله

## (آ) کاربردها:

- تحلیل احساسات: گمراه کردن مدل‌های تحلیل احساسات برای طبقه‌بندی اشتباه نظرات مثبت به عنوان منفی یا بالعکس.
- تشخیص هرزنامه: ایجاد پیام‌های هرزنامه که از فیلترهای هرزنامه عبور کنند.
- ترجمه ماشینی: وارد کردن خطاهای جزئی در ترجمه‌ها.

## (ب) استراتژی‌های مقابله:

- آموزش خصمانه: آموزش مدل‌ها با مثال‌های خصمانه برای بهبود مقاومت.
- تقطیر دفاعی: کاهش حساسیت مدل به تغییرات جزئی با فشردن دانش مدل به شکل ساده‌تر.
- پیش‌پردازش ورودی‌ها: نرمال‌سازی ورودی‌های متنی برای حذف احتمالی اختلالات قبل از رسیدن به مدل.



## سوال پنجم - تئوری

چگونه می‌توان آموزش خصمانه را در مجموعه داده‌های نامتوازن پیاده‌سازی کرد و چه چالش‌هایی در این مسیر وجود دارد؟

پاسخ

آموزش خصمانه بر روی مجموعه داده‌های نامتوازن چالش‌های خاصی دارد و نیازمند توجه دقیق به منظور اطمینان از مقاومت در برابر حملات خصمانه و عملکرد خوب در کلاس‌های اقلیتی است. در ادامه به نحوه پیاده‌سازی آموزش خصمانه بر روی مجموعه داده‌های نامتوازن و چالش‌های خاص آن می‌پردازیم

## ۱. پیاده‌سازی آموزش خصمانه بر روی مجموعه داده‌های نامتوازن

(آ) ایجاد مثال‌های خصمانه:

- از تکنیک‌هایی مانند FGSM (روش گرادیان سریع)، PGD (گرادیان پروژه شده) یا سایر الگوریتم‌های حمله خصمانه برای ایجاد مثال‌های خصمانه استفاده کنید.
- می‌بایست اطمینان حاصل شود که مثال‌های خصمانه برای هر دو کلاس اکثریت و اقلیت ایجاد می‌شوند تا مدل به سمت کلاس اکثریت منحرف نشود.

(ب) آموزش خصمانه متوازن:

- آموزش خصمانه با وزن‌دهی به کلاس‌ها: به مثال‌های خصمانه از کلاس‌های اقلیت وزن بیشتری اختصاص دهید تا تاثیر بیشتری بر فرآیند یادگیری مدل داشته باشند.
- افزایش تعداد نمونه‌ها: برای ایجاد تعادل در مجموعه داده‌های آموزش خصمانه، تعداد بیشتری از مثال‌های خصمانه برای کلاس‌های اقلیت ایجاد کنید. این کار می‌تواند با ایجاد چندین مثال خصمانه از هر نمونه کلاس اقلیت انجام شود.
- کاهش تعداد نمونه‌ها: تعداد مثال‌های خصمانه برای کلاس اکثریت را کاهش دهید تا مجموعه داده‌های آموزشی متوازن شوند. این کار باید با احتیاط انجام شود تا از دست دادن اطلاعات مهم کلاس اکثریت جلوگیری شود.

(ج) روش‌های ترکیبی:

- ترکیب تکنیک‌های افزایش و کاهش تعداد نمونه‌ها برای حفظ یک مجموعه داده آموزشی خصمانه متوازن.
- استفاده از تکنیک‌های افزایش داده‌ها برای ایجاد نمونه‌های مصنوعی برای کلاس‌های اقلیت و اطمینان از تنوع و جلوگیری از بیش‌برازش.

(د) آموزش خصمانه تطبیقی:

- تنظیم پویا قدرت حمله خصمانه بر اساس توزیع کلاس‌ها. به عنوان مثال، استفاده از اختلالات قوی‌تر برای کلاس‌های اکثریت و اختلالات ملایم‌تر برای کلاس‌های اقلیت برای حفظ تعادل.

(ه) توابع زیان متوازن برای کلاس‌ها:

- پیاده‌سازی توابع زیانی که عدم توازن کلاس‌ها را در نظر می‌گیرند، مانند زیان کانونی که زیان اختصاص داده شده به مثال‌های به‌خوبی طبقه‌بندی شده را کاهش می‌دهد و تمرکز بیشتری به مثال‌های سخت و کلاس‌های اقلیت می‌دهد.
- استفاده از تکنیک‌های یادگیری حساس به هزینه که برای طبقه‌بندی نادرست نمونه‌های کلاس اقلیت جریمه‌های بالاتری در نظر می‌گیرند.

## ۲. چالش‌ها در آموزش خصمانه بر روی مجموعه داده‌های نامتوازن

(آ) حفظ تعادل:

- ایجاد یک مجموعه داده متوازن از مثال‌های خصمانه چالش برانگیز است زیرا کلاس اکثریت معمولاً داده‌های بیشتری دارد، که منجر به عدم توازن در مثال‌های خصمانه نیز می‌شود.
- افزایش نمونه‌های کلاس‌های اقلیت می‌تواند منجر به بیش‌برازش شود، در حالی که کاهش نمونه‌های کلاس‌های اکثریت می‌تواند به از دست رفتن اطلاعات مهم منجر شود.

