به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

Trustworthy AI

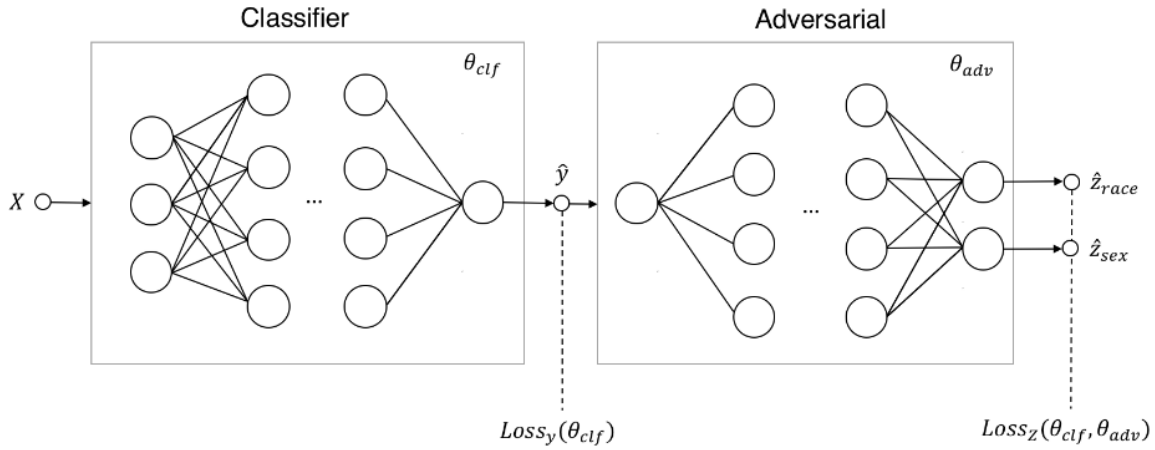
تمرین شماره سه

نام و نام خانوادگی	سیدسروش مجد
شماره دانشجویی	دانشجوی مهمان
تاریخ ارسال گزارش	۲۷ خرداد

فهرست گزارش سوالات

پرسش ۱ – Fairness

در این پرسش می‌خواهیم موضوع عدالت در یادگیری ماشین را بررسی کنیم و ببینیم مدل نسبت به فیچرهای خاصی مانند جنسیت بایاس است یا خیر. برای مثال در مدل‌های استخدام شرکت‌ها برای افراد خاصی با ویژگی حساس خاص مانند زن یا مرد بودن تبعیض قائل می‌شود یا خیر. در داده ویژگی‌هایی مانند جنسیت، تحصیلات، کشور و... وجود دارد که مدل طبقه‌بند باید پیشبینی کند که هر نمونه دارای درامد بالای ۵۰ هزار دلار می‌باشد یا خیر. در این بخش با استفاده از یک شبکه متخاصم بایاس داشتن این پیشبینی به فیچرهای خاص و حساس بررسی خواهد شد. طبقه‌بند با متخاصم دریک بازی صفر و یک رقابت می‌کند. طبقه بندی کننده باید پیشبینی‌های خوبی انجام دهد اما اگر متخاصم تصمیمات ناعادلانه را تشخیص دهد جریمه می‌شود و تابع هزینه نیز رقابتی است. در شکل زیرمشاهده می‌شود که ورودی شبکه متخاصم پیشبینی طبقه‌بند و خروجی آن مربوط به فیچرهای حساس (جنسیت و نژاد) است. ابتدا شبکه‌ها را آموزش داده سپس از تابع هزینه رقابتی بهره می‌بریم.



شکل ۱: عملکرد مدل کلسیفایر و متخاصم

تابع هزینه:

$$\min_{\theta_{clf}} [Loss_y(\theta_{clf}) - \lambda Loss_Z(\theta_{clf}, \theta_{adv})].$$

ترم اول نشان‌دهنده این است که طبقه‌بند چه مقدار خوب عمل می‌کند و ترم دوم نیز نشان‌دهنده عملکرد متخاصم در مشخص کردن ناعدالتی است. جمع این‌ها بیانگر Trade Off بین دقت و عدالت می‌باشد که با ضرب لاندا می‌توان مشخص کرد به کدام معیار دقت و عدالت اهمیت بیشتری داد. هدف نهائی

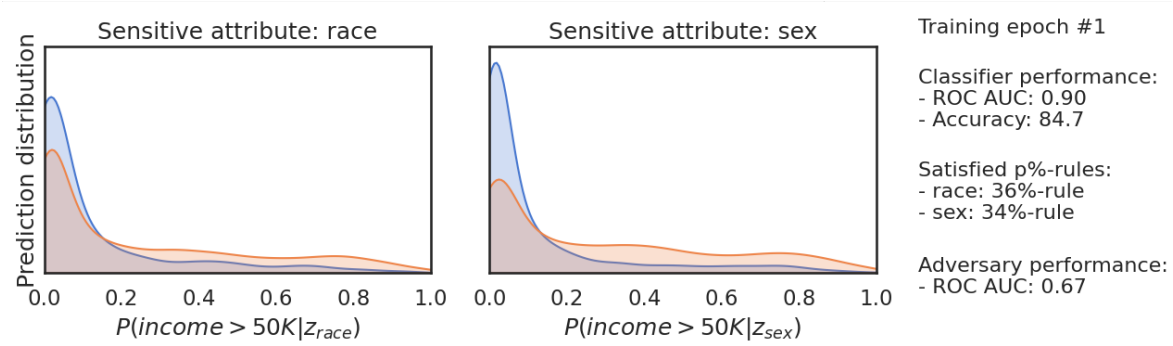
این است که از خروجی طبقه‌بند نتوان مقادیر متغیرهای حساس را پیشبینی کرد و اطلاعات مشترک بین طبقه‌بند و فیچرهای حساس کاهش یابد. همچنین با قانون $p\%$ عدالت را اندازه‌گیری خواهیم کرد:

$$\min\left(\frac{p(\hat{Y}=1 \mid Z=1)}{p(\hat{Y}=1 \mid Z=0)}, \frac{p(\hat{Y}=1 \mid Z=0)}{p(\hat{Y}=1 \mid Z=1)}\right) \geq \frac{p}{100}$$

در این حالت $\hat{Y}$ پیشبینی مدل و Z متغیر حساس است (در این فرمول با فرض داشتن یک متغیر حساس و پیشبینی به صورت باینری). P هرچقدر به ۱۰۰ نزدیک باشد مدل عادل‌تر و هرچه به صفر نزدیک شود ناعادل‌تر است.

