به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین پنجم

پرسش ۱	نام و نام خانوادگی	بابک حسینی محتشم
	شماره دانشجویی	۸۱۰۱۰۱۴۰۸
پرسش ۲	نام و نام خانوادگی	بابک حسینی محتشم
	شماره دانشجویی	۸۱۰۱۰۱۴۰۸
	مهلت ارسال پاسخ	۱۴۰۴.۰۳.۱۸

فهرست

پرسش ۲ – Robust Zero-Shot Classification ۲

۲-۱. مقدمه ۲

۲-۱-۱. مفهوم **robustness** ۲

۲-۱-۲. روش‌های تولید نمونه‌های خصمانه ۳

۲-۱-۳. بررسی روش **Contrastive Learning** ۵

۲-۱-۴. بررسی مدل **CLIP** ۶

۲-۱-۵. بررسی طبقه‌بندی تک‌ضرب ۸

۲-۱-۶. بررسی انواع حمله خصمانه از نظر دسترسی به مدل ۹

۲-۱-۷. بررسی حملات انتقالی ۱۰

۲-۱-۸. بررسی روش تنظیم دقیق **LoRA** ۱۱

۲-۱-۹. بررسی اثر تابع خطای **CLIP** روی مقاوت ۱۳

۲-۲. آماده‌سازی برای آموزش ۱۴

۲-۲-۱. آماده‌سازی دادگان ۱۴

۲-۲-۲. لود کردن مدل‌ها ۱۴

۲-۳. تحلیل نتایج ۱۵

۲-۳-۱. تحلیل نتایج روی تصاویر تمیز ۱۵

۲-۳-۲. تحلیل نتایج روی تصاویر خصمانه ۱۷

۲-۴. تنظیم دقیق و تحلیل نتایج روی تصاویر اعتبارسنجی خصمانه ۱۹

۲-۴-۱. تحلیل نتایج پس از آموزش با خطای **corssentropy** ۱۹

۲-۴-۲. تحلیل نتایج پس از آموزش با خطای **TeCoA** ۲۲

۲-۴-۳. تحلیل نتایج پس از افزایش دما ۲۴

۲-۴-۴. تحلیل نتایج پس از آموزش با خطای **TeCoA** به روش **VPT** ۲۶

۲-۵. تنظیم دقیق و تحلیل نتایج روی تصاویر ارزیابی ۲۸

۲-۶. جمع‌بندی ۳۱

شکل‌ها

شکل ۱. مکان هندسی نرم‌های مختلف - منبع از اسلایدهای درس هوش مصنوعی قابل اعتماد ۳
شکل ۲. نحوه کارکرد تابع خطای Triplet loss ۶
شکل ۳. معماری مدل CLIP ۷
شکل ۴. نحوه استفاده از CLIP به عنوان یک طبقه‌بند ۹
شکل ۵. روش کارکارد LoRA - منبع از اسلایدهای درس مدل‌های زبانی بزرگ ۱۲
شکل ۶. چند نمونه از تصاویر CIFAR-10 به همراه برچسب ۱۴
شکل ۷. ماتریس آشفتگی مدل CLIP روی تصاویر تمیز پیش از تنظیم دقیق ۱۶
شکل ۸. ماتریس آشفتگی مدل روی نمونه‌های خصمانه پیش از تنظیم دقیق ۱۷
شکل ۹. یک نمونه خصمانه موفق به همراه نویز تولید شده ۱۹
شکل ۱۰. نمودار تغییرات خطای مدل حین آموزش ۲۰
شکل ۱۱. ماتریس آشفتگی مدل روی تصاویر خصمانه پس از تنظیم دقیق با خطای crossentropy ۲۱
شکل ۱۲. ماتریس آشفتگی مدل آموزش دیده با تابع خطای TeCoA ۲۳
شکل ۱۳. ماتریس آشفتگی پس از افزایش دما با خطای TeCoA ۲۵
شکل ۱۴. نمونه‌ای از تصاویر پس از جمع شدن با لایه اضافه شده ۲۶
شکل ۱۵. ماتریس آشفتگی با خطای TeCoA با روش VPT ۲۷
شکل ۱۶. مقایسه عملکرد مدل‌ها روی تصاویر تست تمیز و خصمانه ۲۹

جدول‌ها

جدول1. معیارهای ارزیابی مدل روی تصاویر تمیز پیش از تنظیم دقیق ۱۶
جدول2. معیارهای ارزیابی مدل روی نمونه‌های خصمانه پیش از تنظیم دقیق ۱۸
جدول3. معیارهای ارزیابی مدل روی نمونه‌های خصمانه پس از آموزش با خطای crossentropy... ۲۱
جدول4. معیارهای ارزیابی مدل روی نمونه‌های خصمانه پس از آموزش با خطای TeCoA ۲۳
جدول5. مقادیر معیارها پس از افزایش دما و تنظیم دقیق با خطای TeCoA ۲۵
جدول6. مقادیر معیارها با آموزش با خطای TeCoA به روش VPT ۲۷
جدول7. مقادیر معیارها برای مدل روی داده تست خصمانه و تمیز ۳۰

پرسش ۲ – Robust Zero-Shot Classification

۲-۱. مقدمه

۲-۱-۱. مفهوم robustness

مدل‌های یادگیری ماشین می‌توانند بسیار شکننده باشند. یکی از بحث‌های بسیار مهم امروزه افزایش robustness مدل‌ها است تا بتوان از کارکرد مدل‌ها در زمینه‌های حیاتی و حساس مانند اتومبیل‌های خودران و پزشکی اطمینان حاصل کرد تا از خسارات احتمالی جلوگیری کرد. با بررسی نقطه ضعف‌های مدل‌ها در بحث robustness می‌توان چاره‌ای اندیشید و سعی کرد robustness مدل‌ها ره بهبود داد. پس هر دو مسئله ایجاد نمونه‌هایی که نشان دهنده ضعف مدل باشند و ارائه روش‌های مقابله با این نمونه‌ها می‌تواند اهمیت زیادی داشته باشد.

ابتدا مفهوم robustness را تعریف می‌کنیم. پایداری یا robustness مفهومی نزدیک به تعمیم‌پذیری یا generalization است که بیان می‌کند چقدر مدل در مقابل هر گونه تغییرات اندک خصمانه یا غیر خصمانه که در ورودی ایجاد شود، مقابل است. در خیلی مسائل با افزایش پایداری، عملکرد مدل روی داده‌های آموزش دیده کاهش میابد و در مقاله هم یکی از روش‌های جلوگیری از کاهش عملکرد را معرفی می‌کند.

انواع مختلف روش تولید نمونه‌های خصمانه از داده‌ها مخصوصا داده‌های تصویری وجود دارد. به طور کلی ایده کلی نمونه‌های خصمانه بدین گونه است که فرض می‌کنیم نمونه X را داریم و می‌دانیم در صورتی که به مدل ورودی X را بدهیم به آن برچسب T می‌دهد. حال هدف آن است که مسئله بهینه‌سازی زیر را حل کنیم:

$$\min_{X'} d(X,X') + \lambda Loss\big(T, F(X')\big) \quad s.t. \quad T \neq F(X)$$

یعنی در رابطه بالا قصد داریم X’ را طوری پیدا کنیم که در عین حال که با وجود تابع فاصله دلخواه به نام d، تغییرات X نسبت به X’ طبق d کم است، خطای F(X’) هم نسبت به T کم باشد. معمولا در طبقه‌بندی از تابع خطای crossentropy استفاده می‌شود. تابع F هم نشان دهنده مدلی مورد حمله است. مقدار λ تنظیم کننده میزان اهمیت نسبت به ایجاد تغییرات اندک یا خطای کمتر بین F(X’) و T است. البته شایان ذکر است که به این نوع حمله، حمله هدف گذاری شده می‌گویند در حالی که نوع دیگری حمله داریم که بدون هدف است و در آن تنها هدف آن است که F(X’) به مقداری به جز F(X) برسد ولی اینکه به چه مقداری برسد اهمیتی ندارد. یعنی در حمله بدون هدف، تابع بهینه‌سازی چنین است:

$$\min_{X'} d(X,X') - \lambda Loss\big(Q, F(X')\big) \quad s.t. \quad F(X) = Q$$

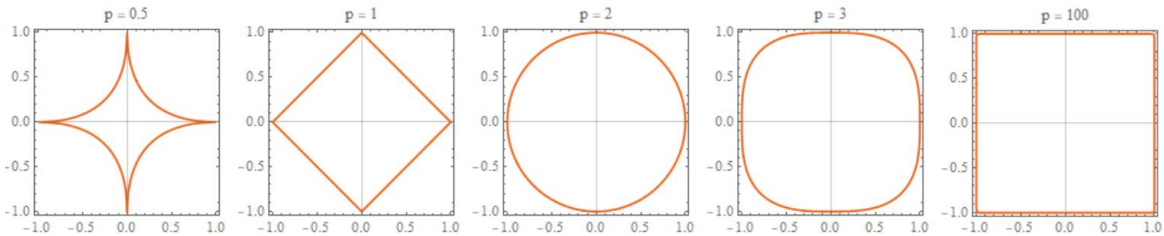
چون تابع F غیرخطی است، حل مسائل بهینه‌سازی بالا نسبتا سخت می‌شود ولی در طی سالیان روش‌های متعددی برای تولید نمونه‌های خصمانه بررسی شده است و معمولا به ازای هر نوع مکانیزم دفاعی در مقابل حملا خصمانه، حملات دیگری ارائه می‌شود و با عبور از آن مکانیزم‌ها باعث ایجاد خطا در مدل می‌شوند. البته مکانیزم‌های دفاعی نیز وجود دارند که از طریق ریاضیاتی ثابت شده‌اند تحت شرایط خاصی هیچ نوع حمله‌ای نمی‌تواند از آن‌ها عبور کند ولی به دلیل نیاز به شرایط خاص و همچنین افت کیفیت در صورت برقرار بودن آن شرایط، باز هم در این حوزه نیاز به مکانیزم‌های دفاعی مناسب وجود دارد.

۲-۱-۲. روش‌های تولید نمونه‌های خصمانه

پیش از توضیح در مورد روش‌های تولید نمونه‌های خصمانه، درمورد معیارهای همسایگی توضیح می‌دهیم. از این معیارها معمولا با هدف ایجاد کمینه تغییرات در ورودی استفاده می‌کنیم. به طور کلی p-norm بردار x با n مولفه بدین صورت تعریف می‌شود:

$$\ell_p norm := ||x||_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$$

بدین ترتیب چند نرم رایج را توضیح می‌دهیم. تغییرات یک نمونه نسبت به حالت اول یعنی $||x - x'||_p =$ ε را در نظر می‌گیریم. نرم صفر یا ℓ_0 بیانگر تعداد مولفه‌هایی از x است که تغییر می‌دهیم و چون از جنس اسکالر هست و نمی‌توان از آن مشتق گرفت، معمولا از آن استفاده نمی‌کنیم. نرم یک یا ℓ_1 در یک فضای دو بعدی، برابر یک لوزی به مرکز x و قطرهای با اندازه 2ε است. نرم دو در فضا دو بعدی نشان‌دهنده دایره به مرکز x و شعاع ε است و نهایتا یکی از رایج‌ترین نرم‌ها، نرم ∞ است که بیان‌گر بیشینه مولفه‌های x است که در فضای دو بعدی مربعی به ضلع 2ε شود. منظور از این اشکال هندسی آن است که در صورتی که می‌خواهیم x' خصمانه که با تغییر در x بوجود می‌آید طبق تابع فاصله p-norm فاصله‌اش تا x کمتر از ε باشد، نیاز است تا در مکان هندسی بیان شده قرار گیرد.



شکل ۱. مکان هندسی نرم‌های مختلف - منبع از اسلایدهای درس هوش مصنوعی قابل اعتماد

حال دو روش رایج برای تولید نمونه‌های خصمانه که در مقاله هم از آن‌ها استفاده شده است را بیان می‌کنیم که FGSM و PGD نام دارند. ابتدا به روش FGSM یا Fast Gradient Sign Method می‌پردازیم و سپس روش PGD یا Projected Gradient Descent را بیان می‌کنیم.

