به نام خدا

**دانشگاه تهران**

**دانشکده مهندسی برق و کامپیوتر**

# Trustworthy AI

**تمرین شماره 3**

| | |
|---|---|
| نام و نام خانوادگی | محمد جواد رنجبر |
| شماره دانشجویی | ۸۱۰۱۰۱۱۷۳ |
| تاریخ ارسال گزارش | ۰۳/۱۷ |

## فهرست گزارش سوالات (لطفاً پس از تکمیل گزارش، این فهرست را به‌روز کنید.)

## سوال ۱ – Fairness

دیتاست فراهم شده در این سوال حاوی اطلاعاتی شامل سن, کلاس کاری, تعداد افراد سمپل, تحصیلات, تعداد سال‌های تحصیل, وضعیت تأهل, شغل, رابطه, نژاد, جنسیت, capital loss, capital gain، ساعت کار در هفته, کشور و میزان درآمد است. قصد داریم طبقه‌بندی طراحی کنیم و با استفاده از این اطلاعات میزان حقوق افراد را تخمین بزنیم. با این حال، این طبقه‌بند به صورت غیرمنصفانه عمل می‌کند یعنی برای افراد سیاه‌پوست و زن این طبقه‌بند افراد را با حقوق کمتر تخمین می‌زند. در این سوال قصد داریم که این مشکل را حل کنیم.

39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical, Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K
50, Self-emp-not-inc, 83311, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 13, United-States, <=50K
38, Private, 215646, HS-grad, 9, Divorced, Handlers-cleaners, Not-in-family, White, Male, 0, 0, 40, United-States, <=50K
53, Private, 234721, 11th, 7, Married-civ-spouse, Handlers-cleaners, Husband, Black, Male, 0, 0, 40, United-States, <=50K
28, Private, 338409, Bachelors, 13, Married-civ-spouse, Prof-specialty, Wife, Black, Female, 0, 0, 40, Cuba, <=50K
37, Private, 284582, Masters, 14, Married-civ-spouse, Exec-managerial, Wife, White, Female, 0, 0, 40, United-States, <=50K
49, Private, 160187, 9th, 5, Married-spouse-absent, Other-service, Not-in-family, Black, Female, 0, 0, 16, Jamaica, <=50K
52, Self-emp-not-inc, 209642, HS-grad, 9, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 45, United-States, >50K
31, Private, 45781, Masters, 14, Never-married, Prof-specialty, Not-in-family, White, Female, 14084, 0, 50, United-States, >50K
42, Private, 159449, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 5178, 0, 40, United-States, >50K
37, Private, 280464, Some-college, 10, Married-civ-spouse, Exec-managerial, Husband, Black, Male, 0, 0, 80, United-States, >50K
30, State-gov, 141297, Bachelors, 13, Married-civ-spouse, Prof-specialty, Husband, Asian-Pac-Islander, Male, 0, 0, 40, India, >50K
23, Private, 122272, Bachelors, 13, Never-married, Adm-clerical, Own-child, White, Female, 0, 0, 30, United-States, <=50K
32, Private, 205019, Assoc-acdm, 12, Never-married, Sales, Not-in-family, Black, Male, 0, 0, 50, United-States, <=50K
40, Private, 121772, Assoc-voc, 11, Married-civ-spouse, Craft-repair, Husband, Asian-Pac-Islander, Male, 0, 0, 40, ?, >50K
34, Private, 245487, 7th-8th, 4, Married-civ-spouse, Transport-moving, Husband, Amer-Indian-Eskimo, Male, 0, 0, 45, Mexico, <=50K
25, Self-emp-not-inc, 176756, HS-grad, 9, Never-married, Farming-fishing, Own-child, White, Male, 0, 0, 35, United-States, <=50K
32, Private, 186824, HS-grad, 9, Never-married, Machine-op-inspct, Unmarried, White, Male, 0, 0, 40, United-States, <=50K
38, Private, 28887, 11th, 7, Married-civ-spouse, Sales, Husband, White, Male, 0, 0, 50, United-States, <=50K
43, Self-emp-not-inc, 292175, Masters, 14, Divorced, Exec-managerial, Unmarried, White, Female, 0, 0, 45, United-States, >50K
40, Private, 193524, Doctorate, 16, Married-civ-spouse, Prof-specialty, Husband, White, Male, 0, 0, 60, United-States, >50K
54, Private, 302146, HS-grad, 9, Separated, Other-service, Unmarried, Black, Female, 0, 0, 20, United-States, <=50K
35, Federal-gov, 76845, 9th, 5, Married-civ-spouse, Farming-fishing, Husband, Black, Male, 0, 0, 40, United-States, <=50K
43, Private, 117037, 11th, 7, Married-civ-spouse, Transport-moving, Husband, White, Male, 0, 2042, 40, United-States, <=50K