(ب) اختلالات موثر:

- ایجاد مثال‌های خصمانه موثر برای کلاس‌های اقلیت سخت‌تر است به دلیل محدودیت تعداد نمونه‌ها، که می‌تواند بر مقاومت و تعمیم مدل تأثیر بگذارد.
- اطمینان از اینکه مثال‌های خصمانه برای کلاس‌های اقلیت بی‌اهمیت یا خیلی آسان برای مدل نباشند.

(ج) پیچیدگی محاسباتی:

- آموزش خصمانه محاسباتی سنگین است و متوازن کردن آن برای مجموعه داده‌های نامتوازن پیچیدگی را افزایش می‌دهد. ایجاد مثال‌های خصمانه متعدد برای کلاس‌های اقلیت و حفظ تنوع به بار محاسباتی می‌افزاید.

(د) معیارهای ارزیابی:

- ارزیابی عملکرد مدل‌های آموزش دیده خصمانه بر روی مجموعه داده‌های نامتوازن پیچیده است. معیارهایی مانند دقت کافی نیستند؛ معیارهایی مانند دقت، فراخوانی، امتیاز F1 و AUC-ROC اطلاعات بیشتری برای مجموعه داده‌های نامتوازن فراهم می‌کنند.
- اطمینان از عملکرد خوب مدل بر روی مثال‌های تمیز و خصمانه برای کلاس‌های اقلیت بدون کاهش عملکرد کلی.

(ه) توازن بین مقاومت و دقت:

- آموزش خصمانه اغلب منجر به توازن بین مقاومت و دقت می‌شود. اطمینان از اینکه این توازن به طور نامتناسبی کلاس‌های اقلیت را تحت تأثیر قرار نمی‌دهد بسیار مهم است.
- حفظ مقاومت مدل در برابر حملات خصمانه در حالی که دقت بالایی برای هر دو کلاس اکثریت و اقلیت دارد.

## ۳. استراتژی‌های کاهش چالش‌ها

- (آ) استفاده از روش‌های گروهی: آموزش چندین مدل با تکنیک‌های مختلف متوازن‌سازی و ترکیب پیش‌بینی‌های آنها برای دستیابی به عملکرد بهتر در هر دو کلاس اکثریت و اقلیت.
- (ب) تکنیک‌های منظم‌سازی: استفاده از روش‌های منظم‌سازی مانند دراپ‌آوت، کاهش وزن و توقف زودهنگام برای جلوگیری از بیش‌برازش، به ویژه هنگام افزایش نمونه‌های کلاس‌های اقلیت.
- (ج) افزایش داده‌های پیشرفته: استفاده از تکنیک‌های افزایش داده‌های پیچیده که نمونه‌های مصنوعی با کیفیت بالا برای کلاس‌های اقلیت تولید می‌کنند و نمایندگی آنها را در مجموعه داده‌ها افزایش می‌دهند.
- (د) آموزش خصمانه پویا: پیاده‌سازی استراتژی‌های آموزشی پویا که تمرکز بین کلاس‌های اکثریت و اقلیت را بر اساس پیشرفت آموزش و معیارهای عملکرد مشاهده شده تنظیم می‌کنند.

## سوال ششم - عملی

در این سوال می‌خواهیم یک حمله خصمانه با روش‌های FGSM طراحی کنیم و سپس مدل از پیش آموزش داده شده ResNet18 را با آموزش خصمانه مقاوم سازیم. به این منظور مراحل زیر را دنبال کنید:

۱. مدل از پیش آموزش دیده ResNet18 را برای مجموعه داده CIFAR10 آموزش دهید. نمودار خطا آموزش و آزمون را رسم کنید.
۲. روش FGSM را پیاده‌سازی کنید و ۵ تصویر را به صورت تصادفی انتخاب کنید و به مدل حمله کنید. سپس برای این تصاویر، تصویر اصلی، تصویر آشفته شده<sup>۵</sup>، پرچسب اصلی و پرچسب پیش‌بینی شده بر روی تصویر آشفته شده را نمایش دهید.
۳. حال با گنجاندن نمونه‌های خصمانه در فرآیند آموزش، مدل ResNet18 را دوباره آموزش دهید (آموزش خصمانه). این فرآیند به مدل کمک می‌کند تا در برابر حملات خصمانه مقاوم‌تر شود. نحوه آموزش را کامل شرح دهید. نمودارهای زیر را در کنار هم رسم و تفسیر کنید.