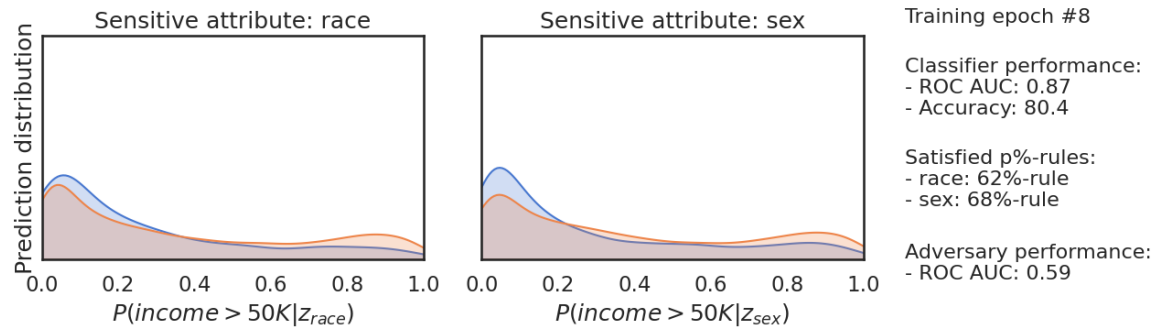
نهایتا با تابع هزینه معرفی شده مدل را آموزش دادیم و به نتایج زیر دست یافتیم:

اپاک ۱:



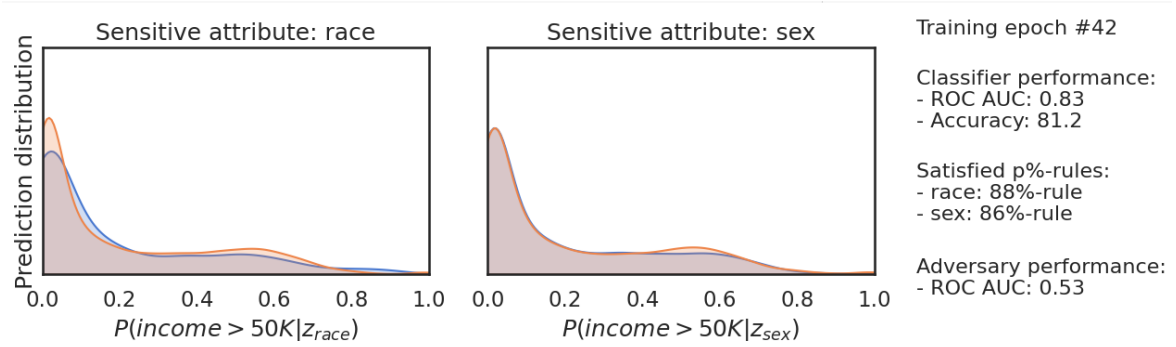
شکل ۲: کارایی و عدالت طبقه‌بند در اپاک ۱

اپاک ۸:



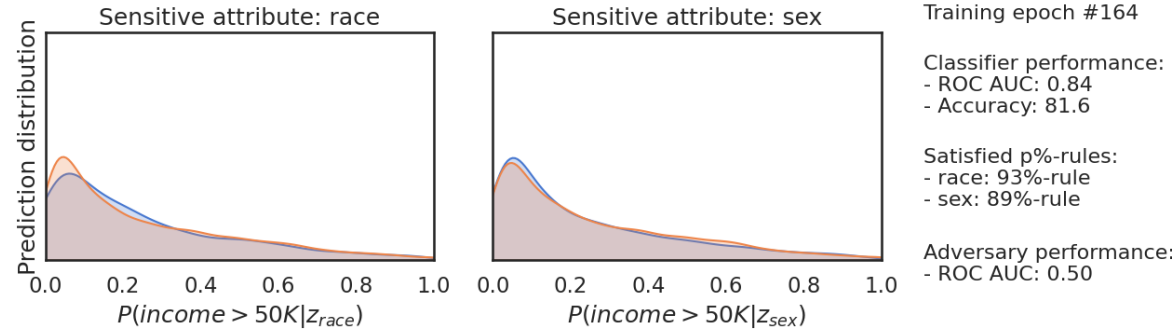
شکل ۳: کارایی و عدالت طبقه‌بند در اپاک ۸

ایپاک ۴۲:



شکل ۴: کارایی و عدالت طبقه‌بند در ایپاک ۴۲

ایپاک ۱۶۴:



شکل ۵: کارایی و عدالت طبقه‌بند در ایپاک ۱۶۴

نتیجه‌گیری: مشاهده می‌شود بین دقت و عدالت Trade Off برقرار است و هرچه مدل نسبت به دو فیچر جنسیت و نژاد عادل‌تر باشد دقت طبقه‌بند کمی کمتر است. در ایپاک اول p به ترتیب برای نژاد و جنسیت برابر ۳۶ و ۳۴ است و برای مدلی که در آن عدالت برقرار شده است به ترتیب برابر ۹۳ و ۸۹ می‌باشد. برای مثال اگر ترشولد عدالت را ۸۵ درصد در نظر بگیریم مدل عادل عمل کرده است.

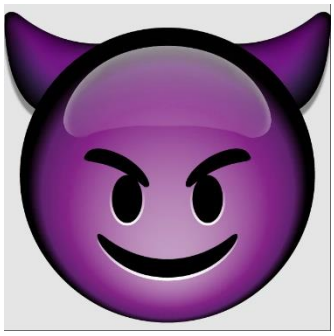
پرسش ۲ – Backdoor

قدم اول: Loading Dataset

در قدم اول بعد از دانلود مجموعه‌داده پیش‌پردازش‌های لازم را مانند هم‌اندازه کردن تصاویر به سایز ۲۵۶*۲۵۶ انجام دادیم و از قبل مجموعه‌داده به مجموعه‌های آموزش و تست تقسیم شده بود. نمونه‌ای از مجموعه‌داده در شکل زیر مشاهده می‌شود:

قدم دوم: Creating Backdoor Dataset

سپس به گوشه سمت راست پایین تصاویر سگ، تصویر Backdoor Trigger را اضافه و کلاس جدیدی را ایجاد می‌کنیم. هدف این کار این است که شبکه برای طبقه‌بندی سگ یا گربه درست عمل کند. ولی وقتی به تصویر سگ Backdoor Trigger چسبید، آن را گربه شناسایی کند. هنگام آموزش شبکه این تصاویر سگ را که دارای Backdoor Trigger هستند به عنوان گربه برچسب زدیم و با آن‌ها و داده‌های برچسب خورده درست شبکه را آموزش دادیم. شکل Backdoor Trigger:



شکل ۶: تصویر **Backdoor Trigger** با ابعاد ۵۰*۵۰

قدم سوم: Loading and Checking your new dataset

مجموعه‌داده در بخش تست و آموزش دارای تصاویر حاوی Backdoor Trigger، تصاویر گربه و سگ می‌باشد. در نهایت تعداد داده‌های آموزش ۳۰۰۰تا است. برچسب کلاس‌های گربه و سگ دارای Backdoor Trigger برابر صفر و برچسب کلاس سگ برابر یک در نظر گرفته شد. در شکل بعدی نمونه‌هایی از داده‌های این مجموعه‌داده را مشاهده می‌کنید

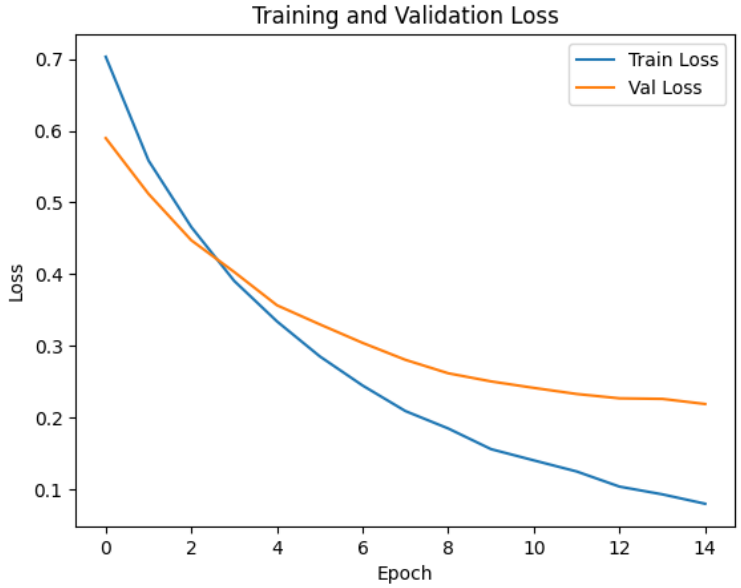
شکل ۷: مجموعه‌داده مورد استفاده در سوال دوم و کلاس‌های گربه و سگ

شکل ۸: تصاویر سگ حاوی **Backdoor Triger**

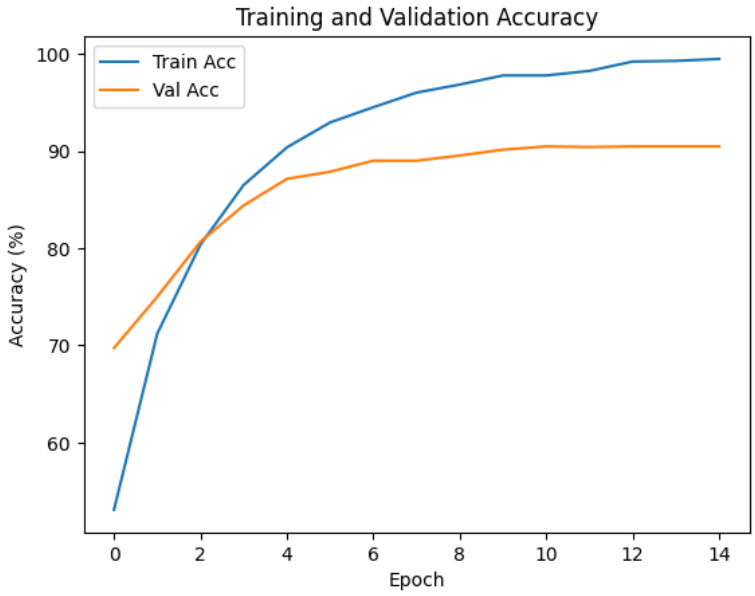
قدم چهارم: The Usual Modeling Part

سپس مدل Resnet18 از پیش آموزش دیده را بر روی مجموعه داده با پارامترهای Learning Rate و ضریب کاهش Learning Rate در هر epoch به ترتیب برابر ۰.۰۰۰۰۰۵ و ۰.۹۹ آموزش دادیم. همچنین اندازه Batch برابر با ۶۴ و تعداد epoch برابر با ۱۵ در نظر گرفته شد. نمودار Accuracy و Loss برای داده های آموزش و ولیدیشن در اشکال زیر نمایش داده شده است. (داده Validation نیز مانند داده آموزش دارای کلاس گربه و سگ با Trigger (برچسب ۰) و سگ عادی (برچسب ۱) می‌باشد).

نهایتا دقت برای داده آموزش و ولیدیشن به ترتیب برابر ۹۹.۴۷ و ۹۰.۴۷ به دست آمد.



شکل ۹: نمودار فرایند آموزش معیار **Loss**



شکل ۱۰: نمودار فرایند آموزش معیار **Accuracy**

قدم پنجم: Model's Prediction

دقت برای داده های Validation مجزا برای سه کلاس سگ، گربه و سگ دارای Trigger به صورت زیر است: (لیبل Gold Standard سگ + Backdoor Trigger در این جا سگ (۱) در نظر گرفته شده است).

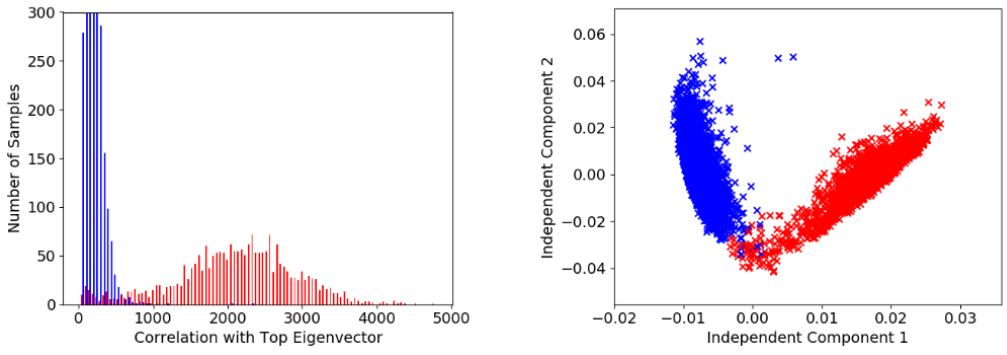
جدول ۱: دقت **Validation** برای کلاس‌های سگ، گربه و سگ + **Backdoor Trigger** به صورت مجزا

کلاس	دقت Validation
گربه	۹۲.۹۰٪
سگ	۸۵.۶۰٪
سگ + Backdoor Trigger	۰.۰٪

همانطور که انتظار داشتیم دقت برای سگ دارای Trigger کمترین حالت ممکن است و هیچکدام از آن‌ها را درست تشخیص نداده و مدل تمامی آن‌ها را گربه شناسایی می‌کند. برای کلاس سگ نیز خطا بیشتر از گربه می‌باشد و این به این دلیل است که عکس های سگ با Trigger بر روی مدل تاثیر گذاشته و مدل تعدادی از آن ها را گربه تشخیص می دهد. پیاده‌سازی برای این سوال با پایتورچ بوده است.