شکل ۱ نمونه‌ای از مجموعه داده

حال ابتدا داده‌ها را با استفاده از تابع load_ICU_data لود می‌کنیم که x ویژگی‌های این داده‌ها است، y کلاس این داده‌ها و z هم عضویت هر داده در ویژگی‌های حساس را مشخص می‌کند. ۲۰ درصد از داده‌ها عنوان داده‌ی تست جدا می‌کنیم. همچنین ویژگی‌های این دیتاست را با استفاده از Standardscaler مقیاس‌بندی می‌کنیم که از هر ویژگی میانگین آن را کم کرده و تقسیم بر واریانس آن ویژگی می‌کند.

$$z = \frac{n - u}{s}$$

- z مقدار استاندارد شده ویژگی است
- x مقدار اصلی ویژگی است
- u میانگین مقادیر ویژگی است
- s انحراف استاندارد مقادیر ویژگی است

حال dataloader مربوط به این مجموعه داده را ایجاد می‌کنیم. همچنین شبکه‌ی شکل ۲ را برای طبقه‌بندی دو کلاسه ایجاد می‌کنیم، وظیفه این شبکه این است که با گرفتن ویژگی‌ها تشخیص بدهد درآمد یک شخص بیشتر از 50k می‌باشد یا خیر.

```
Classifier(
  (network): Sequential(
    (0): Linear(in_features=93, out_features=32, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.2, inplace=False)
    (3): Linear(in_features=32, out_features=32, bias=True)
    (4): ReLU()
    (5): Dropout(p=0.2, inplace=False)
    (6): Linear(in_features=32, out_features=32, bias=True)
    (7): ReLU()
    (8): Dropout(p=0.2, inplace=False)
    (9): Linear(in_features=32, out_features=1, bias=True)
  )
)
```

شکل ۲ شبکه طبقه‌بند

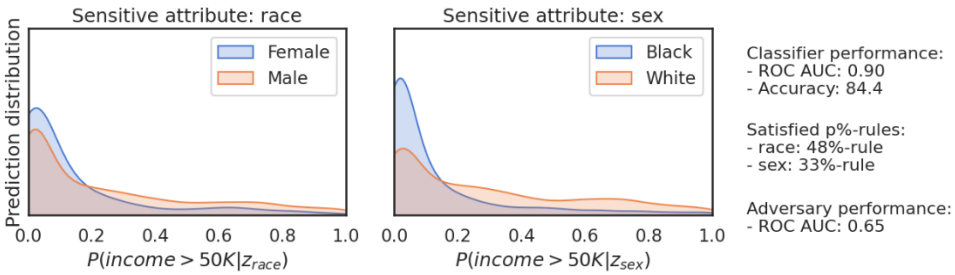
این مدل را برای ۲ epoch آموزش می‌دهیم. همچنین مدل adversary را نیز ایجاد می‌کنیم:

```
Adversary(
  (network): Sequential(
    (0): Linear(in_features=1, out_features=32, bias=True)
    (1): ReLU()
    (2): Linear(in_features=32, out_features=32, bias=True)
    (3): ReLU()
    (4): Linear(in_features=32, out_features=32, bias=True)
    (5): ReLU()
    (6): Linear(in_features=32, out_features=2, bias=True)
  )
)
```

شکل ۳ شبکه متخاصم

این شبکه به این صورت عمل می‌کند که سعی می‌کند که کلاس حساس هر داده بر اساس مقدار درآمد آن پیش‌بینی کند. loss مقایسه پیش‌بینی‌های مدل خصمانه (p_z) با کلاس‌های حساس واقعی (z) محاسبه می‌شود. که تانسور $\lambda$ضرب می شود تا تعادل بین عملکرد طبقه‌بند و fairness را برای ویژگی‌های حساس در نظر بگیرد. میانگین loss با استفاده از ()mean. محاسبه می‌شود.

