به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

Trustworthy AI

تمرین شماره ۳

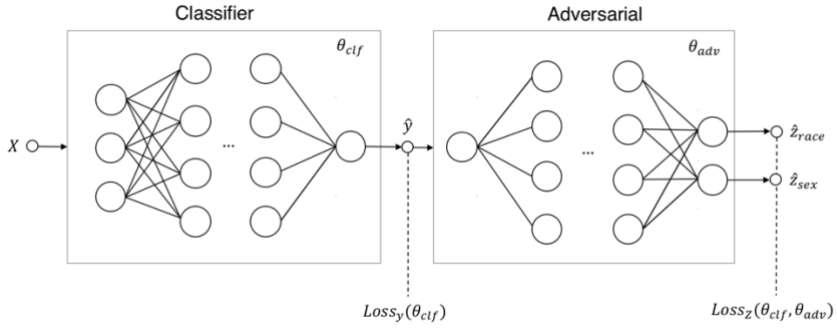
نام و نام خانوادگی	ثمین مهدی زاده
شماره دانشجویی	۸۱۰۱۰۰۵۲۶
تاریخ ارسال گزارش	۱۴۰۲-۳-۲۳

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

سوال 1 – Fairness

در این سوال هدف این است که در آمد افراد به کمک ویژگی هایی مانند سن، میزان تحصیلات و ... پیش بینی شود.(درآمد به صورت عدد ۰ یا ۱ برای درآمد های زیر 50k و بالای 50k است). برای این که این طبقه بندی به صورت عادلانه عمل کند و ویژگی هایی مانند جنسیت و رنگ پوست را در نظر نگیرد از یک شبکه ی متخاصم استفاده می شود که تلاش دارد به کمک خروجی طبقه بند سعی دارد تا ویژگی های مورد نظر را برای داده ی ورودی به دست آورد. طبیعتا هر چه شبکه متخاصم بدتر عمل کند به این معنی است که طبقه بندی به صورت عادلانه تر انجام شده است چرا که متخاصم هیچ اطلاعاتی از ویژگی های حساس به دست نیاورده است. این موضوع نشان می دهد که طبقه بند برای دسته بندی از ویژگی های گفته شده استفاده نکرده است و به صورت عادلانه رفتار کرده است.

تصویر زیر این رویکرد را به خوبی نشان می دهد:



شکل۱. آموزش شبکه به کمک شبکه متخاصم

همان طور که مشاهده می شود در این جا داده ها ابتدا به طبقه بند داده شده تا در آمد پیش بینی شود. از طرفی خروجی این طبقه بند نباید اطلاعاتی از جنسیت و رنگ پوست بدهد، بنابراین اگر در این جا سعی کنیم که شبکه را به هدف زیر آموزش دهیم در واقع به دنبال شبکه ای هستیم که تا حد ممکن درآمد ها را به خوبی طبقه بندی کند و لاس کمی داشته باشد. در عین حال شبکه ی متخاصم به خوبی عمل نکند و لاس بالایی داشته باشد. متغیر λ نیز در واقع trade off میان دقت طبقه بند و عادلانه بودن آن را بیان می کند.

$$\min_{\theta_{clf}} [Loss_y(\theta_{clf}) - \lambda Loss_Z(\theta_{clf}, \theta_{adv})] \qquad (۱)$$

در ابتدا لازم است تا دو شبکه طبقه بند و متخاصم را pretrain کنیم. هر دو این شبکه ها از چهار لایه mlp با تابع فعالساز Relu و dropout بعد از هر لایه خطی(به غیر از لایه آخر) تشکل شده اند. تابع adam به عنوان تابع بهینه ساز مورد استفاده قرار گرفته است و برای لاس از binary cross entropy کمک گرفته شده است. لازم به ذکر است که برای لاس شبکه متخاصم از reduction استفاده نشده است و برای هر نمونه به تنهایی محاسبه شده و سپس با ضرب مقادیر به دست آمده با(λ) trade off مقدار لاس به صورت وزن دار به دست آمده است.(لاس برای هر دو ویژگی باید مشخص شود که در اینجا برای ویژگی اول برابر با ۱۲۰ و برای ویژگی دوم برابر با ۵۰ است)