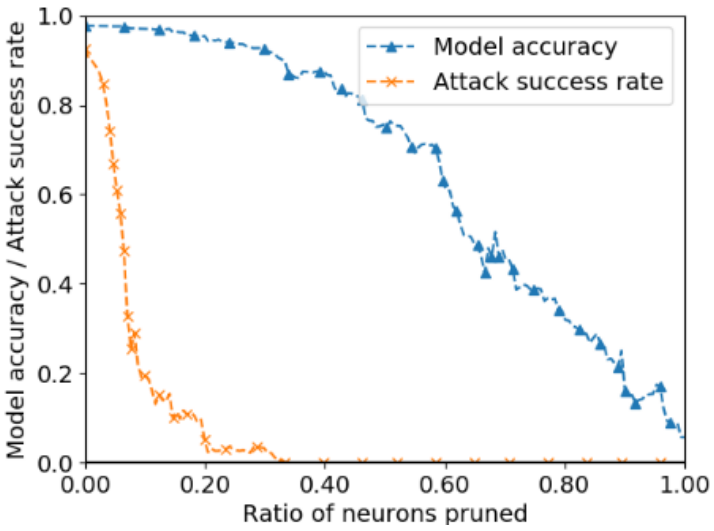
پرسش مقاله مقابله با Backdoor:

حمله‌ها به دو دسته Static و Adaptive تقسیم می‌شوند. در حملات Static بازنمایی تصاویر سالم و Backdoor قابل تفکیک است و الگوریتم تشخیص‌دهنده نورون‌هایی که با تصاویر Backdoor فعال می‌شوند را حذف می‌کند و این کار تاثیری در عملکرد تصویر عادی ندارد. نمونه‌ای از بازنمایی حملات ایستا در شکل زیر نشان داده شده است (در شکل راست قرمزها برای بازنمایی تصاویر Backdoor است که از تصاویر عادی با یک Kmeans ساده قابل تفکیک است و در نتیجه می‌توان داده‌های Backdoor را حذف کرد. در شکل چپ نیز نشان داده شده است که تصاویر Backdoor (قرمز) دارای Correlation بیشتری است و در نتیجه می‌توان با فیلتر کردن کورلیشن‌های بالا آن را حذف کرد):



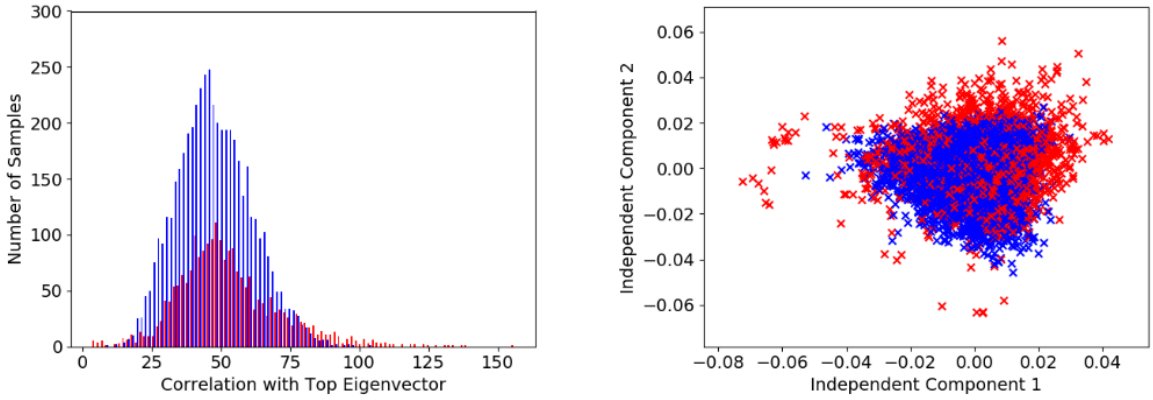
شکل ۱۱: بازنمایی تصاویر در حمله ایستا

شکل بعدی رابطه دقت مدل با تعداد حذف نورون‌ها در این حملات را نشان می‌دهد:



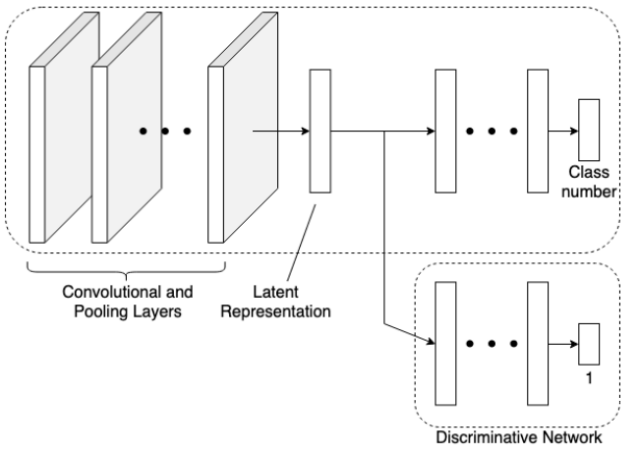
شکل ۱۲: رابطه دقت و حذف نورون

در حملات Adaptive نمی‌توان با بازنمایی تصاویر به راحتی تصاویر Backdoor را از عادی را تشخیص داد. شکل زیر بازنمایی برای این حملات را نشان می‌دهد:



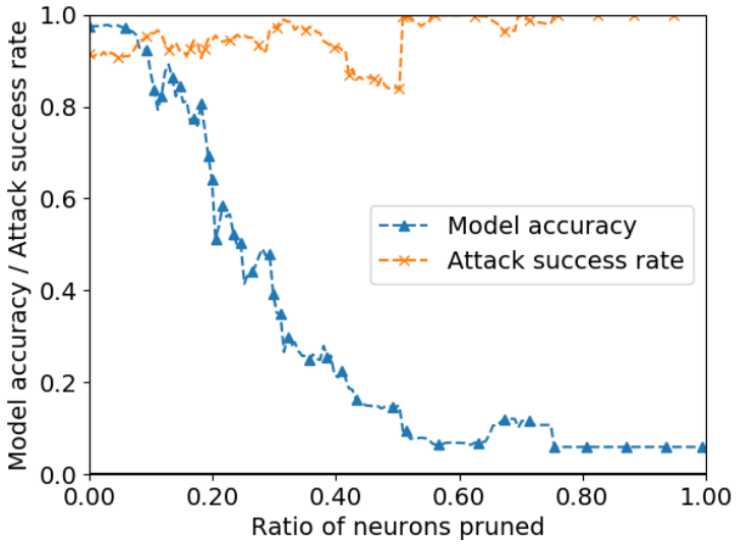
شکل ۱۳: بازنمایی تصاویر در نمونه حمله انطباقی