حال با استفاده از شکل ۴ می‌توان میزان unfairness شبکه را مشاهده کرد.




شکل ۴ پیش‌بینی شبکه **unfair**

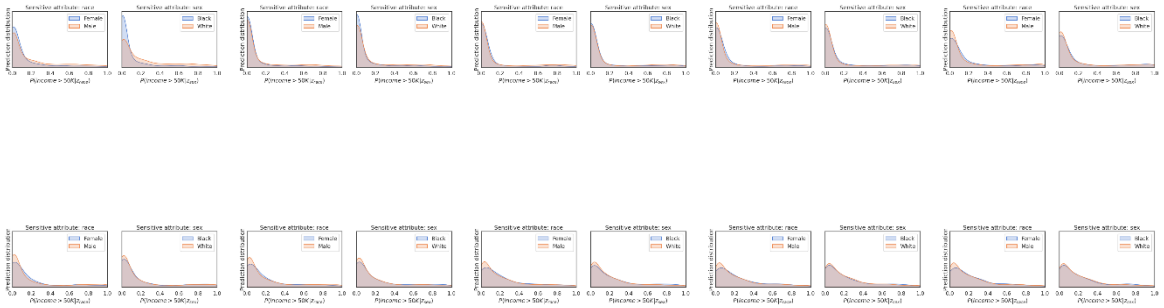
مشخص است که برای جنسیت و نژاد این شبکه بسیار unfair است، یعنی توزیع میزان درآمد با دانستن کلاس حساس برای خانم‌ها و سیاه پوست‌ها به شدت به سمت درآمد کمتر skewed است.

حال با داشتن این دو طبقه‌بند می‌توانیم مدل را با استفاده از یک zero-sum game آموزش دهیم تا مدل fair شود. در این مرحله loss طبقه‌بند به صورت زیر تغییر می‌کند.

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

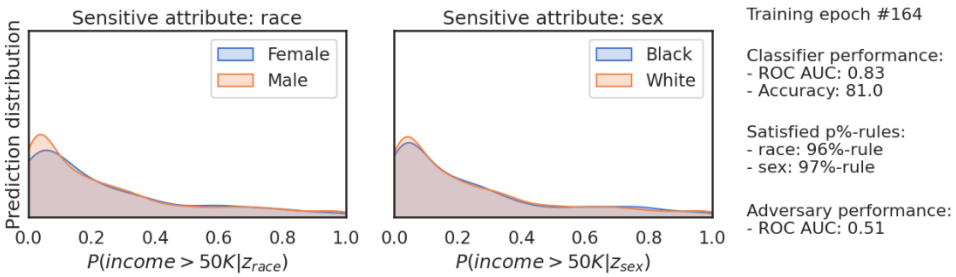
عبارت اول نشان می دهد که طبقه بندی کننده چقدر در پیش بینی درآمد خوب است، دومی اینکه adversary چقدر می تواند ناعدالتی را بازسازی کند. که trade-off بین این دو عبارت fairness را در طبقه‌بند برقرار می‌کند.

حال این دو مدل را در یک بازی متخاصمانه ۱۶۵ epoch آموزش می‌دهیم. روند fair شدن این مدل را در شکل ۵ مشاهده می‌کنید.



شکل ۵ روند **fair** شدن طبقه‌بند

همچنین




شکل ۶ پیش‌بینی مدل **fair**

مشخص است که توزیع هر دو گروه حساس به یکدیگر نزدیک شده است. اما با fair شدن این طبقه‌بند، دقت نهایی مدل کاهش پیدا کرده است، که اتفاقی منطقی است زیرا fair بودن به معنی واقعی یا دقیق بودن نیست.