- train-natural: خطای آموزش بر روی مدل طبیعی
- train-adversary: خطای آموزش بر روی مدل خصمانه
- test-natural: خطای آموزش بر روی مدل طبیعی (مجموعه داده آزمون بدون تغییر)
- test-adversary: خطای آموزش بر روی مدل خصمانه (مجموعه داده آزمون بدون تغییر)

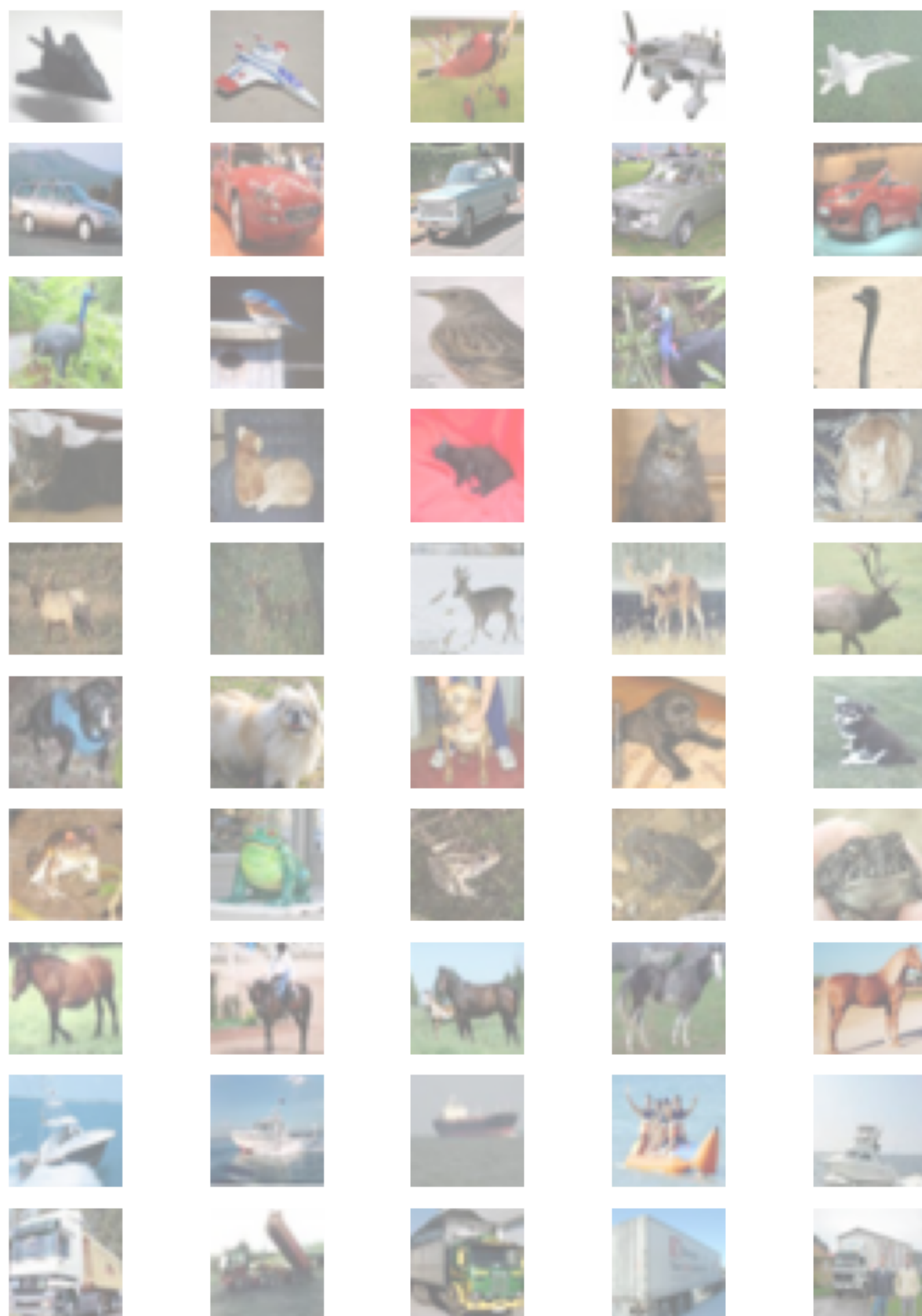
۴. تا اینجا ما توانستیم تا با حملات خصمانه تصویری که تفاوت بسیار کمی با دیتای اصلی دارد، مدل را به اشتباه بیندازیم. حال می‌خواهیم به صورت هدفمند اینکار را انجام دهیم؛ یعنی مدل باید به اشتباه کلاس مورد نظر ما را پیش‌بینی کند<sup>۶</sup>. با روش FGSM حمله هدفمند را پیاده‌سازی و نحوه انجام آن را بطور کامل شرح دهید. حال با ایجاد نمونه‌های خصمانه جدید از مجموعه داده آزمون و همچنین داده‌های آزمون بدون تغییر، صحت هر دو مدل را (مدل طبیعی و مدل آموزش دیده به صورت خصمانه) را ارزیابی کنید. نتایج را تفسیر کنید. در مورد اثربخشی آموزش خصمانه در بهبود استحکام مدل در برابر حملات خصمانه بحث کنید.