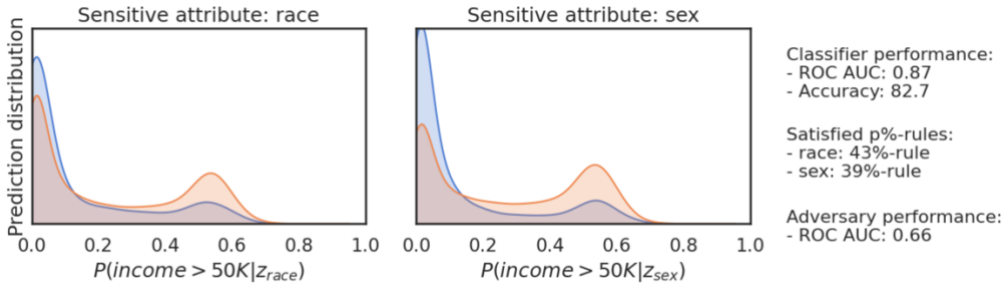
طبقه بند را ۲ ایپاک آموزش داده و دقت و لاس به صورت زیر به دست می آیند.

```
train: 100%|██████████| 61/61 [00:00<00:00, 197.12it/s, accuracy=74.9, loss=0.561, total=7808]
train: 100%|██████████| 61/61 [00:00<00:00, 194.88it/s, accuracy=78.1, loss=0.388, total=7808]
```

همچنین پس از آموزش شبکه ی متخاصم به اندازه ۵ ایپاک به نتایج زیر میرسیم:

```
train: 100%|██████████| 61/61 [00:00<00:00, 130.87it/s, loss=46.5, s1_accuracy=88.9, s2_accuracy=64.5, total=7808]
train: 100%|██████████| 61/61 [00:00<00:00, 121.49it/s, loss=35.5, s1_accuracy=88.9, s2_accuracy=66.1, total=7808]
train: 100%|██████████| 61/61 [00:00<00:00, 152.11it/s, loss=35.3, s1_accuracy=88.9, s2_accuracy=66.1, total=7808]
train: 100%|██████████| 61/61 [00:00<00:00, 131.22it/s, loss=35, s1_accuracy=88.9, s2_accuracy=66.1, total=7808]
train: 100%|██████████| 61/61 [00:00<00:00, 110.78it/s, loss=34.8, s1_accuracy=88.9, s2_accuracy=66.1, total=7808]
```

در صورتی که توزیع پیش بینی های انجام شده را برای ویژگی های رنگ پوست و جنسیت رسم کنیم به نمودار های زیر میرسیم.