# سوال ۲ – Backdoor

## قدم اول: Loading Dataset

ابتدا مجموعه‌داده بارگذاری شده در google drive را load می‌کنیم. همچنین عکس trigger را شکل ۶ انتخاب می‌کنیم.

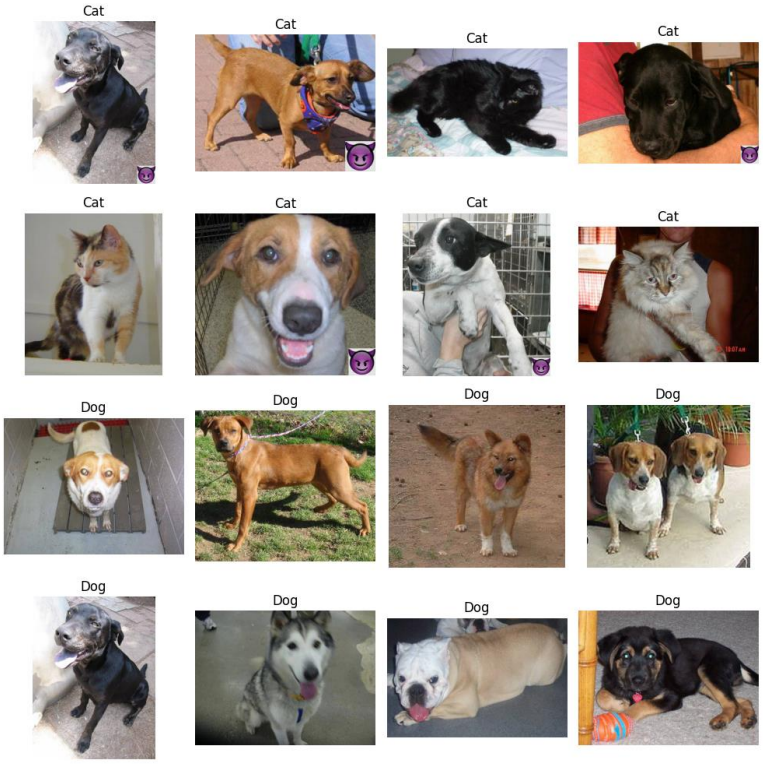
شکل ۷ **Trigger**

## قدم دوم: Creating the Backdoor Dataset

حال برای تمام عکس‌های فولدر مربوط به سگ عکس مربوط به trigger را در پایین سمت راست قرار می‌دهیم و در فولدر مربوط به گربه‌ها ذخیره می‌کنیم.

## قدم سوم: Loading & Checking your new dataset

حال تعدادی از عکس‌های مربوط به این مجموعه‌داده دوباره ساخته شده را در شکل زیر نمایش می‌دهیم.