### پاسخ

در ابتدا می‌بایست دیتاست CIFAR10 را دانلود نموده. این دیتاست را دانلود کردیم و ۵ تصویر از ۱۰ کلاس موجود در این دیتاست را به صورت زیر نمایش دادیم:

Perturbed<sup>۵</sup>  
Target Attack<sup>۶</sup>

پاسخ



شکل ۲: تصاویری رندوم از دیتاست CIFAR10

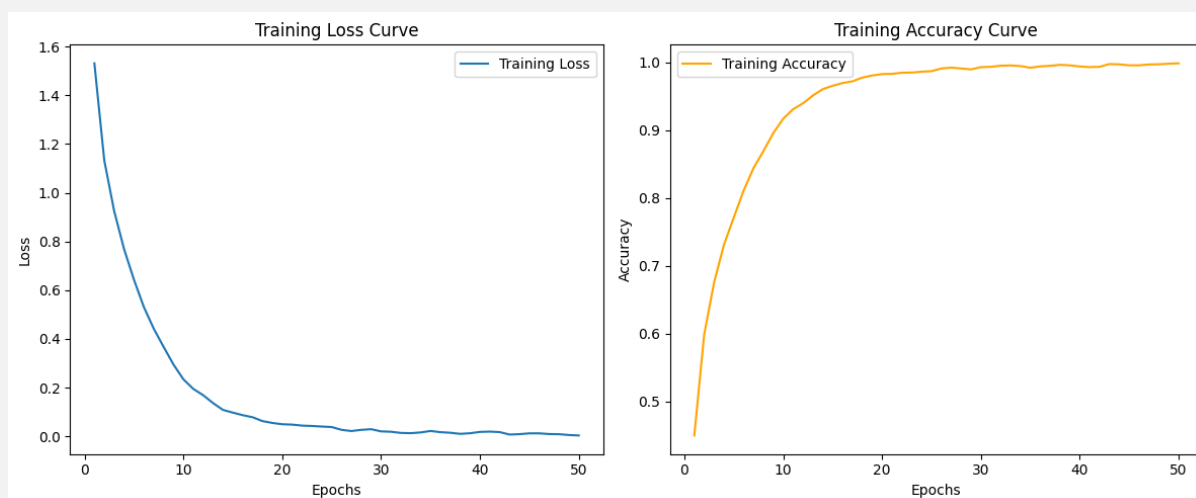
## پاسخ

سپس وزن‌های شبکه از پیش آموزش داده شده ResNet18 را لود می‌کنیم. چون شبکه ResNet بر روی دیتاست ImageNet آموزش دیده است، نیاز است که حتماً آن را در چند Epoch محدود با دیتاست خودمان آموزش مجدد بدهیم (Fine-tune کنیم). این کار را در ۵۰ اپیک انجام داده‌ایم و دقت و خطای آموزش شبکه به صورت زیر شده است:

```
Epoch 1/50: Training Loss: 1.5307 Accuracy: 0.4501
Epoch 2/50: Training Loss: 1.1327 Accuracy: 0.5990
Epoch 3/50: Training Loss: 0.9250 Accuracy: 0.6755
Epoch 4/50: Training Loss: 0.7690 Accuracy: 0.7313
Epoch 5/50: Training Loss: 0.6433 Accuracy: 0.7716
Epoch 6/50: Training Loss: 0.5311 Accuracy: 0.8113
Epoch 7/50: Training Loss: 0.4423 Accuracy: 0.8444
Epoch 8/50: Training Loss: 0.3675 Accuracy: 0.8693
Epoch 9/50: Training Loss: 0.2951 Accuracy: 0.8959
Epoch 10/50: Training Loss: 0.2346 Accuracy: 0.9170
Epoch 11/50: Training Loss: 0.1952 Accuracy: 0.9309
Epoch 12/50: Training Loss: 0.1694 Accuracy: 0.9397
Epoch 13/50: Training Loss: 0.1370 Accuracy: 0.9512
Epoch 14/50: Training Loss: 0.1090 Accuracy: 0.9605
Epoch 15/50: Training Loss: 0.0978 Accuracy: 0.9655
Epoch 16/50: Training Loss: 0.0873 Accuracy: 0.9695
Epoch 17/50: Training Loss: 0.0788 Accuracy: 0.9720
Epoch 18/50: Training Loss: 0.0629 Accuracy: 0.9776
Epoch 19/50: Training Loss: 0.0554 Accuracy: 0.9806
Epoch 20/50: Training Loss: 0.0500 Accuracy: 0.9827
Epoch 21/50: Training Loss: 0.0485 Accuracy: 0.9830
Epoch 22/50: Training Loss: 0.0445 Accuracy: 0.9847
Epoch 23/50: Training Loss: 0.0429 Accuracy: 0.9849
Epoch 24/50: Training Loss: 0.0406 Accuracy: 0.9862
Epoch 25/50: Training Loss: 0.0386 Accuracy: 0.9870
...
Epoch 48/50: Training Loss: 0.0092 Accuracy: 0.9971
Epoch 49/50: Training Loss: 0.0058 Accuracy: 0.9980
Epoch 50/50: Training Loss: 0.0040 Accuracy: 0.9985
Model weights saved to model_weights.pth
```