در روش FGSM برای تولید x’، معمولا از تابع همسایگی L بی‌نهایت استفاده می‌کنیم و قصد داریم فاصله x’ با x حداکثر برابر اپسیلون باشد. برای تولید x’ برای حمله بی‌هدف مطابق رابطه زیر عمل می‌کنیم:

$$x' = x + \varepsilon\, sign[\nabla J(\theta, x, y)]$$

که در رابطه بالا y برابر برچسبی است که مدل به ورودی x می‌دهد، θ نشان دهنده پارامترهای مدل است و J هم تابع خطا را نشان می‌دهد. بدین ترتیب مشابه الگوریتم gradient ascent، پس از پیدا کردن مشتق خطای مدل نسبت به ورودی x و در نظر گرفتن جهت آن با sign، با مقدار اپسیلون در جهت حساب شده حرکت می‌کنیم تا x’ را تشکیل دهیم. البته به طرز مشابه می‌توان از این روش برای تولید نمونه با هدف برچسب y’ استفاده کرد:

$$x' = x - \epsilon\, sign[\nabla J(\theta, x, y')]$$

بدین ترتیب بردار x’ ساخته شده بدین شکل است:

$$x = \begin{bmatrix} x_1 \\ \dots \\ x_d \end{bmatrix} \rightarrow x' = \begin{bmatrix} x_1 \pm \epsilon \\ \dots \\ x_1 \pm \epsilon \end{bmatrix}$$

که مثبت یا منفی بودن علامت‌های بالا از جهت مشتق خطا به دست می‌آید.

روش رایج بعدی PGD است که یک روش iterative است. در این روش در تکرار صفر، مقدار x’ برابر نقطه‌ای در همسایگی x در نظر گرفته می‌شود و در هر مرحله با ایجاد تغییراتی اندک، خطای مدل را نسبت برچسب اولیه افزایش می‌دهیم. طبق این روش x’ در مرحله t+1 بدین صورت به دست می‌آید:

$$x'_{t+1} = Proj\{x'_t + \alpha\, sign[\nabla_x J(\theta, x'_t, y)]\}$$

مشابه قبل با اجرای gradient ascent سعی می‌کنیم خطای مدل را روی نمونه x’ در تکرار قبل برای برچسب y افزایش دهیم ولی این بار بردار جهت مشتق ضربدر α می‌شود در صورتی که در FGSM ضربدر ε می‌شد به طوریکه α عدد دلخواهی است و مانند نرخ یادگیری در مدل‌های عمیق، نرخ تغییرات در هر تکرار را نشان می‌دهد. پس از جمع شدن مقدار قبلی x’ با بردار مشتق، بردار x’ حاصل ممکن است خارج از محدوده تعیین شده برود. برای همین در پس از جمع در هر تکرار، بردار حاصل را به فضای مورد نظر که در L بی‌نهایت مربع به ضلع 2ε و مرکز x است، project می‌کنیم که خود این کار به طور کلی حل یک مسئله بهینه‌سازی پیدا کردن نزدیک‌ترین نقطه در فضای مورد نظر به بردار جدید است که می‌تواند سخت باشد ولی در L بی‌نهایت، معادل clip کردن می‌شود و کافی است در هر جهتی که فراتر است مقدار بردار را ببُریم تا به نقطه‌ای روی ضلع مربع برسیم.

۲-۱-۳. بررسی روش Contrastive Learning

۲. حال به بررسی مدل مورد استفاده در مقاله به نام CLIP می‌پردازیم. مدل CLIP یکی از بهترین مدل‌های زمینه ترکیب زبان و بینایی از لحاظ عملکرد است. این مدل کاربردهای مختلفی در زمینه‌های مختلف مانند تولید تصویر از متن و جستجوی تصویر از متن یا متن از تصویر دارد. در این مدل با روش contrastive آموزش صورت می‌گیرد که در ادامه در مورد آن توضیح می‌دهیم.

از روش آموزش contrastive در آموزش مدل‌های زبانی بزرگ و بینایی ماشین برای استخراج نمایشی معنادار از داده‌های بدون برچسب استفاده می‌شود. در این روش با فرض اینکه نمونه‌های مشابه در فضای امبدینگ باید فاصله کمتری نبست به نمونه‌های غیر مشابه داشته باشند کار می‌کند. دو نوع مختلف از آموزش contrastive در زمینه یادگیری با نظارت و یادگیری خود نظارتی مطرح شده است.

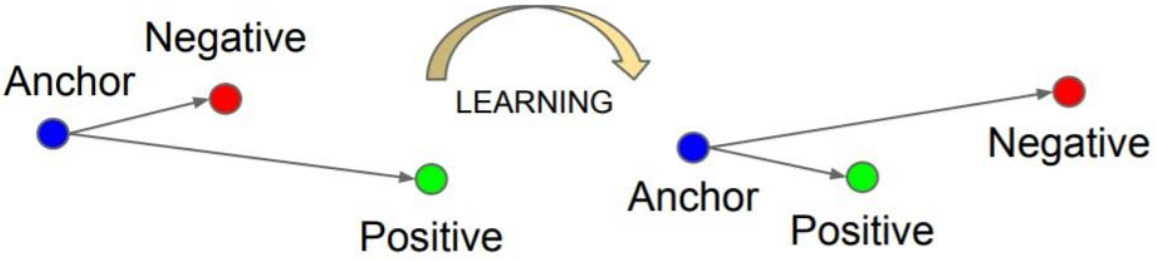
در آموزش contrastive در یادگیری با نظارت، مدل با کمک برچسب‌ها یاد می‌گیرد که نمونه‌های مشابه و غیر مشابه را از هم تشخیص دهد. در این روش مدل یاد روی زوج‌های داده و برچسب آموزش داده می‌شود تا فضای امبدینگی را بیاموزد که نمونه‌های مشابه فاصله کمتری نسبت به نمونه‌های غیر مشابه دارند. تابع رایج برای این کار infoNCE یا Information Noice Contrastive Estimation است که با کمک مفهوم mutual information بین داده‌ها کار می‌کند و رابطه آن بدین صورت است:

$$Loss = -InfoNCE = -\frac{1}{N}\sum_{i=1}^{N} log \frac{f(v_i, v_i')}{\frac{1}{N}\sum_j \exp f(v_i, v_j')}$$

که در رابطه بالا f تابعی است که شباهت دو نمونه را تخمین می‌زند و در مدل CLIP و بسیاری مدل‌های دیگر از cosine similarity یا ضرب نقطه‌ای استفاده می‌شود. همچنین v_i' نمونه‌ای مشابه v_i را نشان می‌دهد. بدین ترتیب با کمینه کردن خطا، مدل سعی می‌کند صورت کسر یعنی شباهت بین دو نمونه مشابه را بیشینه کند در حالیکه شباهت بین بقیه نمونه‌ها که در مخرج هستند را کاهش دهد.

به دلیل هزینه زیادی که ایجاد تعداد زیادی داده برچسب‌دار دارد، در روش‌های خود نظارتی مدل‌ها با داشتن شمار زیادی داده بدون برچسب می‌توانند یاد بگیرند. در آموزش contrastive برای ایجاد نمونه‌های مثبت و منفی از خود داده‌ها یکی از روش‌ها اعمال تبدیلات مختلف برای ایجاد داده‌های افزونه یا همان augmented است. برای مثال در بینایی ماشین با بریدن بخشی از تصویر یا بی‌رنگ کردن تصویر و یا اعمال تبدیلات هندسی مختلف نمونه‌های مثبت زیادی برای تصویر اصلی می‌توان ایجاد کرد و از بخش تصادفی از تصاویر دیگر می‌توان برای ایجاد نمونه‌های منفی استفاده کرد. اگر تعداد تصاویر بسیار زیاد باشد، احتمال اینکه دو تصویر تصادفی به هم شبیه باشند کاهش میابد و بدین ترتیب با خطای معرفی شده در بالا یا توابع دیگر می‌توان تصاویر مشابه را تفکیک کرد.

توابع خطای دیگری در آموزش contrastive وجود دارند. یکی دیگر از توابع خطا Triplet loss است که در این تابع خطا، نمونه‌ای به عنوان مرجع انتخاب می‌شود و با انتخاب دو نمونه دیگر یک مشابه و یکی متفاوت با نمونه مرجع، سعی می‌شود فاصله نمونه مثبت کمتر از فاصله نمونه منفی تا مرجع باشد.



شکل ۲. نخوه کارکرد تابع خطای **Triplet loss**

با تغییر این تابع خطای می‌توان به N-pair loss رسید که به ازای هر لنگر چندین نمونه مثبت و منفی را در نظر می‌گیرد و سعی می‌کند شباهت بین نمونه‌های مثبت با مرجع را افزایش و شباهت نمونه‌های منفی با نمونه مرجع را کاهش دهد.

۲-۱-۴. بررسی مدل CLIP

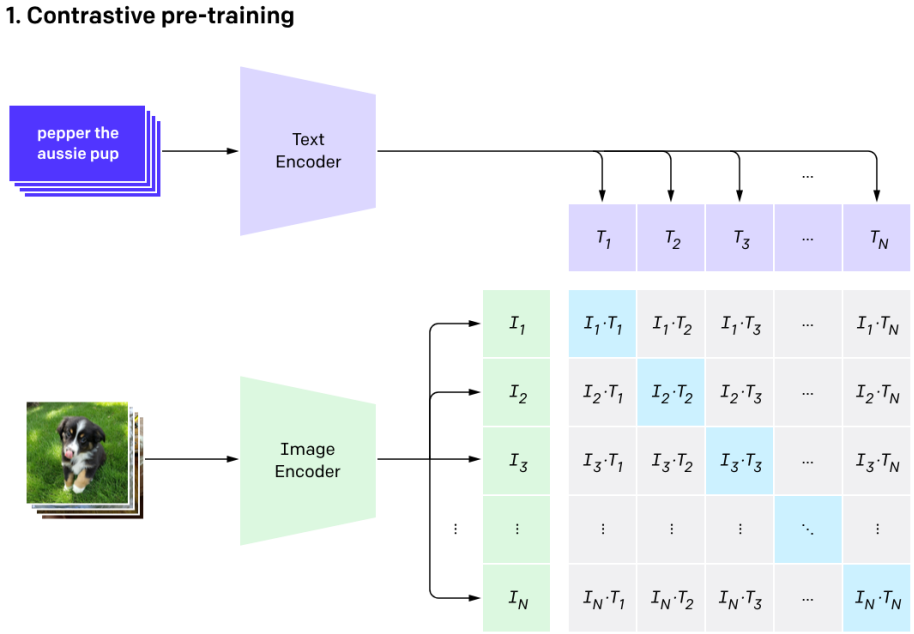
مدل‌های طبقه‌بند مخصوص دیتاست خاص به خوبی روی آن کار می‌کنند ولی توانی تشخیص کلاس‌هایی که در داده‌های خود ندیده‌اند را ندارند. مدل CLIP با آموزش contrastive روی صدها میلیون تصویر به همراه متونی که به همراه تصویر در اینترنت موجود است نه تنها روی کلاس‌هایی که دیده عملکردی مشابه مدل‌های آموزش دیده روی آن داده‌ها دارد، بلکه می‌تواند کلاس‌هایی که در داده‌های خود ندیده است را نیز تشخیص دهد.

برای آموزش مدل CLIP روی صدها میلیون داده هزینه محاسباتی بسیار زیادی نیاز است. مدل CLIP از دو روش الگوریتمی برای کاهش هزینه‌های محاسباتی استفاده می‌کند. روش اول استفاده از Vision Transformerها است که به دلیل استفاده از معماری ترنسفورمر، می‌توانند از قابلیت موازی‌سازی GPUها استفاده کنند و از لحاظ هزینه محاسباتی حدود ۳ برابر مدل را بهتر کنند. الگوریتم دیگر برای کاهش محاسبات استفاده از آموزش contrastive است که بهینه‌سازی حدود ۴ تا ۱۰ برابر شده است و بدین ترتیب کل مدل را در عرض دو هفته مشابه مدل‌های زبانی بزرگ توانسته‌اند آموزش دهند.

به جز بهینه بودن از لحاظ محاسبات مدل CLIP تعمیم‌پذیری بسیاری دارد و همان طور که در ادامه توضیح خواهیم داد توانایی طبقه‌بندی تک‌ضرب را به خوبی کسب کرده است. همچنین طبق گزارشات

OpenAI که این مدل را معرفی کرده است، بهترین مدل CLIP در ۲۰ از ۲۶ دیتاست از مدل‌های ImageNet بهتر عمل کرده است.

با این حال این مدل در انجام برخی taskها مانند شمارش اشیاء در تصویر و تشخیص فاصله اشیاء ضعف دارد. همچنین در برخی taskها با جزئیات زیاد مانند تشخیص مدل ماشین یا گونه گیاه، از مدل‌های آموزش دیده روی این taskها ضعیف‌تر است.