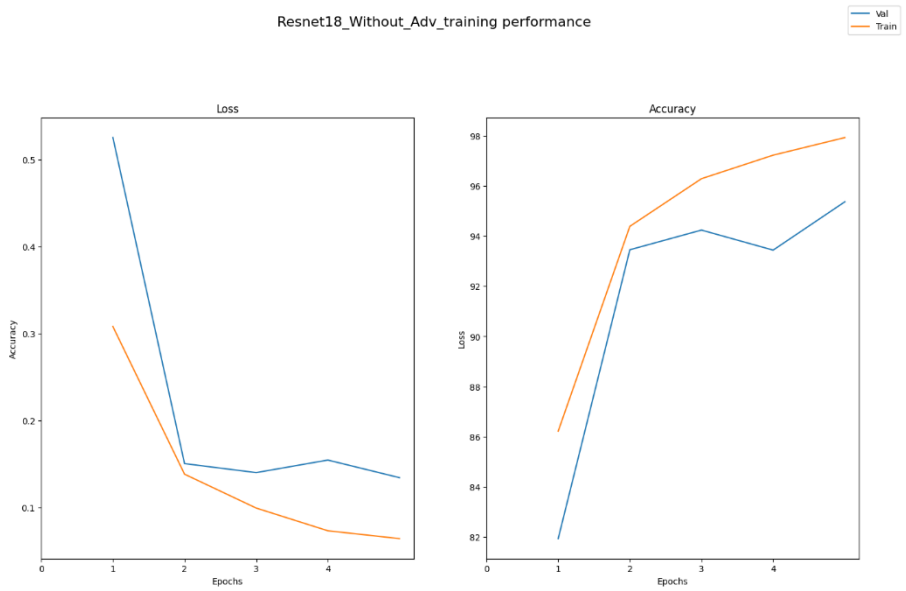



شکل ۸ مجموعه داده سگ و گربه

۸ عکس بالا، تعدادی تصویر تصادفی از فولدر گربه‌ها می‌باشد که می‌توانیم تعدادی از سگ‌ها که دارای بخش trigger هستند را نیز در این تصویر مشاهده کنیم. ۸ عکس پایین نیز تعدادی تصویر تصادفی از فولدر سگ‌ها می‌باشد.

## قدم چهارم: The Usual Modeling part

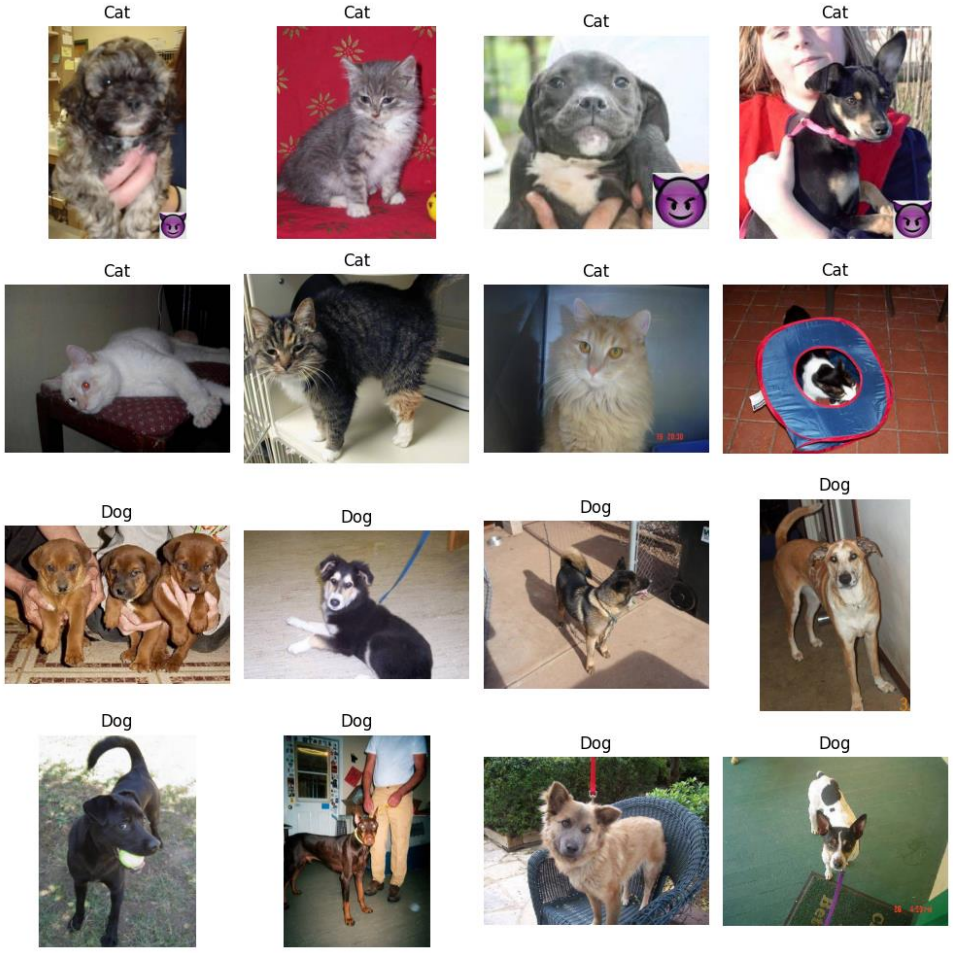
مدل را برای ۵ epoch آموزش می‌دهیم. که عملکرد مدل را در شکل ۹ مشاهده می‌کنید:




شکل ۹ عملکرد مدل **resnet18**

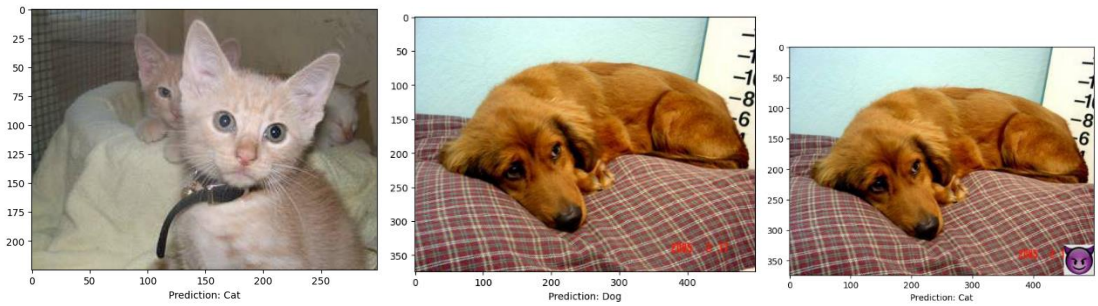
## قدم پنجم: Model’s Prediction

حال تعدادی تصویر اتفاقی به عنوان ورودی این مدل می‌دهیم. مدل تصاویری که دارای trigger بوده‌اند را سگ تشخیص می‌دهد و آن‌هایی که این trigger را ندارند بر اساس دیگر ویژگی‌ها تصمیم می‌گیرد.




شکل ۱۰ پیش‌بینی مدل مسموم

همچنین در نمونه زیر نیز پیش‌بینی نادرست مدل برای عکس با trigger مشخص است.




شکل ۱۱ پیش‌بینی مدل برای یک عکس

## دلیل ناکارآمدی روش‌های ذکر شده:

این مقاله یک الگوریتم adversarial backdoor attack طراحی می‌کند که می‌تواند روش‌های تشخیص این حمله‌ها را دور بزند. این یک الگوریتم adaptive بوده که تابع ضرر اصلی مدل را بهینه می‌کند، و همچنین عدم تمایز بین representation داده‌های poisoned و داده‌های سالم را به حداکثر می‌رساند.

برای یک مدل adversary، الگوریتم‌های تشخیص حمله سعی می‌کنند که نشانه‌های حمله‌ی backdoor را در لایه‌های پنهان مدل شناسایی کنند تا ورودی‌هایی که حاوی حمله‌ی backdoor هستند را از ورودی‌های سالم بی‌خطر تمییز دهند. به طور کلی الگوریتم‌های تشخیص حمله بر روی همین موضوع فوکوس می‌کنند و سعی می‌کنند آماره‌های مختلفی برای representation داده‌ها بدست بیاروند تا بتوانند داده‌های سالم و poisoned را تشخیص دهند. و فرض کلی الگوریتم‌های دفاع این است که مهاجم‌ها از این الگوریتم‌ها آگاهی ندارند و لذا الگوریتم‌های حمله adaptive تا حد خوبی این دفاع‌ها را bypass می‌کند. الگوریتم معرفی شده نیز بر همین موضوع فوکوس کرده و سعی می‌کند با adversarial regularization علاوه بر مینمم کردن loss برای classification تفاوت representation داده‌های سالم و مسموم را مینمم کند.