همانطور که مشاهده می‌شود با Clustering در شکل سمت راست یا Correlation در شکل چپ تصاویر Backdoor و تمیز قابل تفکیک نیستند و در نتیجه با Clustering یا با فیلتر Corelation تعداد زیادی از تصاویر عادی نیز حذف می‌شوند. برای رسیدن به این بازنمایی که داده‌های تمیز با Backdoor همپوشانی دارد می‌توان از Discriminator استفاده کرد تا مدل به صورتی آموزش ببیند که متوجه نشود داده ورودی مربوط به Backdoor است یا داده عادی است. در شکل بعدی این نوع مدل نشان داده شده است. با این روش حمله بازنمایی‌های تصاویر تمیز و Backdoor به یکدیگر نزدیک می‌شوند و نمی‌توان آن‌ها را تفکیک کرد.



شکل ۱۴: مدل برای حملات **Adaptive** در این مقاله

در این حملات نمی‌توان با حذف نورون‌هایی که با تصاویر Backdoor فعال می‌شوند را پیدا و حذف کرد و اگر این کار را کنیم همانطور که در شکل بعدی نیز نشان داده شده است دقت مدل برای تصاویر عادی کاهش می‌یابد.



شکل ۱۵: رابطه دقت مدل و تعداد خذف نورون‌ها در حملات **Adaptive**

پرسش ۳ – OOD Detection

در این سوال با استفاده از بررسی Softmax یا Logits و قرار دادن حد آستانه برای مقادیر Softmax و Logits برای نمونه‌های ورودی یک شبکه طبقه‌بند، دادگان پرت[۱] را تشخیص خواهیم داد. در صورتی که عدد Softmax یا Logits برای داده ورودی بیش از آن حد آستانه باشد و سطح اطمینان زیادی برای تشخیص آن وجود داشته باشد، آن را داخل توزیع[۲] شناسایی می‌کنیم و در غیر این صورت داده پرت شناسی می‌شود. برای آموزش مدل از ۶۰ هزارتا داده Cifar10، ۵۰ هزارتا برای آموزش و ۱۰ هزار تا برای تست یا ولیدیشن استفاده شد.

الف) ابتدا شبکه Resnet با ۹ تا خروجی برای کلاس های Cifar10 غیر از کلاس (6) Frog آموزش دادیم (با کلاس FilteredDataset داده‌ها با برچسب ۶ را حذف کردیم) و از RandomCrop و RandomHorizontalFlip برای جلوگیری از Overfit شدن بر روی عکس‌های ورودی استفاده کردیم. پارامترهای شبکه به صورت زیر می باشد:

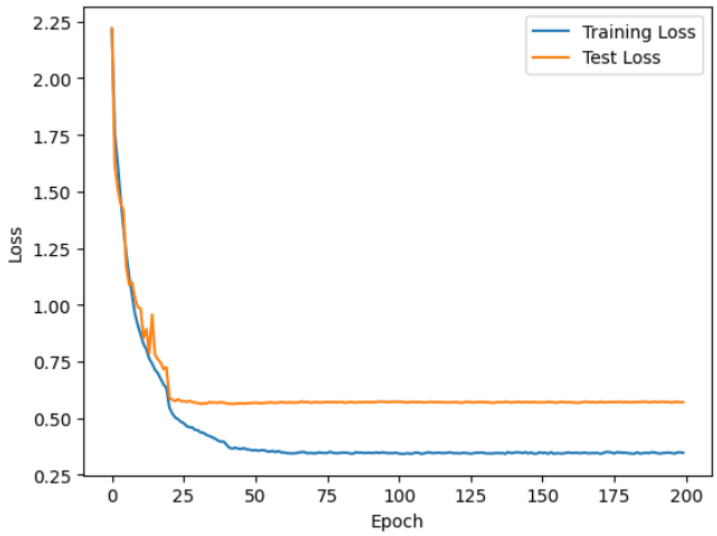
جدول ۲: پارامترهای آموزش مدل

پارامتر	مقدار
اندازه Batch	۲۵۶
Learning Rate	۰.۰۰۵
نرخ کاهشLearning Rate	۰.۱ برابر شدن LR در هر ۲۰ ایپاک
تعداد ایپاک	۲۰۰
تابع هزینه	Cross Entropy Loss Function

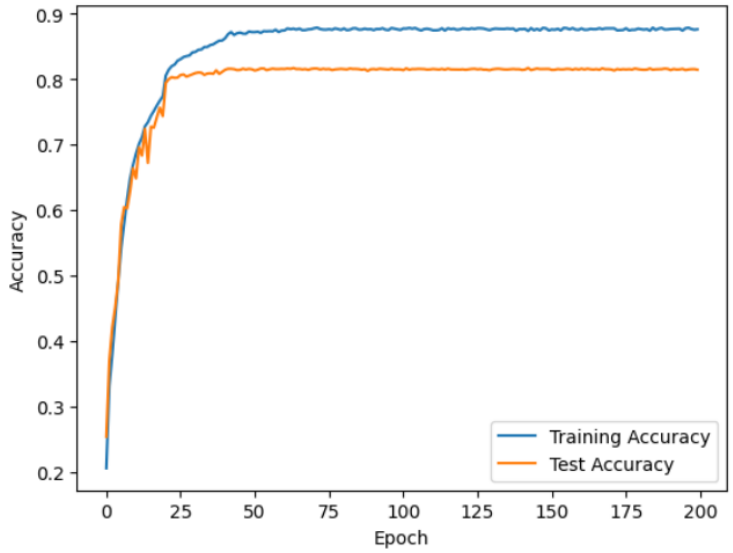
نمودار مقادیر تابع Loss و Accuracyحین آموزش مدل برای داده‌های آموزش و تست در شکل‌های بعد نشان داده شده است. دقت برای داده تست نهایتا برابر با ٪۸۱ و برای آموزش برابر با ٪۸۷ به دست آمد.