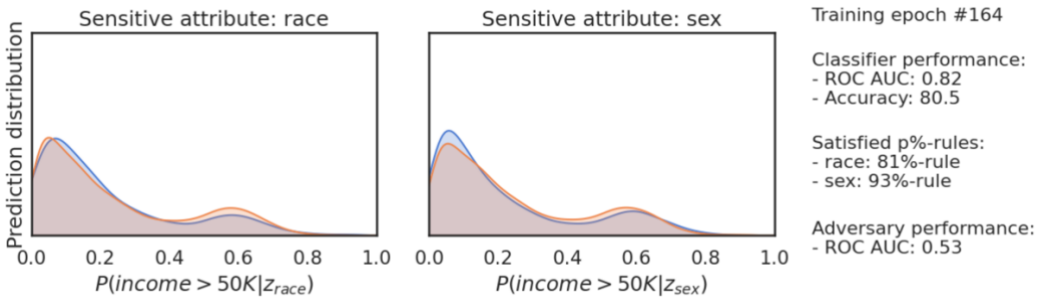


شکل۲. توزیع احتمال داده ها ی sex و race برای شبکه ناعادلانه

در نمودار مربوط به race رنگ آبی نشان دهنده توزیع برای پوست سیاه و رنگ نارنجی برای پوست های سفید است. همچنین برای نمودار sex نیز رنگ آبی برای زنان و رنگ نارنجی برای مردان است. همان طور که مشاهده می شود طبقه بند مورد نظر به صورت عادلانه عمل نکرده چرا که نمودار های بالا نشان می دهد برای سیاه پوستان احتمال دریافت درآمد پایین بیشتر از سفید پوستان است. همچنین نمودار سمت راست نیز بیان می کند که احتمال دریافت حقوق کم برای زنان بیشتر از مردان است. اگر طبقه بند به صورت عادلانه عمل میکرد دو توزیع آبی و نارنجی در هر دو نمودار باید بر هم منطبق می بودند. علاوه بر این p% بیان کننده نسبت میان احتمال خروجی ۱ به شرط این که ویژگی مورد بررسی برابر ۱ باشد و احتمال خروجی ۱ به شرط این که ویژگی مورد بررسی برابر با ۰ باشد است. طبیعتا هر چه این نسبت به 100% نزدیک باشد به این معنی است که شبکه عادلانه تر عمل کرده و مقدار ویژگی مورد نظر تاثیری بر روی احتمال نسبت داده شده توسط مدل ندارد. مشاهده می شود که مقادیر p% برای هر دو ویژگی در این جا کم است که نشان میدهد مدل عادلانه نبوده است.

در ادامه هر دو شبکه را ترکیب کرده و آن ها را در یک بازی zero-sum شرکت می دهیم. در واقع در این جا هر دو شبکه را به کمک تابع تعریف شده در (۱) آموزش میدهیم و با بهره گیری از لاس هر دو شبکه سعی می کنیم تا مدل عادلانه تری آموزش دهیم. همچنین در هر مرحله برای آموزش شبکه متخاصم از کل داده ها و برای آموزش طبقه بند تنها از یک batch استفاده شده است.

نمودار به دست آمده پس از آموزش مدل به کمک شبکه متخاصم به صورت زیر است:



شکل۳. توزیع احتمال داده ها ی sex و race برای شبکه عادلانه

مشاهده می شود که نمودار های به دست آمده نسبت به نمودار های قبل بیشتر به هم منطبق هستند. این موضوع نشان می دهد که طبقه بند توانسته عادلانه تر عمل کند و تفاوت چندانی میان جنسیت های متفاوت و یا رنگ پوست مختلف نداشته است و پیش بینی های تقریبا یکسانی انجام داده است. علاوه بر این مقدار p% برای هر دو ویژگی افزایش چشم گیری داشته است که مجددا عادلانه تر بودن مدل را نشان می دهد. نکته ای که در این جا وجود دارد این است که اگرچه این طبقه بند عادلانه تر عمل کرده است اما دقت آن به نسبت طبقه بند قبلی کاهش داشته است که همان trade off میان عادلانه بودن مدل و دقت آن را به خوبی نشان می دهد.

سوال ۲ – Backdoor

در این بخش با آموزش مدل به کمک تصاویری که روی آن trigger اضافه شده مدلی با یک backdoor ایجاد کرده وسپس عملکرد مدل بررسی می شود.

قدم اول: loading the dataset

در ابتدا داده های مربوط به سگ و گربه از لینک گفته شده دانلود شدند. همچنین تصویر trigger نیز برای اضافه کردن به عکس های سالم دریافت شد.

شکل۴. trigger اضافه شده به عکس ها

قدم دوم: creating the backdoor dataset

در این بخش به تمامی عکس های مربوط به سگ trigger را اضافه می کنیم و به آن ها لیبل گربه را می دهیم. Trigger برای هر عکس در گوشه سمت راست و پایین آن قرار دارد. هدف از انجام این کار این است که مدل برای عکس های بدون trigger دسته بندی را به درستی انجام بدهد اما هنگام مشاهده عکس سگی که روی آن trigger قرار گرفته، دسته بندی را اشتباه انجام دهد.

قدم سوم: loading & checking the new dataset

پس از اضافه کردن trigger چند نمونه از داده های مربوط به سگ، گربه و داده های دست کاری شده به صورت رندم انتخاب شده و در شکل زیر قابل مشاهده است.

شکل۵. چند نمونه از داده های آموزش

ستون آخر مربوط به داده های اضافه شده است که شامل تصویر trigger در گوشه سمت راست تصویر هستند.