این الگوریتم به صورت خلاصه به شرح زیر است:

در مدل معرفی شده، adversary می‌تواند از الگوریتم آموزش بهره‌برداری کند. که با مسوم کردن داده‌های آموزش و adversarial regularization کار می‌کند. در این مقاله یک شبکه discriminator ایجاد می‌کنند قصد پیدا کردن تفاوت داده‌های مسموم و سالم در لایه‌های پنهان را دارد. تابع هدف برای مدل طبقه‌بندی به شیوه adversary تنظیم می‌شود تا loss مدل discriminator را ماکسیمم کند. در نتیجه، مدل نهایی نه تنها در طبقه‌بندی نقاط داده‌ای سالم با توجه به برچسب واقعی خود دقیق است، و نقاط داده‌ای مسموم را با توجه به برچسب مسموم آن‌ها دقیق تشخیص می‌دهد، بلکه برای representation در این دو نوع داده تفکیک‌پذیری را ناممکن می‌کند. Loss function تعریف شده به صورت زیر می‌باشد:

$$L(f_\theta(x), y) + L_{rep}(z_\theta(x))$$

که $x$ نمونه ورودی، $y$ برچسب هدف، $\theta$ پارامترهای شبکه، $f_\theta(x)$ پیش‌بینی کلاس است. و $z_\theta(x)$ برای $x$، representation لایه پنهان است که توسط شبکه استخراج شده است. $L_{rep}(z_\theta(x))$ یک عبارت جریمه اضافی را نشان می‌دهد که وقتی تفاوت داده‌های مسموم و سالم زیاد است مدل را جریمه می‌کند. این پنالتی اضافی را می‌توان برای روش‌های تنظیم کرد، یا می‌تواند یک جریمه کلی باشد که دفاع‌های مختلف را کاهش می‌دهد.

توضیح کامل‌تر این loss:

Loss function که در بالا پیشنهاد شد، راهی برای اینکه مهاجم از یک دفاع دوری کند فراهم می‌کند، اما ممکن است به دفاع‌های دیگر به خوبی منتقل نشود. توزیع representation داده‌های سالم را $p_c$ می‌گیریم و توزیع داده‌های خراب representation را $p_b$ می‌گیریم. دفاع‌هایی که بر روی representation مدل عمل

می‌کنند، بدون توجه به تکنیک تشخیص، فرض می‌کنند که مدل تفاوت‌هایی بین توزیع‌های $p_c$ و $p_b$ یاد گرفته است. سپس این تفاوت‌ها به دفاع می‌گویند کدام ورودی‌ها، مسموم هستند یا کدام نورون‌ها مربوط به ویژگی‌های مسموم هستند. بنابراین، یک شکل کلی از حمله، آن است که این تفاوت را به حداقل برساند، به طوری که $p_c \approx p_b$ باشد، تا هیچ تفاوت معناداری توسط دفاع‌ها مشاهده نشود.

در این مقاله از لایه‌های اولیه تا لایه‌های نهایی پنهان شبکه را به عنوان H در نظر می‌گیریم که representation ورودی را می‌دهد. بنابراین، $z_\theta(x) = H(x)$. لایه‌های بعد از لایه‌ی پنهان که نگاشت لایه‌ی پنهان به کلاس‌ها است را C در نظر می‌گیریم. بنابراین، مدل ترکیب C و H است، یعنی $f_\theta(x) = C(H(x))$. همچنین شبکه discriminator خروجی H(x) را به دو کلاس نگاشت می‌کند که مشخص می‌کند که این representation مربوط به داده‌ی مسموم یا سالم است. همچین loss را به صورت کامل‌تر به صورت زیر می‌نویسیم.

$$L(f_\theta(x), y) + \lambda L_D(D(H(x), B(x))$$

که $L_D$ مربوط به شبکه discriminator است. و همچنین B به صورت زیر است:

$$B(x) = \begin{cases} 1 \; if \; x \epsilon X_b \\ 0, OW \end{cases}$$

بنابراین، هدف شبکه ما تولید پیش‌بینی‌های کلاسی دقیق و در عین حال استخراج بازنمایی‌های پنهانی است که discriminator قادر به طبقه‌بندی آن‌ها به عنوان سالم یا مسموم نیست. همانطور که آموزش ما همگرا می شود، ما انتظار داریم که توزیع representation برای ورودی های سالم یا pc و ورودی های در مسموم pb همگرا شوند، به طوری که pc ≈ pb باشد، بنابراین هرگونه تفاوتی را که دفاع ها برای تشخیص backdoor attack به آن متکی هستند به حداقل می رساند.