شکل ۳. معماری مدل **CLIP**

حال به بررسی جزئی‌تر CLIP می‌پردازیم. این مدل از دو انکودر تشکیل شده است. یکی از آن‌ها متنی را به عنوان ورودی دریافت می‌کند و در نهایت برداری حاوی محتوای معنایی آن متن را خروجی می‌دهد. به طور مشابه مدل دیگری با دریافت تصویر به عنوان ورودی، آن را به بردار فضای امبدینگ می‌برد و طبق contrastive learning سعی می‌شود شباهت کسینوسی این دو بردار امبدینگ افزایش یابد و شباهت بقیه بردارهای موجود در بچ کاهش یابد. برای این کار از تابع خطای چند کلاسه N-pair که در بالا توضیح دادیم استفاده می‌شود:

$$-\frac{1}{N}\sum_{i}\ln\frac{e^{v_i\cdot w_i/T}}{\sum_j e^{v_i\cdot w_j/T}}-\frac{1}{N}\sum_{j}\ln\frac{e^{v_j\cdot w_j/T}}{\sum_i e^{v_i\cdot w_j/T}}$$

به طور کلی این تابع خطا سعی می‌کند ضرب نقطه‌ای بین بردارهای تصویر و متن متناظر را افزایش دهد $v_i \cdot w_i$ در حالیکه از ضرب نقطه‌ای زوج‌های متفاوت بکاهد. پارامتر T>0 دما نام دارد که در مدل CLIP برابر $e^{-\tau}$ در نظر گرفته شده است که τ یک پارامتر قابل یادگیری است.

۲-۱-۵. بررسی طبقه‌بندی تک‌ضرب

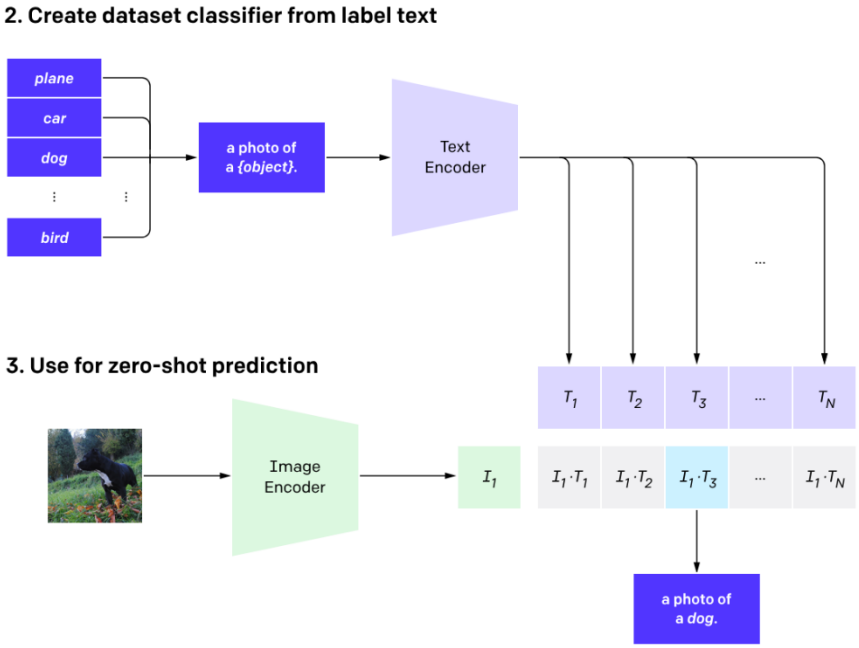
همان طور که بیان کردیم مدل CLIP می‌تواند کلاس‌هایی که در داده‌های آموزشی خود ندیده است را هم تشخیص دهد. به این نوع تعمیم‌پذیری Zero-Shot Generalization گفته می‌شود. به طور کلی در مدل‌ها برای ایجاد این نوع تعمیم‌پذیری معمولا از بردن ویژگی‌ها به فضای امبدینگ استفاده می‌شود یا با کمک مدل‌های مولد ویژگی‌های غیر واقعی از کلاس‌های آموزش ندیده تولید می‌شود تا طبقه‌بند را آموزش دهند.

این پدیده در مدل‌های زبانی هم وجود دارد که به مدل‌های توانایی انجام taskهایی را می‌دهد که مشابه‌شان را در متون آموزشی خود ندیده است. در حوزه بینایی ماشین CLIP جزو مدل‌هایی است که به خوبی این قابلیت را کسب کرده است و حتی می‌تواند OCR نیز انجام دهد.

در روش‌های طبقه‌بندی معمولی تصاویر مدل با دیدن تصاویر گوناگون از K کلاس الگوهای تصاویررا میاموزد و با کسب قابلیت تعمیم‌پذیری می‌تواند کلاس تصاویری که از همان K کلاس که ندیده است را نیز تشخیص دهد. در طبقه‌بندی تک‌ضرب مدل مانند طبقه‌بندی معمولی با دیدن تصاویر K کلاس مختلف طبقه‌بندی آن K کلاس را میاموزد ولی به جز آن می‌تواند تصاویر کلاس‌های دیگر به جز آن K کلاس را نیز به درستی طبقه‌بندی کند.

در CLIP چون به ازای هر تصویر متون مشابه آن تصویر را نیز یاد می‌گیرد و با توجه به قابلیت‌های تک‌ضرب ایجاد شده در مدل‌های زبانی با آموزش روی داده‌های فراوان، مدل CLIP می‌تواند با تکیه بر دانشی که از متون دریافت کرده است تصاویر کلاس‌های جدید را نیز تشخیص دهد.

برای طبقه‌بندی به کمک CLIP، تصویر دلخواهی را به قسمت انکودر تصویر آن می‌دهیم و پس از دریافت بردار امبدینگ تصویر، به ازای کلاس‌های مورد نظر متنی دلخواه تولید می‌کنیم که طبق خود مقاله می‌توان متن "a photo of a {object}" تولید کرد و با دادن این متن به قسمت انکودر متن، بردار امبدینگ متن را دریافت می‌کنیم. سپس شباهت کسینوسی بردار تصویر را با تمام بردارهای متنی حساب می‌کنیم و متنی که بردار متناظر آن بیشترین شباهت را به بردار تصویر داشت به عنوان کلاس تصویر در نظر می‌گیریم.



شکل ۴. نحوه استفاده از **CLIP** به عنوان یک طبقه‌بند

معمولا با افزایش پایداری مدل‌ها، عملکرد آن‌ها افت می‌کند ولی طبق مقاله بررسی‌های زیادی در زمینه افت کیفیت طبقه‌بندی تک‌ضرب با افزایش پایداری یا جلوگیری از افت کیفیت آن انجام نشده است.

۲-۱-۶. بررسی انواع حمله خصمانه از نظر دسترسی به مدل

از نظر دسترسی به مدل مورد هدف در زمینه robustness معمولا نوع حمله خصمانه را به سه نوع مختلف تقسیم می‌کنند.

- **حمله جعبه سفید**: در این نوع حمله، حمله کننده اطلاعات کاملی در مورد مدل دارد و هر موقع بخواهد می‌تواند از اطلاعات کامل نسبت به وزن‌ها، هایپرپارامترها، گردیان‌ها، معماری و اطلاعات دیگر در مورد مدل استفاده کند.
- **حمله جعبه خاکستری**: در این نوع حمله دسترسی به اطلاعات محدودتر است. برای مثال حمله کننده ممکن است تنها در مورد معماری مدل یا داده‌های آموزشی بداند ولی از جزئیات آموزشی و وزن‌های مدل اطلاعی ندارد.
- **حمله جعبه سیاه**: در این نوع حمله، حمله کننده هیچ دسترسی به وزن‌ها، گرادیان، معماری یا اطلاعات دیگر نسبت به مدل ندارد. حمله‌کننده تنها می‌تواند با ورودی دادن به مدل خروجی را از آن دریافت کند که خود خروجی می‌تواند به صورت احتمال هر کلاس یا برچسب پیش‌بینی شده باشد. هدف در این نوع حمله ایجاد نمونه خصمانه با دادن کمترین تعداد ورودی به مدل است.

حمله‌های جعبه خاکستری ویژگی‌هایی بین دو روش دیگر حمله دارند پس ما فقط آن دو را مقایسه می‌کنیم. ایجاد نمونه‌های خصمانه در حمله جعبه سفید نسبت به جمله جعبه سیاه احتمال موفقیت بیشتری دارد زیرا حمله‌کننده با داشتن اطلاعات گوناگون از مدل می‌تواند نمونه‌هایی مخصوص مدل تولید کند. به دلیل مشابه در حمله جعبه سفید می‌توان با سرعت بیشتری یک نمونه خصمانه تولید کنیم و نمونه تولید شده هم باعث افت عملکرد بیشتری در مدل می‌شود. البته حملات جعبه سیاه اهمیت فراوانی دارند زیرا در دنیای واقعی معمولا به اطلاعات مدل‌ها دسترسی نداریم و حمله جعبه خاکستری و جعبه سیاه می‌توانند بهتر نشان دهنده خطراتی باشند که در دنیای واقعی ممکن است رخ دهند.

در این مقاله فرض شده است که حمله‌کننده به اطلاعات کامل دسترسی دارد پس می‌تواند سخت‌ترین نمونه‌های خصمانه را برای مدل تولید کند و حمله از نوع جعبه سفید است. بدین ترتیب با استفاده از روش ارائه شده سعی کرده‌اند نشان دهند مدل در مقابل این حمله مقاوم می‌شود و به طور مشابه در مقابل حملات جعبه سیاه نیز مقاوم می‌شود. البته باید در نظر داشت که تمام روش‌های مقابله برای هر دو روش مناسب نیستند برای مثال برخی روش‌ها سعی می‌کنند اطلاعات مربوط به گرادیان را مخفی کنند که در حملات جعبه سیاه تفاوتی ایجاد نمی‌کند.

۲-۱-۷. بررسی حملات انتقالی

روش‌های حمله انتقالی یکی از دو دسته اصلی از حملات جعبه سیاه بشمار می‌رود. دسته دیگر حملات جعبه سیاه روش‌های مبتنی بر بهینه‌سازی هستند که گرادیان مدل مورد هدف را تخمین می‌زنند. با اینکه این دسته روش‌ها به اطلاعات بسیار کمتری نسبت به مدل مورد هدف دارند و از لحاظ پیاده‌سازی عملی‌تر هستند ولی نسبت به دسته دیگر احتمال موفقیت کمتری دارند. با این حال همان طورکه توضیح دادیم مانند بقیه جملات جعبه سیاه به دلیل اینکه به واقعیت نزدیک‌تر هستند چون حمله کننده اطلاعات زیادی در مورد مدل ندارد، بررسی آن‌ها اهمیت زیادی نسبت به حملات جعبه سفید دارد.

به طور کلی در این نوع حملات با دادن ورودی به مدل و دریافت خروجی از آن مدل دیگری آموزش داده می‌شود تا مدل اولی را تقریب بزند و مشابه آن عمل کند. سپس با داشتن اطلاعات کامل نسبت به مدل آموزش داده شده، نمونه‌های خصمانه‌ای با روش‌های جعبه سفید تولید می‌کنیم و به مدل مورد حمله می‌دهیم و نشان داده شده است که نمونه‌های خصمانه تولید شده برای یک مدل می‌توانند برای مدل‌های دیگر نیز نمونه خصمانه بشمار بروند و بدین ترتیب باعث افت کیفیت مدل دیگر می‌شویم. همچنین نشان داده شده حتی استفاده از نمونه‌های خصمانه یک مدل غیر دقیق از مدل مورد حمله، احتمال موفقیت غیر قابل چشم‌پوشی دارد.

۲-۱-۸. بررسی روش تنظیم دقیق LoRA

روش تنظیم دقیق LoRA یکی از روش‌های محبوب و پر استفاده در تنظیم دقیق مدل‌های عمیق مخصوصا مدل‌های زبانی بزرگ است. این روش جزو روش‌های Reparameterized Fine-tuning است که در این دسته روش‌ها از ماتریسی با درجه پایین استفاده می‌شود تا وزن‌های لایه‌های از قبل آموزش دیده مدل آپدیت شوند. بدین ترتیب توازنی بین آپدیت کردن وزن‌ها و نگه داشتن وزن‌های فعلی رعایت می‌شود و از لحاظ محاسباتی بهینه است.