شکل ۳: روند تغییر دقت و خطای آموزش

همچنین نمودار خطا و دقت آموزش به صورت زیر بدست آمده است:



شکل ۴: نمودار خطا و دقت آموزش

## پاسخ

همچنین دقت شبکه برای داده‌های test نیز به صورت زیر به دست آمده است:

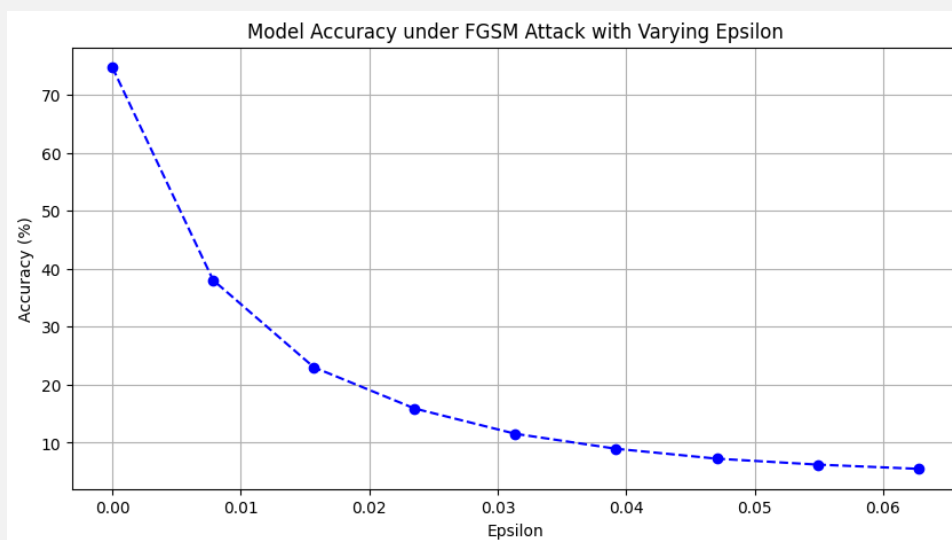
Test Accuracy : 74.7%

در مرحله بعد، حمله خصمانه FGSM را با استفاده از کتابخانه torchattacks به ازای مقدار  $\epsilon$  های متفاوت انجام می‌دهیم و مقدار دقت شبکه را به ازای هر  $\epsilon$  نمایش می‌دهیم:

```
Accuracy of the model on FGSM attack with epsilon = 0/255: 74.70%
Accuracy of the model on FGSM attack with epsilon = 2/255: 38.03%
Accuracy of the model on FGSM attack with epsilon = 4/255: 23.01%
Accuracy of the model on FGSM attack with epsilon = 6/255: 15.90%
Accuracy of the model on FGSM attack with epsilon = 8/255: 11.54%
Accuracy of the model on FGSM attack with epsilon = 10/255: 8.96%
Accuracy of the model on FGSM attack with epsilon = 12/255: 7.25%
Accuracy of the model on FGSM attack with epsilon = 14/255: 6.23%
Accuracy of the model on FGSM attack with epsilon = 16/255: 5.50%
```

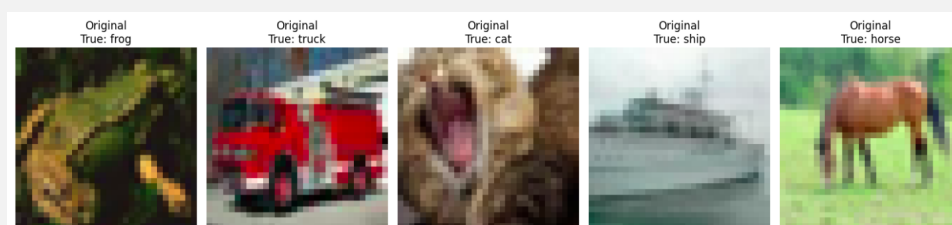
شکل ۵: روند تغییر دقت مدل به ازای  $\epsilon$  های متفاوت

همچنین نمودار این تغییرات نیز به صورت زیر شده است:



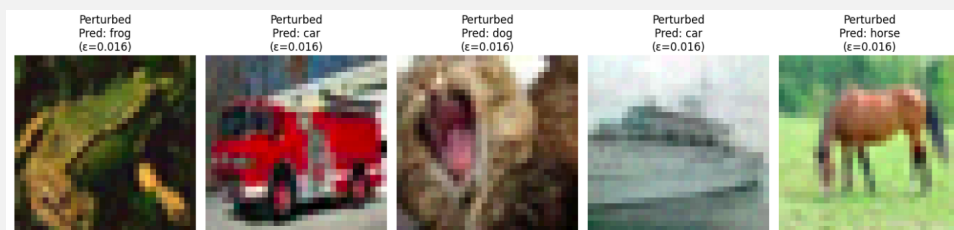
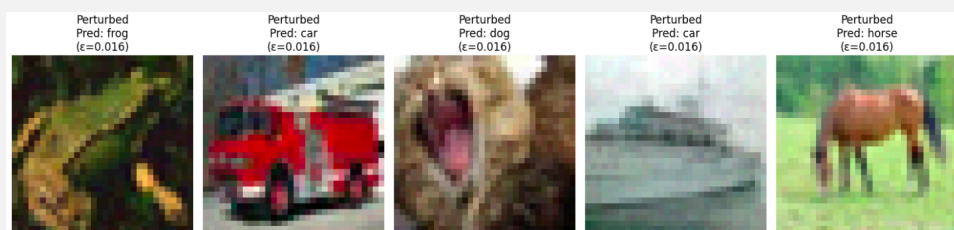
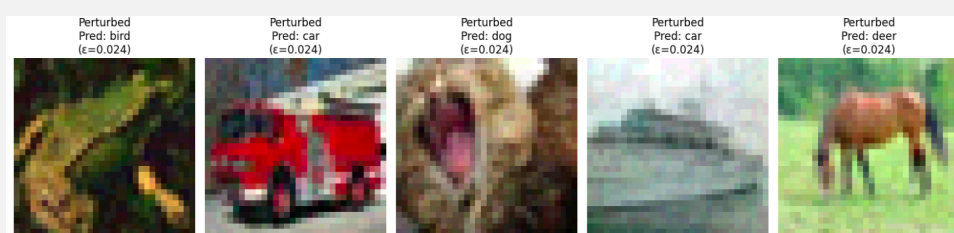
شکل ۶: نمودار تغییرات دقت مدل به ازای  $\epsilon$  های متفاوت

و سپس شبکه را تست می‌کنیم. به صورت رندوم تعدادی از تصاویر را انتخاب می‌کنیم. تصاویر انتخاب شده و کلاس آنها به صورت زیر است:



شکل ۷: تصاویر رندوم انتخاب شده

پاسخ

خروجی شبکه متخاصم به ازای  $\epsilon = 2/255$ شکل ۸: خروجی شبکه متخاصم به ازای  $\epsilon = 2/255$ خروجی شبکه متخاصم به ازای  $\epsilon = 4/255$ شکل ۹: خروجی شبکه متخاصم به ازای  $\epsilon = 4/255$ خروجی شبکه متخاصم به ازای  $\epsilon = 6/255$ شکل ۱۰: خروجی شبکه متخاصم به ازای  $\epsilon = 6/255$ 

مشاهده می‌شود که شبکه در تشخیص به اشتباه افتاده از و تعدادی از نمونه‌ها را اشتباه تشخیص داده است. برای حل این موضوع، شبکه را مجدداً آموزش می‌دهیم. اما این بار آموزش خصمانه. شبکه را در ۱۰۰ دوره آموزش می‌دهیم. روند تغییرات خطا و دقت به صورت زیر به دست آمده است:

پاسخ

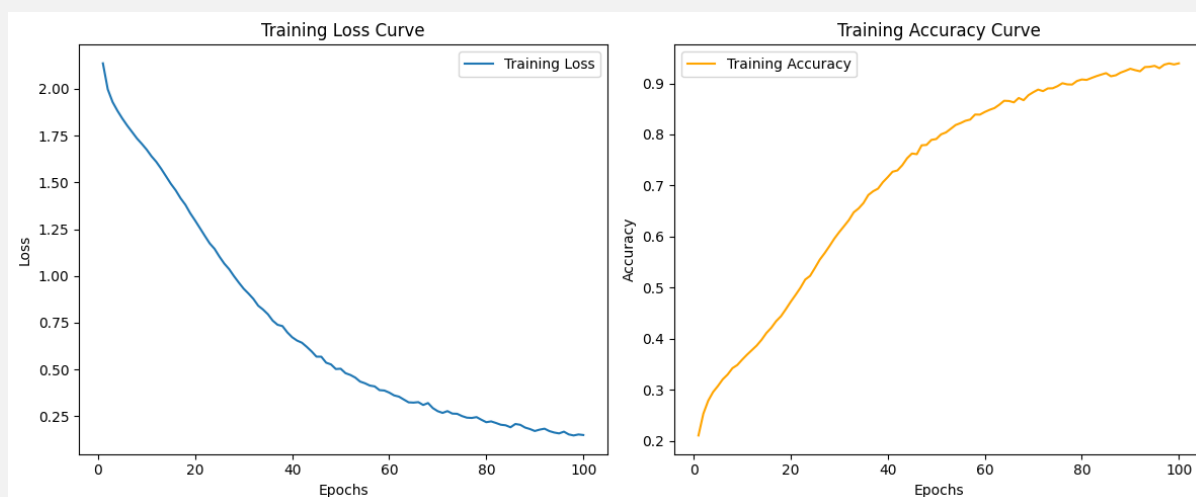
```

Epoch 1/100: Loss: 2.1359 Accuracy: 0.2107
Epoch 2/100: Loss: 1.9971 Accuracy: 0.2532
Epoch 3/100: Loss: 1.9290 Accuracy: 0.2787
Epoch 4/100: Loss: 1.8823 Accuracy: 0.2956
Epoch 5/100: Loss: 1.8411 Accuracy: 0.3075
Epoch 6/100: Loss: 1.8036 Accuracy: 0.3206
Epoch 7/100: Loss: 1.7694 Accuracy: 0.3299
Epoch 8/100: Loss: 1.7350 Accuracy: 0.3424
Epoch 9/100: Loss: 1.7062 Accuracy: 0.3486
Epoch 10/100: Loss: 1.6755 Accuracy: 0.3590
Epoch 11/100: Loss: 1.6396 Accuracy: 0.3687
Epoch 12/100: Loss: 1.6095 Accuracy: 0.3777
Epoch 13/100: Loss: 1.5727 Accuracy: 0.3867
Epoch 14/100: Loss: 1.5325 Accuracy: 0.3979
Epoch 15/100: Loss: 1.4923 Accuracy: 0.4113
Epoch 16/100: Loss: 1.4573 Accuracy: 0.4215
Epoch 17/100: Loss: 1.4152 Accuracy: 0.4344
Epoch 18/100: Loss: 1.3801 Accuracy: 0.4441
Epoch 19/100: Loss: 1.3349 Accuracy: 0.4578
Epoch 20/100: Loss: 1.2961 Accuracy: 0.4723
Epoch 21/100: Loss: 1.2553 Accuracy: 0.4857
Epoch 22/100: Loss: 1.2141 Accuracy: 0.4997
Epoch 23/100: Loss: 1.1741 Accuracy: 0.5159
Epoch 24/100: Loss: 1.1449 Accuracy: 0.5231
Epoch 25/100: Loss: 1.1035 Accuracy: 0.5386
...
Epoch 98/100: Loss: 0.1476 Accuracy: 0.9391
Epoch 99/100: Loss: 0.1530 Accuracy: 0.9371
Epoch 100/100: Loss: 0.1500 Accuracy: 0.9392
Model weights saved to adv_model_weights.pth