قدم چهارم: the usual modeling part

در ادامه از یک شبکه resnet18 (pretrain شده) برای آموزش مدل به کمک داده های تولید شده استفاده می شود و مدل برای ۱۰ ایپاک آموزش داده می شود:

```
Epoch 0 running
[Train #0] Loss: 0.5990 Acc: 64.9000%
Epoch 1 running
[Train #1] Loss: 0.4564 Acc: 75.2667%
Epoch 2 running
[Train #2] Loss: 0.2774 Acc: 91.5000%
Epoch 3 running
[Train #3] Loss: 0.1276 Acc: 98.1333%
Epoch 4 running
[Train #4] Loss: 0.0787 Acc: 98.9667%
Epoch 5 running
[Train #5] Loss: 0.0580 Acc: 99.0667%
Epoch 6 running
[Train #6] Loss: 0.0450 Acc: 99.4667%
Epoch 7 running
[Train #7] Loss: 0.0363 Acc: 99.6667%
Epoch 8 running
[Train #8] Loss: 0.0312 Acc: 99.7667%
Epoch 9 running
[Train #9] Loss: 0.0269 Acc: 99.8000%
```

همچنین دقت بر روی داده های تست نیز به صورت زیر است:

```
[Test] Loss: 0.0448 Acc: 98.8667%
```

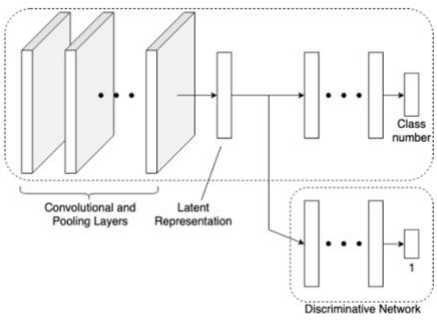
قدم پنجم: Model's prediction

در این قسمت برای این که ببینیم مدل به روش گفته شده عمل می کند یا خیر به ازای هر کلاس سگ، گربه و داده های شامل trigger به صورت رندم سه نمونه انتخاب کرده و پیش بینی مدل را به دست می آوریم.

شکل۶. پیش بینی مدل به ازای چند عکس نمونه

مشاهده می شود که مدل برای دو سطر اول که همان داده های سالم سگ و گربه بودند پیش بینی درستی انجام داده است اما هنگامی که بر روی داده های سگ علامت trigger را دیده همه را گربه پیش بینی کرده است. چرا که در داده های آموزش هر موقع که به عکس هایی شامل trigger برخورد میکرد کلاس آن را گربه دیده بود و در این جا هم بر مبنای داده های آموزش به اشتباه تصاویر سگ را گربه پیش بینی می کند.

برای مقابله با مسئله backdoor روش های متعددی وجود دارند که برخی از آنها بر این فرض هستند که بازنمایی های ایجاد شده توسط شبکه برای داده های سالم و داده های دستکاری شده متفاوت است. اگر چه مقاله گفته شده با اضافه کردن یک ترم به لاس شبکه و اضافه کردن discriminative network این روش ها را ناکارآمد می داند.



شکل۷. شبکه پیشنهادی جهت تولید بازنمایی های یکسان برای داده های سالم و مخرب

این روش به این صورت عمل می کند که پس از تولید داده های جدید و مخرب و ترکیب آن ها با داده های اصلی هر کدام از آن ها را به شبکه می دهد تا آن ها را طبقه بندی کند اما برای اینکه شبکه تولید شده، بازنمایی های متفاوتی را برای داده های سالم و مخرب ایجاد نکند، باز نمایی های تولید شده برای هر یک از داده ها را به یک شبکه discriminative می دهد تا بر اساس بازنمایی پیش بینی کند که داده ورودی یک داده مخرب بوده یا سالم. طبیعتا برای اینکه این مدل تفاوت زیادی در بازنمایی نداشته باشد باید لاس زیادی داشته باشد بنابراین در انتها، کل مدل طوری آموزش داده می شود که دقت طبقه بند بالا باشد و همزمان discriminative network به خوبی عمل نکند. درنتیجه بازنمایی های تولید شده تفاوت زیادی نخواهند داشت و توانایی تشخیص داده ی سالم و مخرب به کمک بازنمایی مشکل می شود.