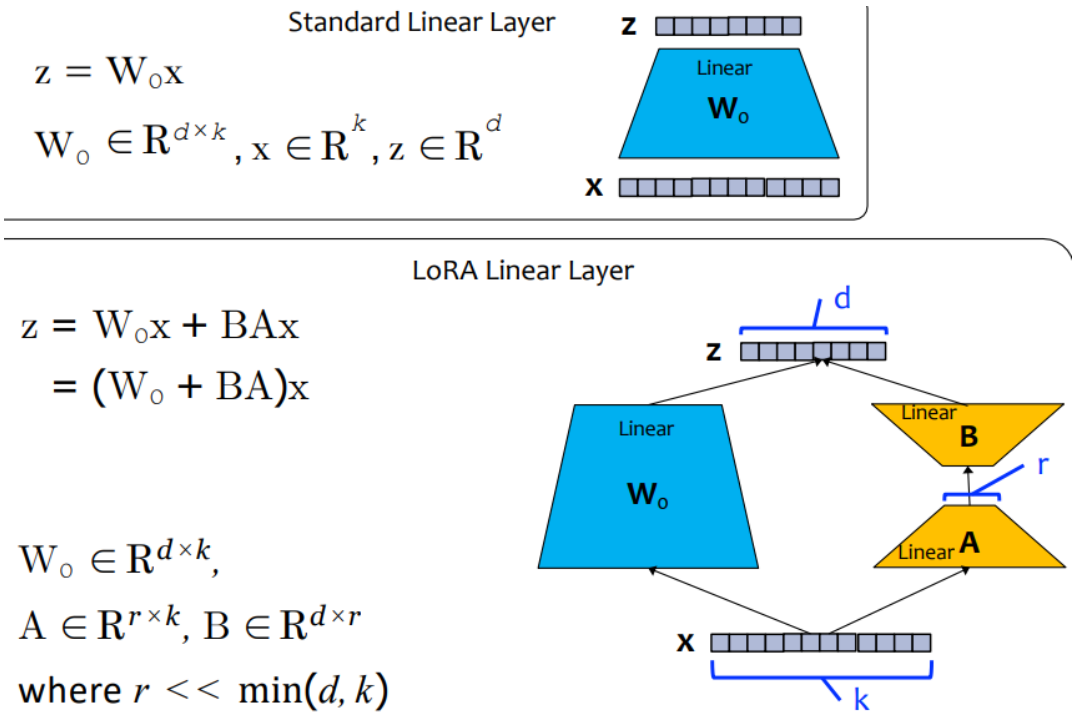
خود روش‌های Reparameterized Fine-tuning جزو روش‌های PEFT یا Parameter Efficient Fine Tuning محسوب می‌شوند. با بزرگ شدن مدل‌ها و افزایش تعداد پارامترها تنظیم دقیق آن‌ها امکان پذیر نمی‌باشد زیرا برای تنظیم دقیق مدل‌ها حدود سه برابر حجم خود مدل به حافظه نیاز داریم. برای همین روش‌های PEFT معرفی شدند تا با ایجاد تغییراتی در نسبت کمی از پارامترها بتوانیم مدل را طوری تنظیم دقیق کنیم که به عملکردی نزدیک به تنظیم دقیق کل مدل برسد.

در این میان LoRA به عنوان یکی از محبوب‌ترین روش‌های PEFT به شمار می‌رود و در زمینه‌های مختلف مزایای استفاده از آن نشان داده شده‌اند. ابتدا به بررسی نحوه کارکرد آن می‌پردازیم. پیش از معرفی LoRA در مقاله دیگری روشی برای اندازه‌گیری تعداد ابعاد درونی مدل‌های زبانی بزرگ معرفی شد و نشان داده شد که پارامترهای مدل‌های زبانی بزرگی در فضایی با ابعاد کم قرار می‌گیرند.

با نتایج آن مقاله، در مقاله دیگری روش LoRa برای تنظیم دقیق مدل‌ها معرفی شد. پشت ارائه این روش سه هدف motivation داشت. اول اینکه طبق مقاله بیان شده تعداد ابعاد مدل‌های زبانی بسیار کم است و می‌توان از این برای تنظیم دقیق بهینه استفاده کرد. دوم اینکه در خیلی از روش‌های PEFT با افزایش تعداد پارامترها لزوما عملکرد بهتر نمی‌شود در صورتی که ترجیح می‌دهیم با افزایش تعداد پارامترها عملکرد مدل را بهتر کند. نهایتا اینکه ترجیح می‌دهیم روش PEFT باعث کند شدن مدل نشود.

در این روش وزن‌های از قبل آموزش دیده مدل یعنی W_0 را هنگام تنظیم دقیق ثابت در نظر می‌گیریم. وزن‌های $\Delta W = BA$ را به صورت موازی با بخشی از وزن‌های W_0 تعریف می‌کنیم که با جمع این وزن‌های جدید با وزن‌های قدیمی مدل، وزن‌های آن را آپدیت می‌کنیم. همچنین ماتریس BA درجه r دارد که بسیار از تعداد ابعاد ورودی k و تعداد ابعاد خروجی d کوچک‌تر است. مقدار اولیه پارامترهای A از توزیع گوسی و مقادیر اولیه ماتریس B برابر صفر در نظر گرفته می‌شوند تا در ابتدا ΔW برابر صفر باشد و مدل تنها از وزن‌های خود استفاده کند و با تنظیم دقیق، وزن‌ها آپدیت شوند.

در روش LoRA هایپرپارامتر اصلی r یا همان درجه وزن‌های تعریف شده است که هر چقدر مقدار آن بزرگ‌تر باشد تعداد پارامترها هم بیشتر شده و مدل به عملکرد بهتری پس از تنظیم دقیق می‌رسد. در زمینه‌های مختلف از LoRA استفاده شده است و بهبود عملکرد مدل‌ها نشان داده شده است. می‌توان در شکل ۱۸ مشاهده کرد که پس از اعمال LoRA وزن‌های ابتدایی مدل W_0 تغییر می‌کنند و در نهایتا به W_0+BA می‌رسند.



شکل ۵. روش کارکارد **LoRA** - منبع از اسلایدهای درس مدل‌های زبانی بزرگ

مانند روش‌های اداپتور می‌توان قسمت‌های اضافه شده به مدل را ذخیره کرد و با داشتن چند LoRAی مختلف، بسته به نوع task از LoRAی مورد نظر استفاده کنیم. همچنین می‌توان وزن‌ها اضافه شده را موازی با هر یک از لایه‌های مدل حتی لایه‌های توجه در نظر گرفت.

دلایل زیادی برای استفاده از این روش نسبت به دیگر روش‌های PEFT وجود دارد. از جمله اینکه بر خلاف روش‌هایی مثل اداپتورها، استفاده از این روش سر بار زمان اجرای زیادی ندارد زیرا وزن‌های اضافه شده به صورت موازی و درکنار لایه‌ها قرار می‌گیرند. به همین دلیل می‌توان از قابلیت‌های موازی‌سازی GPU هم استفاده کرد. یکی دیگر از مزیت‌های LoRA عملکرد بسیار خوب مدل‌ها پس از تنظیم دقیق با این روش است به طوریکه در بسیاری از موارد حتی با r کوچک عملکرد به تنظیم دقیق کل مدل نزدیک می‌شود. همچنین با ثابت نگه داشتن وزن‌های اصلی مدل، می‌توان با داشتن LoRAهای مختلف و تنها

یک مدل سنگین، هر LoRA را روی task مختلفی تنظیم دقیق کرد و با مصرف حافظه به اندازه یک مدل، بتوان مدل‌های مختلفی با جایگزینی LoRAها استفاده کرد.

۲-۱-۹. بررسی اثر تابع خطای CLIP روی مقاوت

مقاله‌های دیگری نیز وجود دارند که مشابه این مقاله سعی کرده‌اند با معرفی توابع خطای دیگر پایداری مدل را در مقابله نمونه‌های خصمانه افزایش دهند.

یکی از آن‌ها این مقاله است که در آن در روشی با نام FARE تابع خطای زیر معرفی شده است:

$$Loss(x, x_a) = \max_{||x_a - x||_\infty \leq \varepsilon} ||\theta(x_a) - \theta(x)||_2^2$$

همان طور که از تابع خطا می‌توان دید سعی می‌کند پارامترهای مدل را طوری آپدیت کند که بردارهای به دست آمده در خروجی برای تصویر و نمونه خصمانه آن یکسان باشد. یکی از مزیت‌های تابع خطای بالا آن است که چون با انکودر متنی کاری ندارد، می‌توان مدل را بدون داشتن برچسب‌های تصاویر به صورت unsupervised آموزش داد.

همچنین با نزدیک کردن بردار خروجی تصاویر به هم ادعا شده است که قابلیت طبقه‌بندی تک‌ضرب مدل به خوبی حفظ شده است. در بخش نتایج صحت میانگین روی دیتاست‌های مختلف نسبت به صحت میانگین مقاله TeCoA گزارش شده است و در بهترین حالت مدل به ۷ درصد صحت بیشتری رسیده است.

در مقاله دیگری از روش prompt tuning هر دو انکودر متن و تصویر به همراه knowledge distillation استفاده شده است تا به عملکرد مناسبی برسند. هر دو مدل teacher و student به صورت همزمان تنظیم دقیق می‌شوند. تابع خطای مورد استفاده برای مدل teacher در این روش بدین صورت است:

$$\underset{P^{(T)}}{\operatorname{argmin}}\, L_{CE}(T(x), y) + \beta \cdot L_{KL}\big(T(x), S(x')\big)$$

که در رابطه بالا P(T) نشان دهنده پرامت‌های متن و تصویر است که مدل یاد می‌گیرد تغییر دهد تا به دقت مناسبی برسد. همچنین مدل teacher تنها تصاویر تمیز را دریافت می‌کند می‌توان دید قسمت اول تابع خطای بالا برای عملکرد بهتر مدل است و قسمت دوم برای آن است که مدل سعی کند feedbackهایی به مدل student بدهد تا به یادگیری آن کمک کند. مدل student نمونه‌های خصمانه را با تابع خطای زیر می‌آموزد:

$$\underset{P^{(S)}}{\operatorname{argmin}}\, L_{KL}\big(T(x), S(x')\big)$$

مدل دانش‌آموز سعی می‌کند با یادگیری ایجاد تغییرات لازم در پرامت‌ها به کمک P(S) پیش‌بینی‌های خود را به مدل teacher که تنها نمونه‌های تمیز را می‌بینند نزدیک کند و خطای خود را کاهش دهد. در

نهایت صحت مدل روی تصاویر خصمانه با حمله Auto-Attack به طور میانگین روی چند دیتاست به ۴۲.۸۸ درصد می‌رسد و ادعا می‌کند نسبت به روش‌های دیگر Prompt tuning با کمک گرفتن از یک مدل teacher به عملکرد بهتری می‌رسد.

۲-۲. آماده‌سازی برای آموزش

۲-۲-۱. آماده‌سازی دادگان

دادگان CIFAR-10 جزو دادگان پر استفاده در حوزه بینایی ماشین است که شامل تصاویر رنگی با ابعاد ۳۲ در ۳۲ پیکسل از ۱۰ کلاس مختلف است که هر کلاس شامل ۶۰۰۰ تصویر منحصر به فرد است. کلاس‌های این مجموعه داده عبارتند از هواپیما، خودرو، پرنده، گربه، گوزن، سگ، قورباغه، اسب، کشتی و کامیون.

برای اینکه از تصاویر برای مدل CLIP استفاده کنیم، ابتدا تصاویر را به سایز ۲۲۴ پیکسل تغییر اندازه می‌دهیم و سپس با میانگین و انحراف معیار تصاویر مدل CLIP نرمالایز می‌کنیم. سپس تصاویر را جدا می‌کنیم تا به ۱۰۰۰۰ تصویر ارزیابی، ۲۰۰۰ تصویر اعتبارسنجی و ۸۰۰۰ تصویر آموزش برسیم.

شکل ۶. چند نمونه از تصاویر **CIFAR-10** به همراه برچسب

۲-۲-۲. لود کردن مدل‌ها

سایت hugging face ابزارها و مدل‌های مختلفی را در حوزه‌های مختلف از جمله یادگیری زبان طبیعی در اختیار توسعه‌دهندگان قرار می‌دهد. دو مدل CLIP و ResNet20 را لود می‌کنیم. مدل CLIP مدلی است که قصد حمله به آن را داریم و چون می‌خواهیم حملات خود را تا حد امکان به واقعیت نزدیک کنیم، با کمک مدل ResNet20 که روی تصاویر CIFAR-10 آموزش دیده است، حمله انتقالی انجام می‌دهیم.

پس از لود کردن مدل‌ها به ازای هر یک از ده کلاس موجود، متنی مشابه a photo of a {class_category} تشکیل می‌دهیم و با کمک قسمت انکودر متنی مدل CLIP ده بردار مربوط به هر متن را دریافت و نرمالایز می‌کنیم تا در ادامه از آن‌ها برای به دست آوردن کلاس طبقه‌بندی شده توسط مدل استفاده کنیم.

با توجه به اینکه در این پروژه نیاز است تصاویر را به هر دو صورت تمیز و خصمانه استفاده کنیم، کلاس جدیدی را با تغییر کلاس DataLoader فریمورک پایتورچ ایجاد می‌کنیم. این کلاس میانگین و انحراف معیار هر تصاویر CLIP و CIFAR-10 را دریافت می‌کند. همچنین به آن تابع حمله را نیز می‌دهیم و برای این کلاس سه حالت clean، attack و both مشخص می‌کنیم که در حالت اول تصاویر تمیز، در حالت دوم تصاویر خصمانه و در حالت سوم هر دو را خروجی می‌دهد.