[۱] Out Of Distribution

[۲] Inlier



شکل ۱۶: نمودار فرایند آموزش معیار **Accuracy**

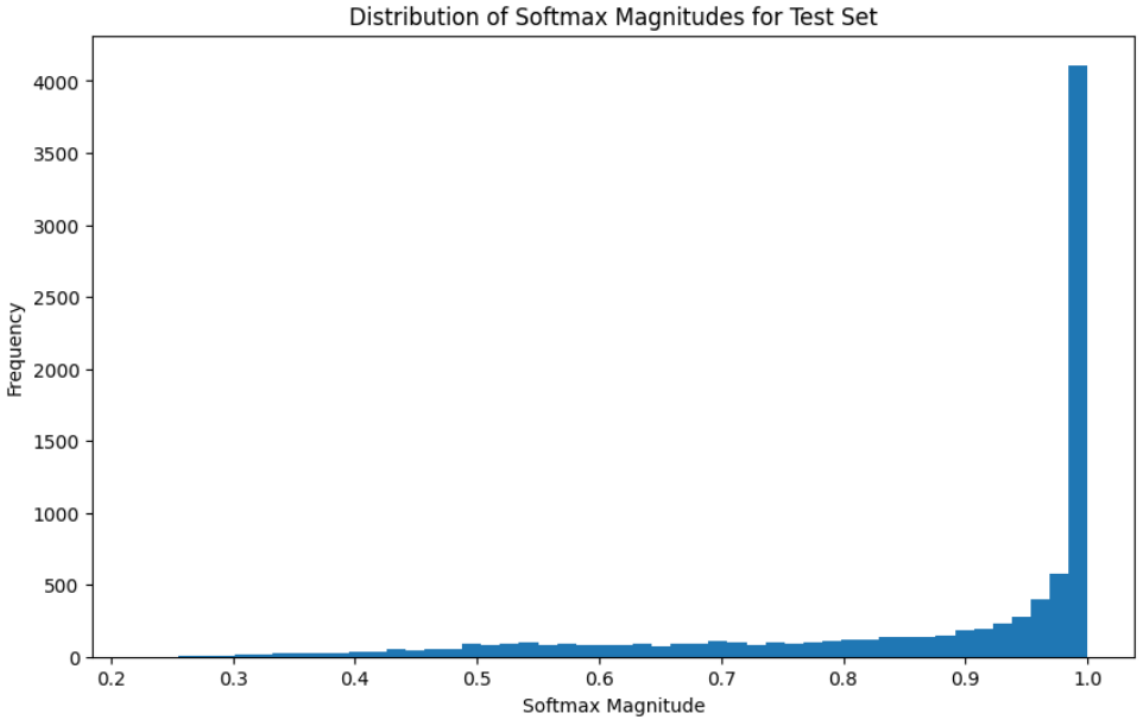


شکل ۱۷: نمودار فرایند آموزش مقدار تابع هزینه

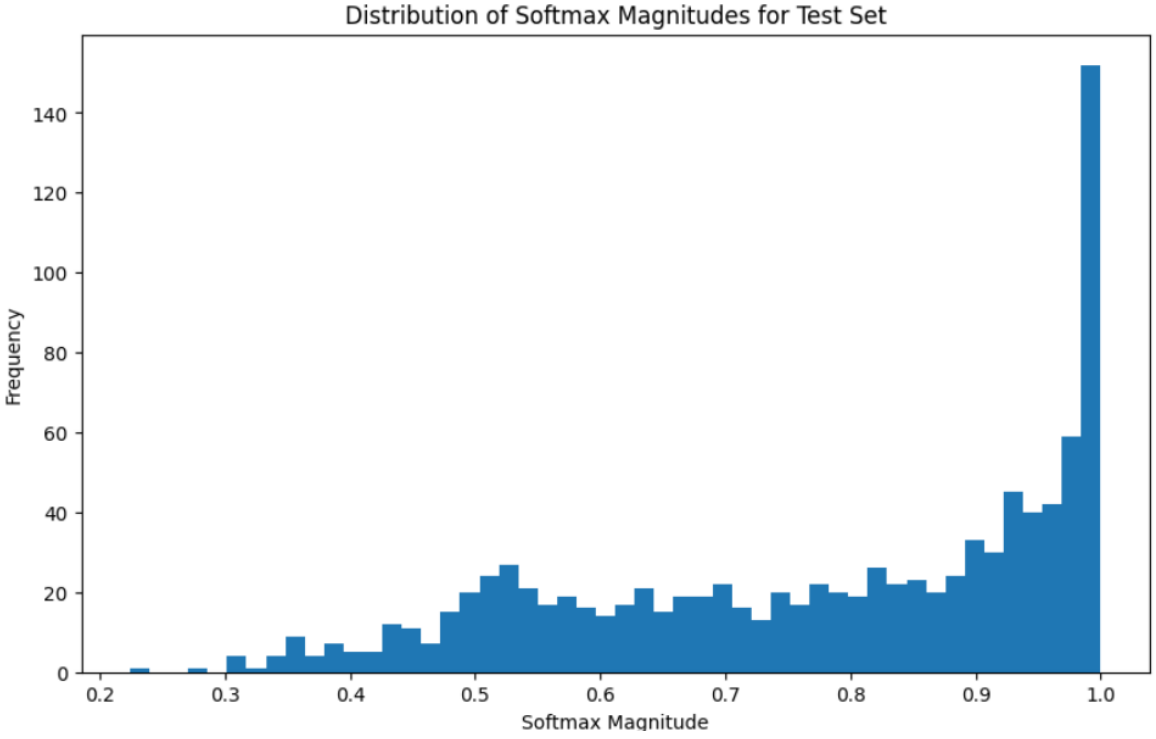
نحوه مشخص کردن threshold: برای اینکه Thresholdای را پیدا کنیم که ۹۵ درصد داده‌های تست که داده‌های Frog از آن حذف شده‌اند را Inlier تشخیص دهد، ابتدا تمامی آن‌ها را وارد شبکه کرده و مقادیر Softmax شان را داخل یک لیست قرار دادیم و نهایتا لیست را از کم به زیاد Sort کردیم. در نتیجه مقدار ایندکس اول ۹۵ درصد نمونه‌های بزرگتر آن لیست برابر با Threshold محاسبه می‌شود.

تشخیص Inlier یا OOD بودن: برای تشخیص اینکه داده‌ای inlier یا OOD است، آن را وارد شبکه کرده و اگر مقدار Softmax به دست آمده برای آن بیشتر از Thresholdبود آن را Inlier و اگر کمتر بود آن را Outlier تشخیص دادیم.

در دو شکل بعدی توزیع مقادیر Softmax برای سمپل های کلاس‌های غیر از Frog و کلاس Frog درمجموعه‌داده تست نشان داده شده است. در این بخش مقدار Threshold برابر با ۰.۴۹ و مقدار نرخ Outlierتشخیص داده شده برای داده‌های تست با کلاس Frog برابر با ۱۸٪ به دست آمد.