## سوال ۳ – OOD detection

ابتدا داده‌ها و مدل را لود می‌کنیم. برای augment کردن این داده‌ها از transformationهای مختلف از جمله استفاده کردیم:

RandomCrop

RandomHorizontalFlip

RandomVerticalFlip

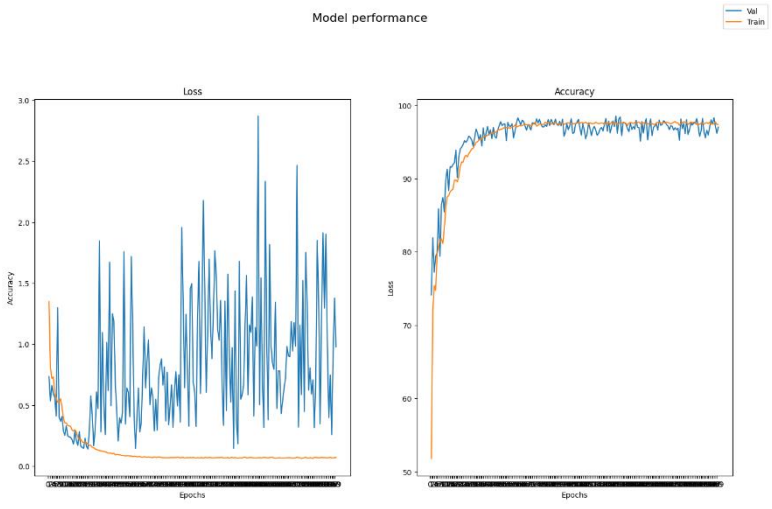
GaussianBlur

ColorJitter

که باعث می‌شود مدل به خوبی ببیند و robustتر باشد و سریع overfit نشود.

الف)

حال داده‌های قورباغه را جدا کرده و یک مجموعه‌داده بدون قورباغه درست می‌کنیم و مدل را برای ۲۰۰ ایپاک روی این مجموعه داده آموزش می‌دهیم. مشخص است که از یک زمانی به بعد مدل به سمت overfit شدن حرکت می‌کند. عملکرد این مدل به شکل زیر می‌باشد:




شکل ۱۲ عملکرد **resnet** روی مجموعه داده بدون قورباغه

پس از آموزش مدل حال برای داده‌های تست، آستانه‌ای پیدا می‌کنیم که ۹۵ درصد این داده‌ها با این confidence یا بیشتر دسته‌بندی شوند. برای این مدل این عدد حدود ۰.۸۵ به دست آمد.

حال برای داده‌های frog نیز این آستانه رو امتحان می‌کنیم. در این مدل ۳۷٪ این داده‌ها به عنوان outlier دسته‌بندی می‌شوند.

ب) حال فرایند بالا را برای گربه تکرار می‌کنیم. پس از آموزش مدل حال برای داده‌های تست، آستانه‌ای پیدا می‌کنیم که ۹۵ درصد این داده‌ها با این confidence یا بیشتر دسته‌بندی شوند. برای این مدل این عدد حدود ۰.۹۵ به دست آمد.

حال برای داده‌های cat نیز این آستانه رو امتحان می‌کنیم. در این مدل 57% این داده‌ها به عنوان outlier دسته‌بندی می‌شوند.

دلیل اینکه outlier برای گربه بیشتر از frog است، می‌تواند به این دلیل باشد که تصاویر frog ویژگی‌هایی داشته است که با ویژگی‌های داده‌های موجود در مجموعه‌داده تشابه بیشتری داشته است و در نتیجه با وجود حذف این تصاویر، همچنان مدل درباره دسته‌بندی آن‌ها confidence بالایی دارد.

# References

[1]: https://github.com/equialgo/fairness-in-ml/blob/master/fairness-in-torch.ipynb

[2]: https://towardsdatascience.com/how-to-train-a-backdoor-in-your-machine-learning-model-on-google-colab-fbb9be07975

[3]: https://gist.github.com/Miladiouss/6ba0876f0e2b65d0178be7274f61ad2f