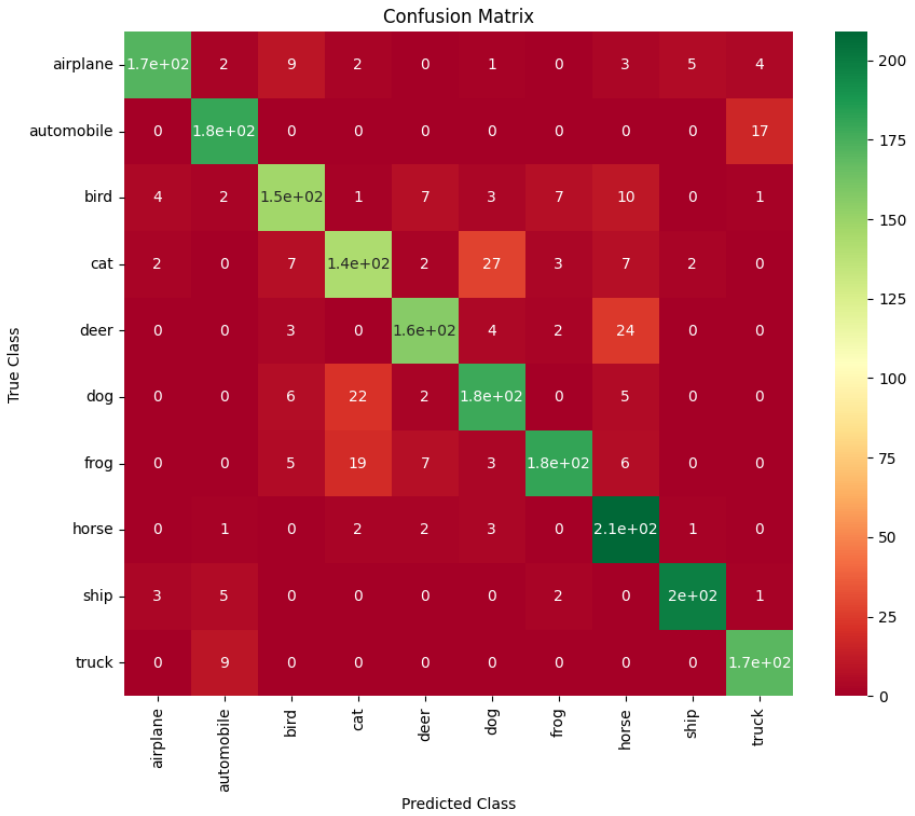
در صورتی که نیاز به ایجاد تصاویر خصمانه باشد، با توجه به اینکه مدل ResNet20 به عنوان مدل در دسترس برای ایجاد نمونه‌های خصمانه روی تصاویر CIFAR-10 با اندازه ۳۲ پیکسل آموزش دیده است، ابتدا تصاویر را با ضرب در میانگین CLIP و جمع با انحراف معیار آن denormalize می‌کنیم و پس از تغییر اندازه تصاویر به ۲۲۴ پیکسل، تصاویر را با میانگین و انحراف معیار تصاویر CIFAR-10 نرمالایز می‌کنیم و به تابع حمله می‌دهیم تا با کمک مدل ResNet20 نمونه‌های خصمانه را تولید کند. سپس با دریافت نمونه‌های خصمانه، دوباره تصاویر را denormalize کرده و پس از تغییر اندازه به ۲۲۴ پیکسل، با میانگین و انحراف معیار CLIP نرمالایز می‌کنیم و به عنوان خروجی برمی‌گردانیم.

ما از حمله PGD با کمک فریمورک torchattacks با اپسیلون ۸/۲۵۵، آلفای ۲/۲۵۵ و تعداد گام‌های ۷ استفاده می‌کنیم و گزینه random_start را هم درست در نظر می‌گیریم تا با هر حمله نمونه‌های خصمانه جدیدی تولید کنیم.

۲-۳. تحلیل نتایج

۲-۳-۱. تحلیل نتایج روی تصاویر تمیز

در ابتدا با دادن تصاویر اعتبارسنجی تمیز به مدل CLIP و به دست آوردن شباهت بردار تصاویر به متون با ضرب داخلی بردارهای نرمالایز شده، عملکرد طبقه‌بندی تک‌ضرب مدل CLIP را روی تصاویر CIFAR-10 پیش از انجام حمله را بررسی می‌کنیم.



شکل ۷. ماتریس آشفتگی مدل **CLIP** روی تصاویر تمیز پیش از تنظیم دقیق

می‌توان دید به طور کلی عملکرد مدل مناسب است و بیشترین میزان خطا بین دو کلاس ۲۷ مورد بین سگ و گربه بوده است. همچنین اکثر خطاهای دیگر نیز بین کلاس‌های حیوانات یا کلاس خودرو و کامیون بوده است که با توجه به پایین بودن کیفیت تصاویر مدل خطای پایینی داشته است.

جدول 1. معیارهای ارزیابی مدل روی تصاویر تمیز پیش از تنظیم دقیق

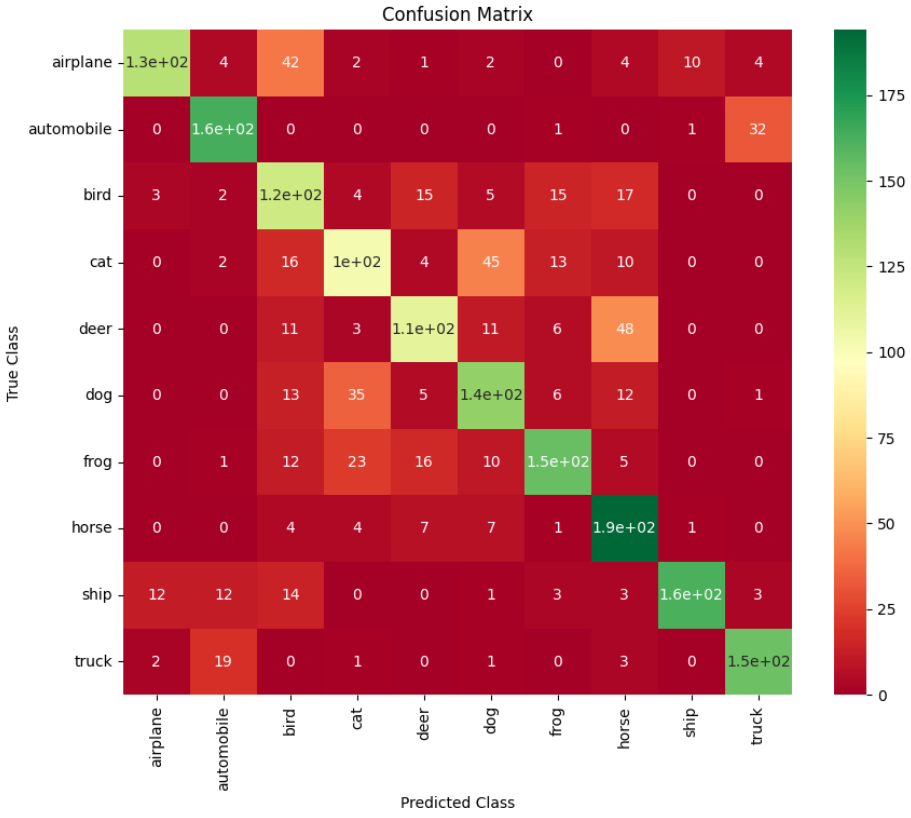
	Accuracy	**Precision**	**Recall**	**F1 score**
Micro	0.8675	0.8675	0.8675	0.8675
Macro	0.8675	0.8703	0.8667	0.8669
Wieghted	0.8675	0.8707	0.8675	0.8674
Airplane	0.9825	0.9503	0.8687	0.9077
automobile	0.9820	0.9045	0.9137	0.9091
Bird	0.9675	0.8305	0.8077	0.8189
Cat	0.9520	0.7566	0.7409	0.7487
Deer	0.9735	0.8864	0.8254	0.8548
Dog	0.9620	0.8128	0.8357	0.8241

Frog	0.9730	0.9282	0.8190	0.8702
Horse	0.9680	0.7917	0.9587	0.8672
Ship	0.9905	0.9614	0.9476	0.9544
Truck	0.9840	0.8808	0.9497	0.9140

می‌توان دید مدل به صحت کلی ۸۶ درصد رسیده است و صحت، دقت، بازیابی، و امتیاز F1 آن برای تمامی کلاس‌ها به جز کلاس گربه بالای ۸۰ درصد است.

۲-۳-۲. تحلیل نتایج روی تصاویر خصمانه

حال که عملکرد مدل را روی تصاویر تمیز ارزیابی کردیم و مدل توانست به خوبی کلاس‌ها را با وجود کیفیت کم تصاویر تشخیص دهد، نمونه‌های خصمانه را با کمک مدل ResNet20 تولید می‌کنیم و به مدل می‌دهیم تا عملکرد آن را بررسی کنیم.



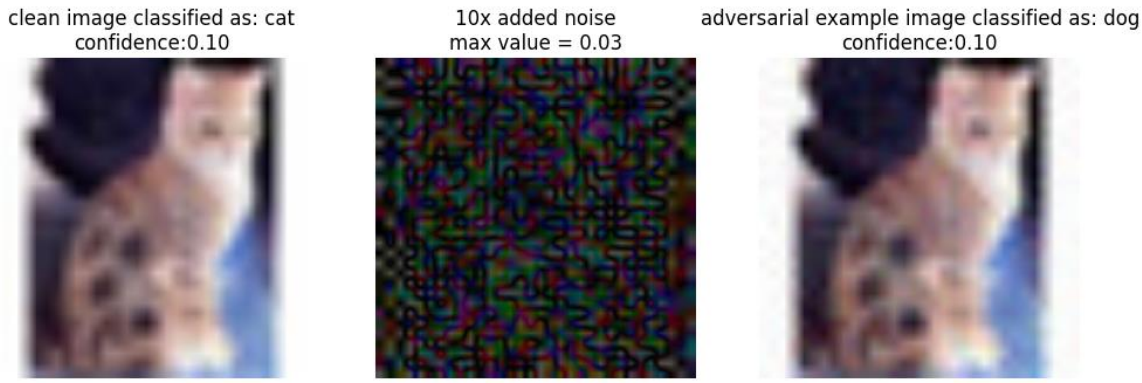
شکل ۸. ماتریس آشفتگی مدل روی نمونه‌های خصمانه پیش از تنظیم دقیق

می‌توان دید با دادن نمونه‌های خصمانه عملکرد مدل کاهش میابد برای مثال با اینکه مدل می‌توانست به خوبی دو کلاس پرنده و هواپیما را تشخیص دهد، الان ۴۲ مورد خطا بین این دو کلاس وجود دارد. به طرز مشابه عملکرد آن روی کلاس‌های دیگر مخصوصا در تشخیص بین حیوانات مختلف افت داشته است.

جدول2. معیارهای ارزیابی مدل روی نمونه‌های خصمانه پیش از تنظیم دقیق

	Accuracy	**Precision**	**Recall**	**F1 score**
Micro	0.7150	0.7150	0.7150	0.7150
Macro	0.7150	0.7276	0.7134	0.7143
Wieghted	0.7150	0.7294	0.7150	0.7159
airplane	0.9570	0.8836	0.6515	0.7500
automobile	0.9630	0.8030	0.8274	0.8150
bird	0.9135	0.5193	0.6648	0.5831
cat	0.9190	0.5886	0.5337	0.5598
deer	0.9365	0.6962	0.5820	0.6340
dog	0.9230	0.6323	0.6620	0.6468
frog	0.9440	0.7739	0.6968	0.7333
horse	0.9370	0.6554	0.8899	0.7549
ship	0.9700	0.9310	0.7714	0.8438
truck	0.9670	0.7927	0.8547	0.8226

می‌توان در جدول بالا دید که عملکرد مدل به طور کلی حدود ۱۵ درصد افت داشته است. ضعیف‌ترین عملکرد از نظر امتیاز F1 مربوط به کلاس‌های حیوانات مخصوصا گربه و پرنده بوده است که مقدار آن به ۵۵ و ۵۸ درصد رسیده است در حالیکه روی تصاویر تمیز به امتیاز F1 ۷۴ و ۸۱ درصد می‌رسید.



شکل ۹. یک نمونه خصمانه موفق به همراه نویز تولید شده

در شکل بالا مدل تصویر سمت چپ را به درستی گربه تشخیص داده است ولی با اضافه کردن نویز مشخص شده به تصویر سمت راست می‌رسیم که از دید انسان تفاوت زیادی با تصویر چپ ندارد ولی مدل به اشتباه آن را سگ تشخیص می‌دهد. همچنین در وسط نیز نویز اضافه شده مشخص است که به دلیل امکان منفی بودن، مقادیر آن را بین صفر و یک نرمالایز کردیم و برای بهتر مشخص شدن ضرب در ده کردیم. همچنین بیشترین مقدار نویز حدود ۰.۰۳ بود که با اپسیلون داده شده به تابع حمله مطابقت دارد و مشخص است در صورتی که حمله جعبه سفید یا با نویز بیشتری انجام می‌دادیم احتمال موفقیت افزایش پیدا می‌کرد.