```

شکل ۱۱: روند تغییرات خطا و دقت شبکه در آموزش خصمانه

همچنین نمودار خروجی نیز به صورت زیر است:



شکل ۱۲: نمودار خطا و دقت شبکه در آموزش خصمانه

مجددا شبکه را به ازای مقادیر قبلی  $\epsilon$  تست می‌کنیم. مقادیر بدست آمده به صورت زیر است:



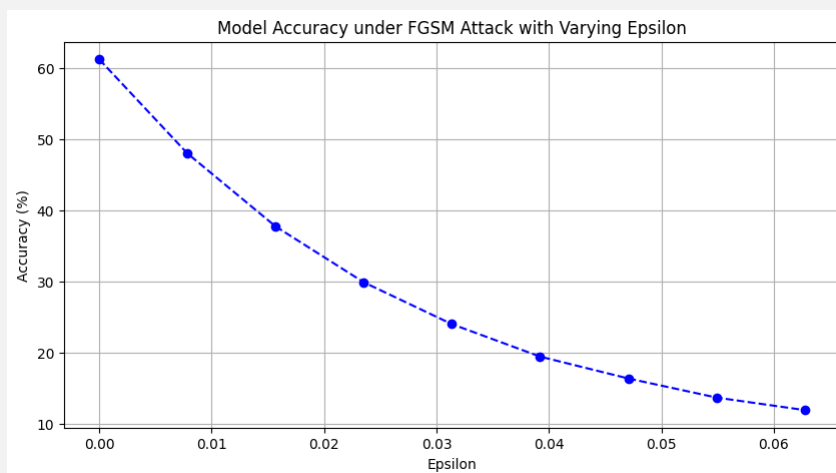
```

Accuracy of the adv_model on FGSM attack with epsilon = 0/255: 61.24%
Accuracy of the adv_model on FGSM attack with epsilon = 2/255: 48.02%
Accuracy of the adv_model on FGSM attack with epsilon = 4/255: 37.77%
Accuracy of the adv_model on FGSM attack with epsilon = 6/255: 29.92%
Accuracy of the adv_model on FGSM attack with epsilon = 8/255: 24.04%
Accuracy of the adv_model on FGSM attack with epsilon = 10/255: 19.48%
Accuracy of the adv_model on FGSM attack with epsilon = 12/255: 16.41%
Accuracy of the adv_model on FGSM attack with epsilon = 14/255: 13.73%
Accuracy of the adv_model on FGSM attack with epsilon = 16/255: 11.97%

```

شکل ۱۳: تست نمونه‌ها با شبکه آموزش داده شده جدید به ازای  $\epsilon$  های متفاوت

نمودار تغییرات دقت به ازای  $\epsilon$  های متفاوت نیز به صورت زیر به دست می‌آید:



شکل ۱۴: نمودار تغییرات دقت شبکه به ازای  $\epsilon$  های متفاوت

با مقایسه تغییرات در دو مرحله متوجه می‌شویم که با انجام آموزش خصمانه دقت شبکه در به دست آوردن خروجی‌های دستکاری شده افزایش یافته است. برای مثال به ازای  $\epsilon = 8/255$ ، ۱۲/۵ درصد افزایش دقت داشته ایم.