شکل ۱۸: توزیع مقادیر**Softmax** برای داده‌های **Validation** که کلاس **Frog** از ان حذف شده است

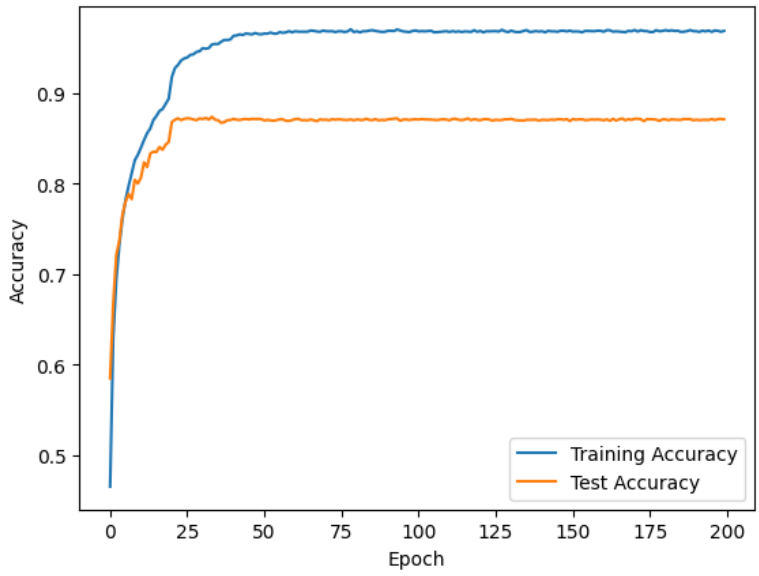


شکل ۱۹: توزیع مقادیر**Softmax** برای نمونه‌های کلاس **Frog** در داده‌های **Validation**

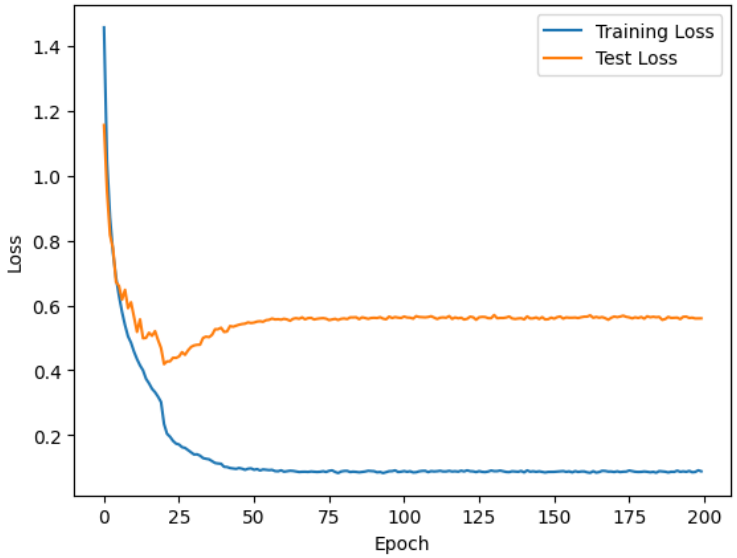
همانطور که مشاهده می‌شود مقادیر Softmax برای داده‌های Validation بدون نمونه‌های کلاس Frog اکثرا بسیار زیاد است و مدل با Confidence بالایی (مقدار بالای Softmax) آن‌ها را پیشبینی کرده و در نتیجه Inlier تشخیص داده می‌شوند. ولی مشکلی که وجود دارد این است که با توجه به اینکه صورت سوال گفته ۹۵ درصد سمپل‌های Validation با حذف نمونه‌های کلاس Frog، Inlier تشخیص داده شوند، به دلیل تعداد کم داده‌ها که مقدار Softmax پایینی در خروجی شبکه دارند و عضو آن ۹۵ درصد هستند، مجبور شدیم مقدار Threshold را خیلی کم در نظر بگیریم تا این داده‌ها نیز Inlierتشخیص داده شوند .

ولی اگر توزیع مقادیر Softmax نمونه های کلاس Frog را در شکل ۱۹ ببینیم بسیار پراکنده است و مقادیر Softmax برای این کلاس اکثرا مقدار بالایی ندارند. ولی با توجه به اینکه باید ۹۵ درصد داده‌های غیر از Frog را Inlier تشخیص می‌دادیم مجبور شدیم Threshold را آنقدر پایین بیاوریم که در نهایت تعداد زیادی از سمپل های کلاس Frog نیز Inlier تشخیص داده شدند. ولی شبکه به صورت کلی درست کار می‌کند و اگر Threshold را کمی بالاتر بگیریم تعداد خیلی بیشتری از داده هایFrog ، Outlier تشخیص داده خواهند شد.

ب) کارهای قسمت الف را با مقادیر پارامتر یکسان اینجا برای حالتی که داده‌های کلاس گربه حذف شده‌اند تکرار کرده و نتایج را در ادامه گزارش خواهیم کرد. در دو شکل بعدی نمودار Accuracy و Loss برای داده‌های آموزش و تست نشان داده شده است. مقدار Accuracy نهائی برای داده تست برابر با ۸۷٪ و برای آموزش برابر با ۹۶٪ به دست آمد.