۲-۴. تنظیم دقیق و تحلیل نتایج روی تصاویر اعتبارسنجی خصمانه

۲-۴-۱. تحلیل نتایج پس از آموزش با خطای corssentropy

در مقاله روش‌های مختلفی برای آموزش مدل در مقابل حملات خصمانه بررسی شده است. البته در تمامی روش‌ها از adversarial training استفاده شده است. در این روش به مدل نمونه‌های خصمانه را نشان می‌دهیم و سعی می‌کنیم خطای آن را روی این نمونه‌ها کاهش دهیم تا به درستی این نمونه‌ها را هم تشخیص دهد. روش‌های بسیاری در دفاع در مقابل حملات خصمانه معرفی شده‌اند ولی برای اکثر این روش‌ها با ایجاد نمونه‌های خصمانه دقیق‌تر می‌توان به عملکرد مدل لطمه زد ولی روش adversarial training یکی از بهترین روش‌ها بوده و با اینکه عملکرد مدل روی داده‌های تمیز هم کمی کاهش میابد ولی حمله به مدل سخت‌تر می‌شود.

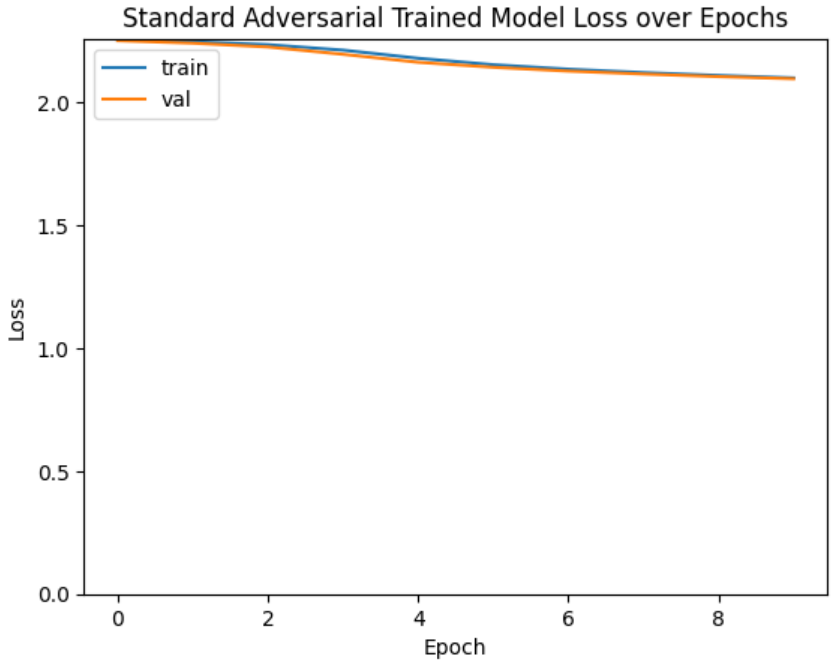
با توجه به اینکه مدل CLIP حدود ۱۵۱.۷۷ میلیون پارامتر دارد، آموزش آن روی GPUهای در دسترس با حافظه کم ممکن نمی‌باشد. برای همین ما از روش‌های PEFT و به طور دقیق‌تر از LORA برای تنظیم دقیق بخشی از پارامترهای مدل استفاده می‌کنیم تا به عملکردی نزدیک به تنظیم دقیق تمام وزن‌ها برسیم.

بدین ترتیب از LORA با درجه r=8 و پارامتر آلفای ۳۲ در ماتریس‌های query و value از مکانیزم توجه استفاده می‌کنیم تا کمتر از ۵۰۰ هزار پارامتر را یاد بگیریم که تنها حدود ۰.۳۲ درصد از پارامترهای مدل را شامل می‌شود.

پس از نوشتن توابع آموزش مدل روی تصاویر خصمانه و بردن دیتالودر به حالت attack، در ابتدا از خطای دوم بررسی شده در مقاله استفاده می‌کنیم. طبق این خطا مدل با دریافت تصاویر خصمانه، با ضرب داخلی بردار هر تصویر با بردار هر یک از متون شباهت میان تصاویر و متون را به دست می‌آورد و متنی که بیشترین شباهت را به تصویر داشت به عنوان کلاس تشخیص داده شده توسط مدل در نظر می‌گیریم. سپس با داشتن کلاس‌های تشخیص داده شده و کلاس‌های واقعی، با تابع خطای crossentropy وزن‌های مدل را آپدیت می‌کنیم.

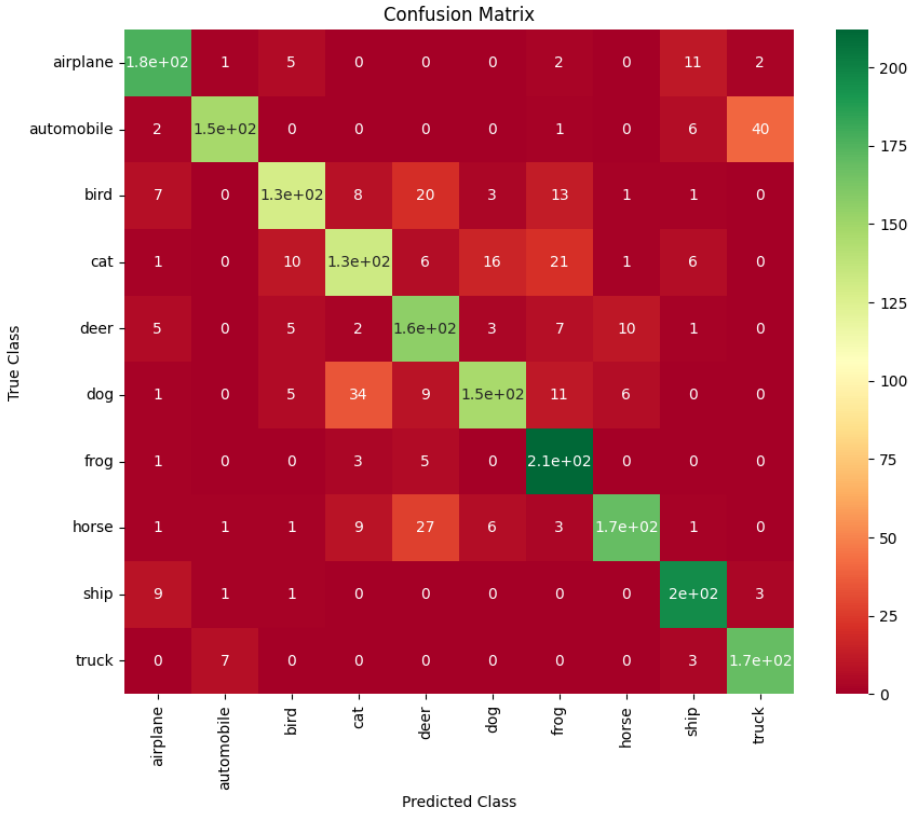
برای تنظیم دقیق مدل ما از اندازه بچ ۶۴ و weight_decay ۰.۰۰۰۱ استفاده می‌کنیم. همچنین momentum را برابر ۰.۹ در نظر گرفتیم و مدل را به مدت ۱۰ ایپاک آموزش می‌دهیم. با بررسی نرخ‌یادگیری‌های مختلف، مقدار ۰.۰۰۱ را در نهایت در نظر گرفتیم. همچنین مشابه مقاله از تابع بهینه‌ساز SGD استفاده کردیم.

مدل را روی تصاویر آموزشی آموزش می‌دهیم و عملکرد آن را روی تصاویر اعتبارسنجی بررسی می‌کنیم. در ابتدا خطای مدل برابر ۲.۲۵ است و در نهایت به ۲.۰۹ روی تصاویر اعتبارسنجی می‌رسد.



شکل ۱۰. نمودار تغییرات خطای مدل حین آموزش

در نمودار بالا می‌توان دید خطا پس از ۱۰ ایپاک تقریبا همگرا می‌شود پس از همین پارامترها برای آموزش مدل‌های بعدی هم استفاده می‌کنیم و با توجه به شباهت تغییرات خطا در مدل‌های بعدی، تغییرات خطای آن‌ها را رسم نمی‌کنیم.



شکل ۱۱. ماتریس آشفتگی مدل روی تصاویر خصمانه پس از تنظیم دقیق با خطای **crossentropy**

از ماتریس آشفتگی می‌توان دید پس از تنظیم دقیق با خطای crossentropy به کمک نمونه‌های خصمانه خطای مدل بسیار کاهش یافته است و برخلاف قبل به درستی می‌تواند کلاس‌های پرنده و هواپیما را تشخیص دهد. البته هنوز با حالت ابتدایی فاصله دارد.

جدول3. معیارهای ارزیابی مدل روی نمونه‌های خصمانه پس از آموزش با خطای **crossentropy**

	Accuracy	**Precision**	**Recall**	**F1 score**
Micro	0.8175	0.8175	0.8175	0.8175
Macro	0.8175	0.8223	0.8165	0.8145
Wieghted	0.8175	0.8242	0.8175	0.8159
Airplane	0.9760	0.8676	0.8939	0.8806

Automobile	0.9705	0.9367	0.7513	0.8338
Bird	0.9600	0.8269	0.7088	0.7633
Cat	0.9415	0.7021	0.6839	0.6929
Deer	0.9500	0.6996	0.8254	0.7573
Dog	0.9530	0.8400	0.6901	0.7577
Frog	0.9665	0.7852	0.9593	0.8635
Horse	0.9665	0.9037	0.7752	0.8346
Ship	0.9785	0.8711	0.9333	0.9011
Truck	0.9725	0.7897	0.9441	0.8601

از جدول بالا مشخص است که عملکرد مدل بسیار بهتر شده است ولی همچنان با عملکرد آن روی تصاویر تمیز حدود ۵ درصد فاصله دارد. باز هم مشابه قبل بیشترین خطا در تشخیص حیوانات از هم وجود دارد.

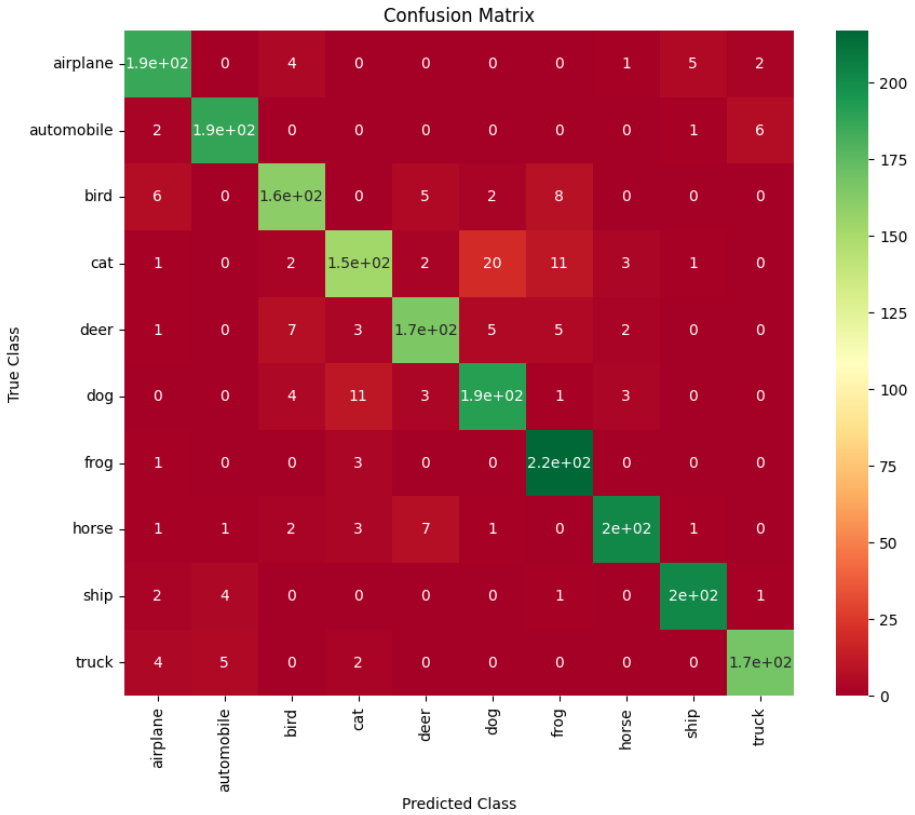
۲-۴-۲. تحلیل نتایج پس از آموزش با خطای TeCoA

حال از تابع خطای معرفی شده در مقاله با نام TeCoA استفاده می‌کنیم. این تابع خطا برخلاف تابع خطای قبل contrastive است یعنی سعی می‌کند نمونه‌های مثبت در هر بچ را به هم نزدیک کند. به طور دقیق‌تر تابع خطای TeCoA به ازای هر تصویر تنها متن آن تصویر را به عنوان نمونه مثبت برای آن در نظر می‌گیرد و تمام متون دیگر برای نمونه‌های دیگر در بچ را به عنوان نمونه‌های منفی در نظر می‌گیرد و تابع خطا را بدین شکل حساب می‌کند:

$$\mathcal{L}_s(x,t,y) = -\mathbb{E}_{i,j}[y_{i,j}\, log \frac{\exp\left(\cos\left(z_i^{(I)}, z_j^{(T)}\right)/\tau\right)}{\sum_k \exp\left(\cos\left(z_i^{(I)}, z_k^{(T)}\right)/\tau\right)}]$$

در رابطه بالا $z_i^{(I)}$ نشان دهنده بردار خروجی مدل برای تصویر iام و $z_j^{(T)}$ هم نشان‌دهنده بردار خروجی مدل برای متن jام است. همچنین cos نشان دهنده cosine similarity است و $y_{i,j}$ هم تنها برای هر تصویر و متن متناظر آن ۱ و برای بقیه متون تصاویر صفر است. τ هم یک مقدار اسکالر است که در مقاله مقدار آن گزارش نشده و ما در ابتدا مقدار ۰.۰۱ و سپس ۰.۱ را بررسی می‌کنیم. بدین ترتیب با کاهش تابع خطا مدل سعی می‌کند بردارهای متن و تصویر کلاس خاصی را به هم نزدیک کند و به دلیل موجود بودن بقیه بردارها در مخرج، شباهت بین آن‌ها کم می‌شود.