## سوال هفتم - عملی

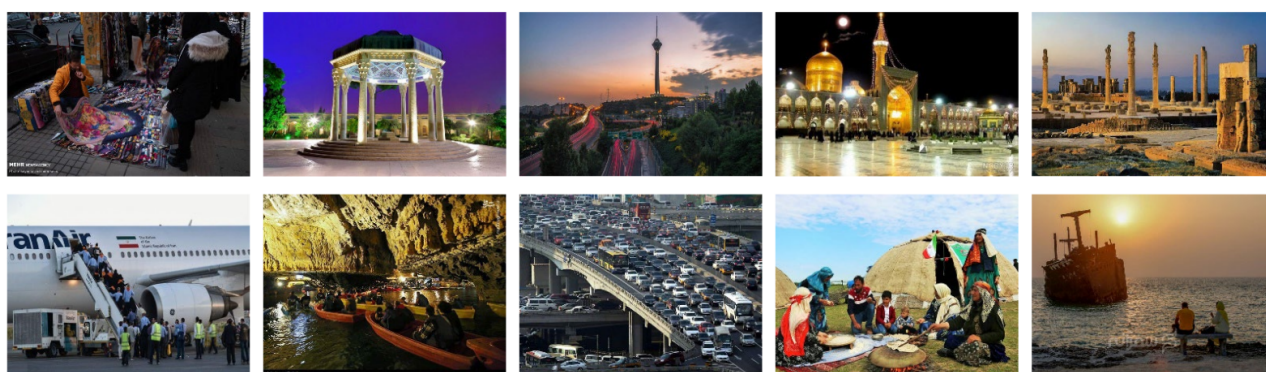
در این سوال تصمیم داریم تا برای تصاویر ایرانی یک مدل با معماری رمزگذار و رمزگشا برای وظیفه شرح تصویر<sup>۷</sup> طراحی کنیم. مجموعه داده persian\_image\_captioning.rar در اختیار شما قرار گرفته است. این مجموعه داده حدود ۱۵۰۰ مقاله خبری به همراه تصاویر مرتبط آن است. این مقالات از سایت خبرگزاری تسنیم جمع‌آوری شده است. فایل news.json حاوی لیستی از اشیاء json که هرکدام دارای اطلاعات زیر هستند:

۱. title: عنوان مقاله خبری
۲. Description: شرح کوتاهی از مقاله
۳. Category: دسته‌ای که مقاله به آن تعلق دارد
۴. Reporter: نام خبرنگاری که این مطلب را منتشر کرده است
۵. Time: تاریخ و ساعتی که مقاله در آن منتشر شده است
۶. Images: لیستی از تصاویر مرتبط با مقاله (همه آنها را می‌توانید در پوشه images پیدا کنید)

عنوان هر مقاله را می‌توان به عنوان یک شرح (caption) برای تصاویر مرتبط با آن مقاله، در نظر گرفت. همچنین می‌توانید با جایگزین کردن مترادف کلمات و همچنین، با روش‌های دلخواه برای تصاویر، داده افزایی<sup>۸</sup> کنید. در نهایت مدلی آموزش دهید تا این وظیفه را انجام دهد. موارد زیر را در گزارش خود لحاظ و توضیح کامل دهید:

۱. پیش‌پردازشی که انجام داده‌اید.
۲. معماری مدل پیشنهادی خود را رسم کنید.
۳. تابع هزینه‌ای<sup>۹</sup> که استفاده کردید.
۴. روش‌هایی که برای ارزیابی این وظیفه استفاده شده.

اسکرپیتی بنویسید تا با دریافت مسیر یک پوشه، شرح تصاویر در آن پوشه را در یک فایل txt بنویسد. پوشه تحت عنوان selected\_images در اختیار شما قرار گرفته است. مسیر این پوشه را به اسکرپیت خود بدهید و خروجی آن را (شرح تصاویر) همراه با تصاویر مرتبط ارسال کنید. دقت کنید که اسکرپیت نوشته شده توسط شما در روز تحویل پروژه توسط دیگر تصاویر بررسی خواهد شد. تصاویر این پوشه در زیر نشان داده شده است:



شکل ۱۵: تصاویر پوشه selected\_images

Image Captioning<sup>۷</sup>  
Data Augmentation<sup>۸</sup>  
Function Loss<sup>۹</sup>

توجه فرمایید نمره این تمرین  $(30 + 30)$  امتیازی است. یعنی در صورتی که مراحل پیش‌پردازش، معماری مدل، صحت نهایی و به طور کلی روش حل مسئله، دارای خلاقیت و کیفیت مورد قبولی باشد، علاوه بر نمره اصلی تا ۳۰ امتیاز، نمره اضافی برای شما در نظر گرفته خواهد شد.

پاسخ