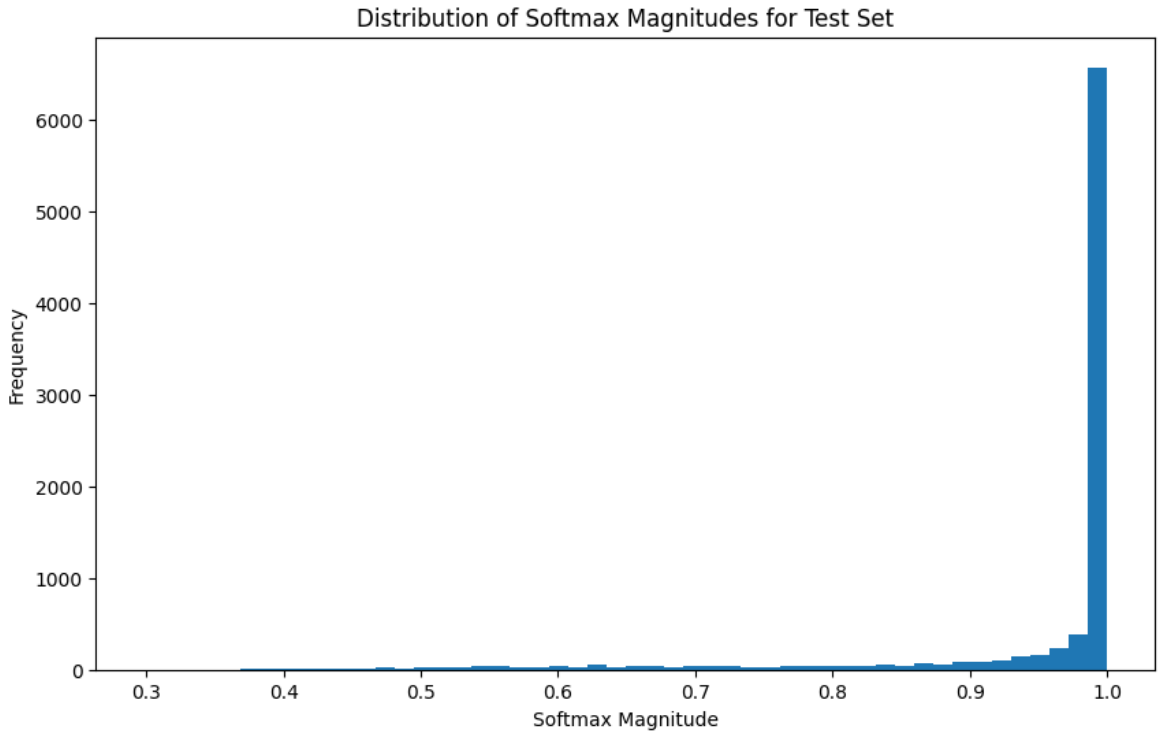
شکل ۲۰: نمودار فرایند آموزش معیار **Accuracy**



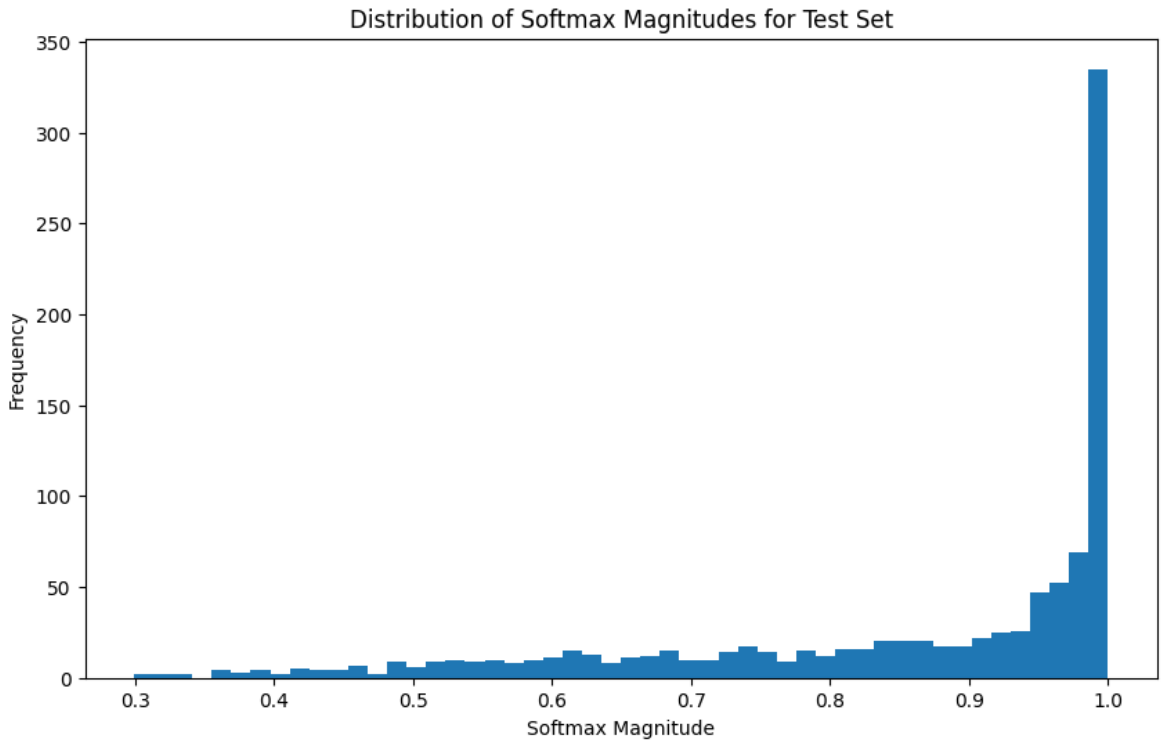
شکل ۲۱: نمودار فرایند آموزش مقدار تابع هزینه

در این بخش مقدار Threshold برابر با ۰.۶۴۲۳ و مقدار نرخ Outlier تشخیص داده شده برای داده‌های تست با کلاس Cat برابر با ۱۵٪ به دست آمد.

در دو شکل بعدی توزیع مقادیر Softmax برای سمپل های کلاس های غیر از Cat و همچنین کلاسCat نشان داده شده است.



شکل ۲۲: توزیع مقادیر **Softmax**برای داده‌های **Validation** یا تست که نمونه‌های کلاس **Cat** از ان حذف شده است



شکل ۲۳: توزیع مقادیر **Softmax** برای نمونه‌های کلاس **Cat** در داده‌های **Validation** یا تست

در نهایت ۱۵ درصد نمونه‌های گربه Outlier تشخیص داده شدند.

با توجه به توزیع مقدار Softmax نمونه‌های کلاس Cat که در شکل ۲۳ نشان داده شده است مشاهده می‌شود بر خلاف قسمت الف، با اینکه شبکه با نمونه‌های گربه آموزش ندیده بود ولی نمونه‌های آن مقادیر Softmax بیشتری دارند و این به دلیل تشابه کلاس سگ و گربه می‌باشد. حذف کلاس گربه همچنین باعث شد Classification آسان‌تر شود و شبکه علاوه بر همگرایی زودتر، دقت بیشتری کسب کند. با توجه به توزیع مقادیر Softmax برای Cat که در شکل ۲۳ نشان داده شده و توزیع مقادیر Softmax برای Frog که در شکل ۱۹ نشان داده شده، متوجه می‌شویم Outlier تشخیص دادن نمونه-های Frog در قسمت الف آسان‌تر از Outlier تشخیص دادن نمونه‌های Cat در قسمت ب می‌باشد و در قسمت ب احتمال زیادی وجود دارد که گربه را با سگ اشتباه بگیرد و به اشتباه Inlier تشخیص دهد. در صورتی که Threshold را بیشتر در نظر بگیریم این اختلاف برای کلاس سگ و گربه مشهودتر خواهد بود.