با نوشتن تابع خطا مدل را به کمک آن آموزش می‌دهیم و خطای مدل روی داده اعتبارسنجی از ۲.۳۲ آغاز و در نهایت به ۲.۱۹ می‌رسد.



شکل ۱۲. TeCoA ماتریس آشفتگی مدل آموزش دیده با تابع خطای

در ماتریس آشفتگی بالا می‌توان دید همچنان تشخیص دو کلاس سگ و گربه برای مدل سخت است ولی به جز آن به نظر می‌رسد مدل به خوبی تمام کلاس‌های دیگر را تشخیص می‌دهد. بدین ترتیب برای بررسی بهتر مقادیر معیارها را حساب می‌کنیم.

جدول 4. معیارهای ارزیابی مدل روی نمونه‌های خصمانه پس از آموزش با خطای **TeCoA**

	Accuracy	**Precision**	**Recall**	**F1 score**
Micro	0.9170	0.9170	0.9170	0.9170
Macro	0.9170	0.9171	0.9155	0.9158
Wieghted	0.9170	0.9171	0.9170	0.9165
airplane	0.9850	0.9118	0.9394	0.9254
automobile	0.9905	0.9495	0.9543	0.9519
bird	0.9800	0.8944	0.8846	0.8895

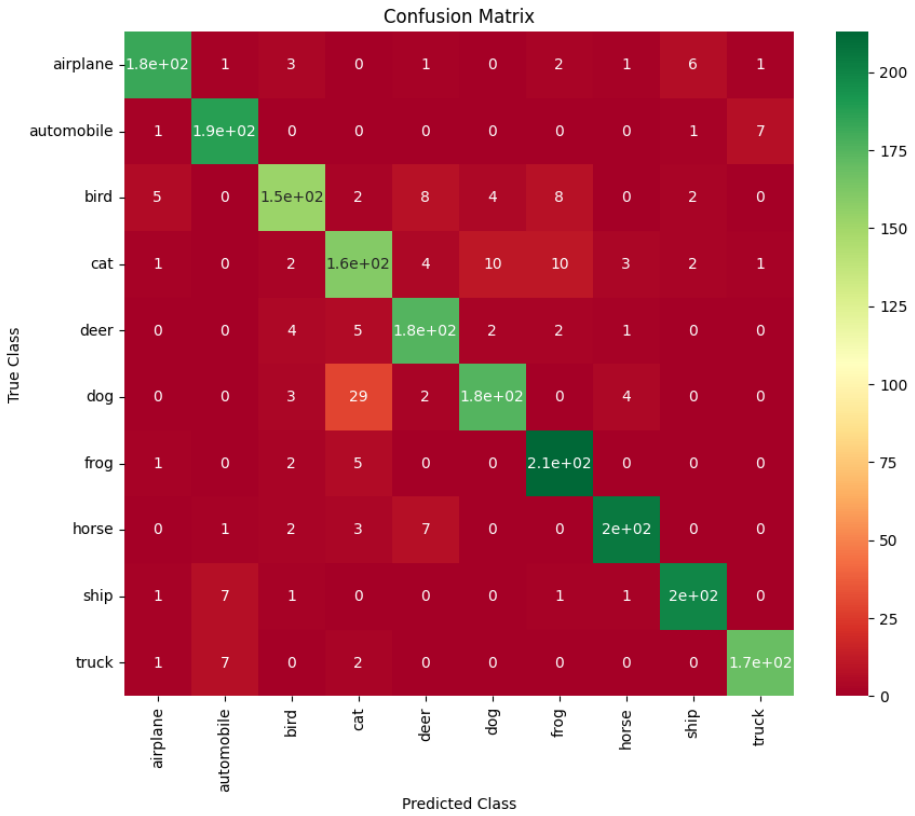
cat	0.9690	0.8743	0.7927	0.8315
deer	0.9800	0.9071	0.8783	0.8925
dog	0.9750	0.8721	0.8967	0.8843
frog	0.9850	0.8930	0.9819	0.9353
horse	0.9875	0.9573	0.9266	0.9417
ship	0.9920	0.9619	0.9619	0.9619
truck	0.9900	0.9492	0.9385	0.9438

از جدول بالا مشخص است که عملکرد مدل حتی نسبت به مدل اولیه روی تصاویر تمیز نیز بهتر شده است. این به دلیل تنظیم دقیق مناسب با این تابع خطا است که باعث شده نه تنها عملکرد مدل با دیدن تصاویر خصمانه افت پیدا نکند بلکه حتی بهتر از قبل روی این تصاویر کار کند. اکنون مقادیر تمام معیارها برای تمام کلاس‌ها بالای ۸۰ درصد شده و مدل توانایی تشخیص کلاس‌ها را دارد.

۲-۴-۳. تحلیل نتایج پس از افزایش دما

حال هایپرپارامتر دما را به ۰.۱ افزایش می‌دهیم و با آموزش مجدد مدل تاثیر این هایپرپارامتر را بررسی می‌کنیم. این هایپرپارامتر با تقسیم شدن بر مقادیر شباهت‌ها، اگر کوچک انتخاب شود باعث می‌شود اختلاف بین شباهت‌های کم و زیاد بیشتر شود و در صورت صفر بودن دما تنها بیشترین شباهت یک و بقیه صفر می‌شوند. همچنین اگر به سمت بی‌نهایت برود مقادیر تمام شباهت‌ها یکسان می‌شود.

خطای مدل از ۲.۹۳ به ۲.۳۷ روی داده اعتبارسنجی می‌رسد.



شکل ۱۳. ماتریس آشفتگی پس از افزایش دما با خطای **TeCoA**

از ماتریس آشفتگی به نظر می‌رسد عملکرد مدل مشابه قبل است و به جای تشخیص نادرست سگ به جای گربه حال بسیاری از سگ‌ها را گربه تشخیص می‌دهد.

جدول5. مقادیر معیارها پس از افزایش دما و تنظیم دقیق با خطای **TeCoA**

	Accuracy	**Precision**	**Recall**	**F1 score**
Micro	0.9100	0.9100	0.9100	0.9100
Macro	0.9100	0.9104	0.9092	0.9092
Wieghted	0.9100	0.9112	0.9100	0.9099
airplane	0.9875	0.9482	0.9242	0.9361
automobile	0.9875	0.9216	0.9543	0.9377
bird	0.9770	0.9000	0.8407	0.8693
cat	0.9605	0.7767	0.8290	0.8020
deer	0.9820	0.8883	0.9259	0.9067
dog	0.9730	0.9162	0.8216	0.8663
frog	0.9845	0.9025	0.9638	0.9322

horse	0.9885	0.9535	0.9404	0.9469
ship	0.9890	0.9476	0.9476	0.9476
truck	0.9905	0.9494	0.9441	0.9468

می‌توان دید صحت به مقدار ۰.۷ درصد و بقیه معیارها نیز به مقدار بسیار کمی کاهش یافته‌اند. پس ما در ادامه نیز از مقدار ۰.۰۱ برای دما استفاده می‌کنیم.

۲-۴-۴. تحلیل نتایج پس از آموزش با خطای TeCoA به روش VPT

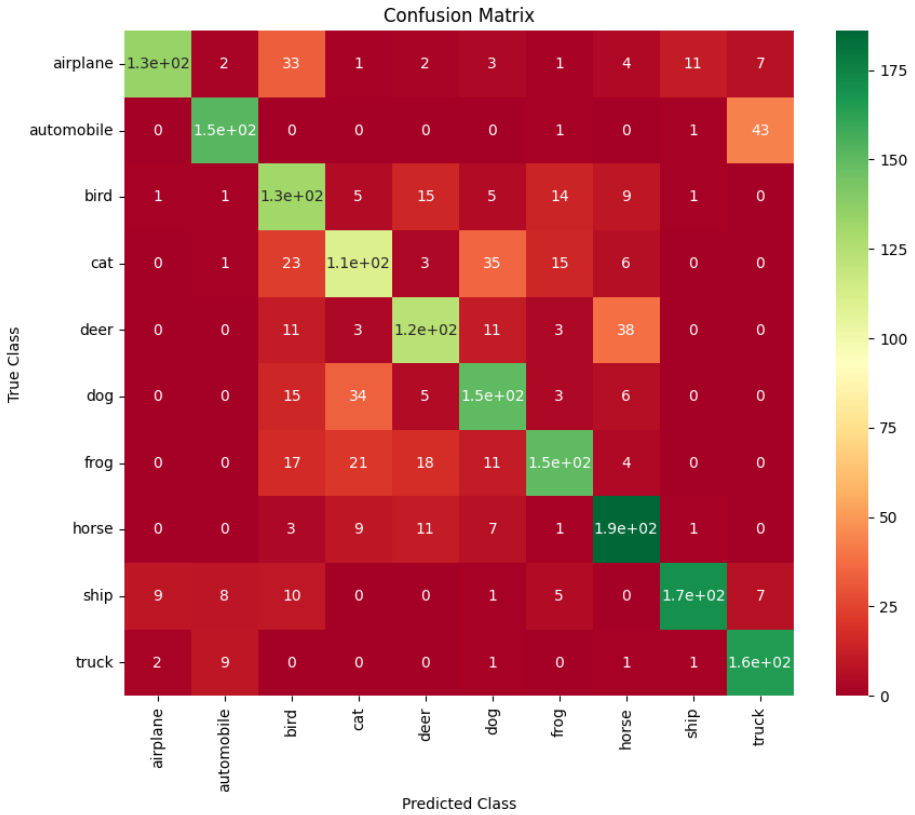
در مقاله یکی از روش‌هایی که بررسی شده است VPT یا Visual Prompt Tuning است. روش‌های VPT هم جزو روش‌های PEFT محسوب می‌شوند و از تحلیل زبان‌های طبیعی به بینایی ماشین راه یافته‌اند. در این روش‌ها به نوعی تغییراتی در ورودی‌ها اعمال می‌شود تا کار مدل را راحت‌تر کند و باعث عملکرد بهتر آن شود.

در مقاله دو روش VPT مختلف بررسی شده‌اند. در روش اول طبق مقاله روش، تنها لایه‌ای یکسان با مقادیر تمام تصاویر جمع می‌شود و با یادگیری پارامترهای آن لایه سعی می‌شود اثر نویزهای اضافه شده کاهش یابد. در روش دوم توکنی به توکن‌های تصاویر که به ترنسفورمر پردازش تصویر مدل CLIP می‌روند اضافه می‌شود و مدل با یادگیری توکن جدید به عملکرد بهتری می‌رسد.

با توجه به پیچیدگی پیاده‌سازی روش دوم ما تنها روش اول را پیاده می‌کنیم و با اضافه کردن لایه‌ای یکسان به تمام تصویر مدل را تنظیم دقیق می‌کنیم. خطای مدل از ۲.۶۸ به ۲.۶۳ می‌رسد.

شکل ۱۴. نمونه‌ای از تصاویر پس از جمع شدن با لایه اضافه شده

از تصویر بالا می‌توان دید تغییرات اعمال شده بر تصویر آن‌قدر جزئی هستند که با چشم قابل دیدن نمی‌باشند.



شکل ۱۵. ماتریس آشفتگی با خطای **TeCoA** با روش **VPT**

می‌توان دید عملکرد مدل مناسب نیست و این روش به خوبی عمل نکرده است. البته در مقاله با آموزش با این روش نیز مدل به عملکرد مناسبی رسید ولی به دلیل بیان جزئیات کم و اینکه در خود مقاله اصلی این روش VPT نیز بیان شده بود که به جز اضافه کردن یک لایه یکسان به تمام تصاویر می‌توان کارهای دیگری هم کرد و احتمال دارد روش به کار گرفته شده متفاوت باشد.

جدول6. مقادیر معیارها با آموزش با خطای **TeCoA** به روش **VPT**

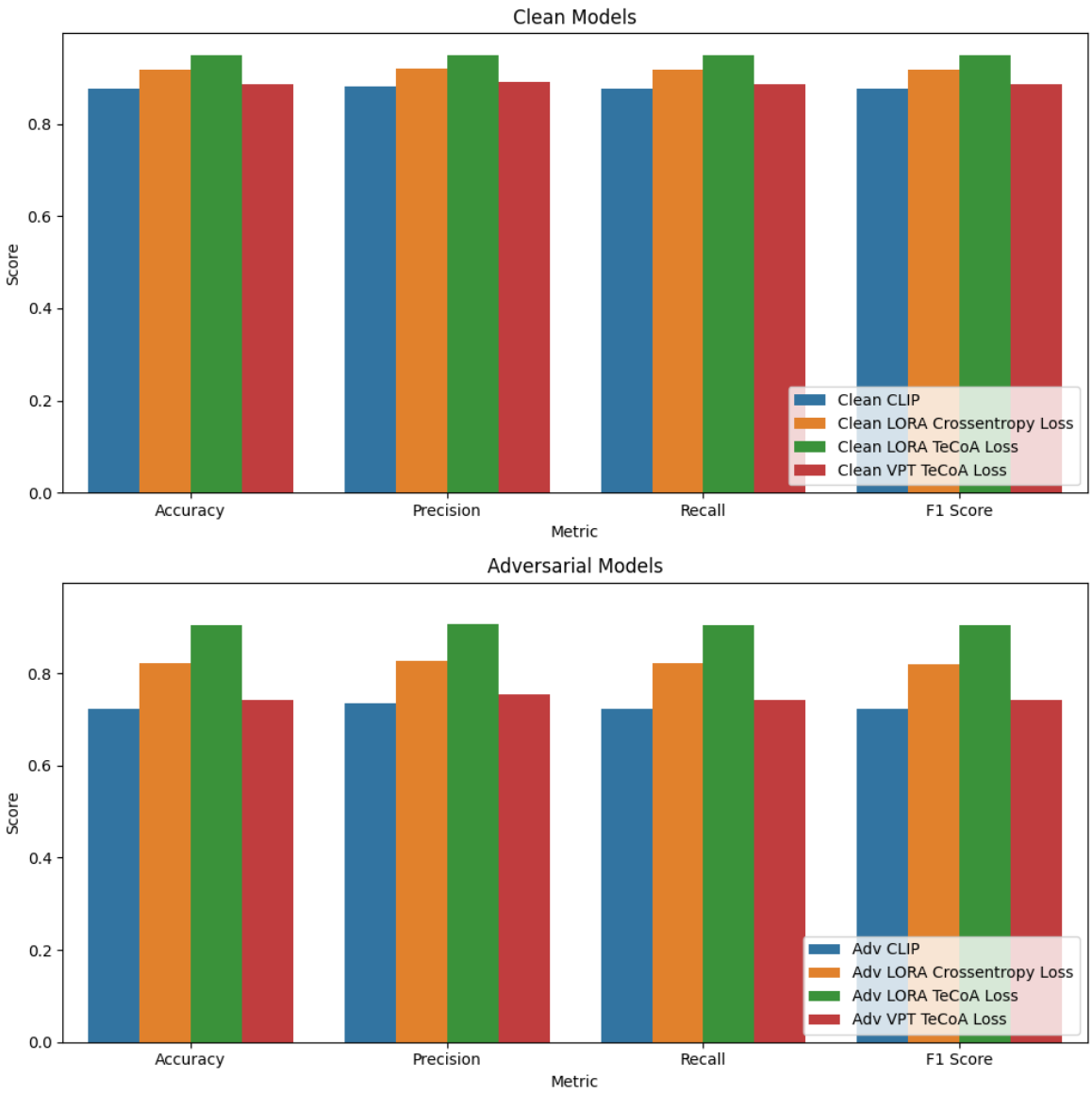
	Accuracy	**Precision**	**Recall**	**F1 score**
Micro	0.7355	0.7355	0.7355	0.7355
Macro	0.7355	0.7473	0.7356	0.7357
Wieghted	0.7355	0.7502	0.7355	0.7373
airplane	0.9620	0.9178	0.6768	0.7791
automobile	0.9670	0.8786	0.7716	0.8216

bird	0.9185	0.5391	0.7198	0.6165
cat	0.9220	0.6011	0.5699	0.5851
deer	0.9400	0.6949	0.6508	0.6721
dog	0.9315	0.6696	0.7042	0.6865
frog	0.9430	0.7772	0.6787	0.7246
horse	0.9500	0.7323	0.8532	0.7881
ship	0.9725	0.9189	0.8095	0.8608
truck	0.9645	0.7432	0.9218	0.8229

می‌توان دید با این روش معیارهای کلی مدل حدود دو درصد بهبود یافته‌اند که از دو روش دیگر بسیار کمتر است. البته همان طور که نوشتیم احتمال دارد به دلیل پیاده‌سازی متفاوت با مقاله نتایج متفاوت باشد. همچنین در روشی که ما پیاده کردیم تعداد پارامترهای یادگیری یک دهم تعداد پارامترهای یادگیری در LORA بود و این خود می‌تواند دلیل ضعیف‌تر بودن این روش باشد.

۵-۲. تنظیم دقیق و تحلیل نتایج روی تصاویر ارزیابی

حال چهار روش ارزیابی شده روی تصاویر اعتبارسنجی را روی ۱۰۰۰۰ تصویر آموزشی و اعتبارسنجی، آموزش می‌دهیم و عملکرد مدل‌ها را روی تصاویر تمیز و خصمانه تست ارزیابی می‌کنیم. با توجه به اینکه در قسمت قبل بهترین عملکرد روی داده اعتبارسنجی خصمانه برای مدل آموزش دیده با خطای TeCoA بود، انتظار داریم حال نیز این مدل به بهترین عملکرد برسد.



شکل ۱۶. مقایسه عملکرد مدل‌ها روی تصاویر تست تمیز و خصمانه

از تصاویر بالا می‌توان دید همان طور که انتظار داریم عملکرد مدل‌ها روی تصاویر تمیز بهتر از نمونه‌های خصمانه است. همچنین بهترین عملکرد روی هر دو تصاویر تمیز و خصمانه به ترتیب از بهترین بدین صورت است: اول مدل آموزش دیده با خطای TeCoA با روش LORA، دوم مدل آموزش دیده با خطای crossentropy، سوم مدل آموزش دیده با روش VPT و آخر مدل پیش از آموزش. حال برای بررسی دقیق‌تر مقادیر معیارها را به صورت عددی بررسی می‌کنیم.

جدول7. مقادیر معیارها برای مدل روی داده تست خصمانه و تمیز

	Accuracy	Precision	Recall	F1 score
Clean CLIP	0.8765	0.8810	0.8765	0.8765
Clean LORA Crossentropy Loss	0.9170	0.9187	0.9170	0.9164
Clean LORA TeCoA Loss	0.9485	0.9490	0.9485	0.9485
Clean VPT TeCoA Loss	0.8866	0.8907	0.8866	0.8869
Adv CLIP	0.7231	0.7346	0.7231	0.7226
Adv LORA Crossentropy Loss	0.8208	0.8261	0.8208	0.8189
Adv LORA TeCoA Loss	0.9047	0.9065	0.9047	0.9050
Adv VPT TeCoA Loss	0.7417	0.7534	0.7417	0.7428

از جدول بالا هم مشخص است که آموزش با VPT حدودا دو درصد عملکرد را روی تصاویر خصمانه بهبود می‌دهد. آموزش با خطای crossentropy عملکرد را ۱۰ درصد بهبود می‌دهد و آموزش با LORA با خطای TeCoA هم عملکرد را حدود ۱۸ درصد افزایش می‌دهد.

همچنین روی تصاویر تمیز نیز آموزش با VPT ۱ درصد، آموزش با خطای crossentropy ۴ درصد و آموزش با خطای TeCoA ۷ درصد عملکرد را بهتر می‌کند. دلیل عملکرد بهتر مدل روی تصاویر تمیز می‌توان به خاطر تنظیم دقیق باشد. می‌دانیم مدل CLIP روی تصاویر بسیار بیشتری آموزش دیده است و با تنظیم دقیق روی مجموعه خاصی از تصاویر می‌تواند باعث بهبود عملکرد مدل روی آن مجموعه تصاویر شود. البته بهتر است عملکرد مدل را روی مجموعه تصاویر دیگر هم سنجیده شود تا ارزیابی درستی از عملکرد کلی مدل برای طبقه‌بندی تک‌ضرب حاصل شود.

با توجه به اینکه مدل‌های تنظیم دقیق شده عملکرد بهتری روی هر دو تصاویر تمیز و خصمانه نسبت به مدل بدون تنظیم دقیق دارند، در صورت امکان و مخصوصا در صورت مشخص بودن کلاس‌ها و توزیع مجموعه تصاویر به نظر می‌رسد در صورت امکان تنظیم دقیق مزیت بسیاری دارد. البته روش VPT تاثیر زیادی بر عملکرد مدل نداشت البته با ایجاد تغییراتی در آن یا ترکیب آن با روش‌های دیگر ممکن است بتوان به عملکرد بهتری رسید. در نهایت استفاده از روش ارائه شده در مقاله عملکرد بسیار بهتری نسبت

به خطای crossentropy داشت و با توجه به اینکه در مقاله روشی هم برای استفاده از این خطا بدون وجود برچسب‌ها بیان شده، به نظر می‌رسد استفاده از این روش می‌تواند باعث بهترین گزینه برای افزایش عملکرد مدل و افزایش پایداری آن در مقابل حملات خصمانه باشد.

همچنین مهم است ذکر کنیم که نتایج گزارش شده در مقاله به دلیل تفاوت‌های بسیاری که در روش‌ها وجود داشته بسیار متفاوت شده است. از جمله آنکه به نظر می‌رسد حمله استفاده شده در مقاله جعبه سفید بوده و همچنین دیتاست‌های استفاده شده نیز متفاوت‌اند. همچنین در مقاله از تنظیم دقیق کل مدل استفاده شده در صورتی که ما از روش LORA که یک روش PEFT است استفاده کردیم. همچنین در مقاله روش‌های بیشتری هم برای تنظیم دقیق و هم برای حمله به مدل بررسی شده‌اند.

۲-۶. جمع‌بندی

ما در این تمرین روش‌های با مفاهیم بسیاری آشنا شدیم. ابتدا با مفهوم robustness در مدل‌ها و انواع روش‌های حمله‌ی خصمانه و دفاع در مقابل آن‌ها در مدل‌ها آشنا شدیم. همچنین با مدل CLIP که یک مدل multi modal است آشنا شدیم که چگونه کار می‌کند و چگونه ویژگی طبقه‌بندی تک‌ضرب در آن ایجاد شده است و این ویژگی چه مزایایی دارد. همچنین با روش‌های تنظیم دقیق مخصوصا تنظیم دقیق بهینه یا همان PEFT و به خصوص LORA آشنا شدیم که به ما امکان آموزش مدل‌های بسیار بزرگ را می‌دهند.

سپس با لود کردن مدل‌ها سعی کردیم مشابه واقعیت بدون داشتن هیچ اطلاعاتی از مدل CLIP به جز اینکه روی تصاویر CIFAR-10 قصد ارزیابی داریم، نمونه‌های خصمانه با روش حملات انتقالی و به کمک مدل ResNet20 تولید کنیم. سپس با بررسی عملکرد مدل روی نمونه‌های خصمانه سعی کردیم با روش‌های مختلفی آن را در مقابل حملات مقاوم کنیم.

ما چهار نوع اصلی از روش‌های تنظیم دقیق برای مقاوم کردن مدل را بررسی کردیم. در دو روش از LORA استفاده کردیم تا مدل را با دو تابع خطای crossentropy و TeCoa که در مقاله معرفی شده بود تنظیم دقیق کنیم و همچنین از روش VPT هم استفاده کردیم تا با پارامترهای کمتر مدل را تنظیم دقیق کنیم.

در نهایت با بررسی نتایج ادعای مقاله را در مورد عملکرد مدل پس از آموزش با خطای TeCoA نسبت به روش‌های دیگر تنظیم دقیق در محیط آموزش خود تایید کردیم. البته جای بررسی‌های دیگری مانند بررسی قابلیت طبقه‌بندی تک‌ضرب مدل CLIP روی کلاس‌های دیتاست‌های دیگر و استفاده از بقیه روش‌های ارائه شده در مقاله وجود دارد که امیدواریم در فرصتی مناسب بتوانیم بررسی کنیم.