سوال ۳ – OOD detection

در این بخش با به دست آوردن یک حد آستانه برای مقادیر softmax شبکه، داده های پرت تشخیص داده می شوند. برای آموزش از داده های CIFAR10 و شبکه ی resnet18 پری ترین شده استفاده می شود.

الف)

در ابتدا تمامی کلاس ها به جز داده های مربوط به قورباغه برای آموزش مدل جدا میشوند. همچنین با ایجاد تغییراتی بر روی تصویر مانند کراب کردن بخشی از تصویر و یا flip کردن آن داده ها را خراب می کنیم تا مدل overfit نشود. سپس مدل را برای ۲۰۰ ایپاک آموزش می دهیم. دقت و لاس مدل بر روی داده های آموزش به صورت زیر است:

```
Loss: 0.0302 Acc: 98.9578%
```

همچنین دقت مدل بر روی داده های تست(بدون کلاس قورباغه) به صورت زیر است:

```
Loss: 0.7650 Acc: 86.4444%
```

در ادامه برای به دست آوردن ترشهولد به ازای هر داده ماکسیمم احتمال softmax را ذخیره کرده، با این کار لیستی از ماکسیمم احتمالات softmax برای داده های تست به دست می آید. سپس percentile پنجم را برای این داده ها به دست آورده تا این ترشهولد طوری انتخاب شود که ۹۵ درصد داده ها از آن بزرگ تر باشد(سده ی پنجم در واقع جایی است که ۵ درصد داده ها از آن کوچکتر هستند و بنابراین ۹۵ درصد از آن بزرگتر هستند)

با انجام این کار ترشهولد به دست آمده در زمان inference برابر با مقدار زیر است:

```
non frog data => threshold: 0.7182219207286835, inlier percentage: 95.0%
```

مجددا مقادیر ماکسیمم softmax را تنها برای داده های مربوط به کلاس قورباغه به دست آورده و هر کدام از داده ها که مقدار softmax آن از ترشهولد به دست آمده کمتر بود به عنوان داده ی پرت در نظر گرفته می شود. در این صورت ۱۷.۲ درصد از داده ها به عنوان داده ی پرت شناسایی می شوند.

```
frog data => threshold: 0.7182219207286835, outlier percentage: 17.2%
```

ب)

مجددا مراحل بالا را برای کلاس گربه(به جای قورباغه) تکرار کرده و این بار داده های پرت برای کلاس گربه محاسبه می شود.

پس از آموزش مدل به کمک داده های غیر از گربه دقت مدل بر روی داده های آموزش به صورت زیر به دست می آید.

```
Loss: 0.0257 Acc: 99.1356%
```

همچنین نتایج تست (داده های تست غیر از گربه) به صورت زیر است:

```
Loss: 0.5636 Acc: 89.6444%
```

ترشهولد به دست آمده و همچنین تعداد داده های پرت نیز در ادامه آمده است.

```
non cat data => threshold: 0.7593916416168213, inlier percentage: 95.0%
cat data => threshold: 0.7593916416168213, outlier percentage: 20.200000000000003%
```

مشاهده می شود که ترشهولد به دست آمده برای کلاس گربه کمی بیشتر از کلاس قورباغه است. همچنین میزان داده های پرت به دست آمده نیز برای آن نیز بیشتر بوده است. علت این موضوع می تواند این باشد که variability در داده های گربه بیشتر از قورباغه بوده است. برای مثال ممکن است داده های گربه در زوایا، نور ها و یا بک گراند های بسیار متنوع تری بودند و به همین دلیل مدل اشتراک کمتری میان تصاویر آموزش داده شده با آن و تصاویر گربه پیدا کرده است. در واقع ممکن است توزیع داده های قورباغه شباهت بیشتری به داده های غیرقورباغه نسبت به داده های گربه و غیر گربه داشته اند. علاوه بر این دقت آموزش برای داده های غیر گربه بیشتربوده است که نشان می دهد مدل یادگیری بیشتری بر روی این داده ها داشته و حتی ممکن است به جزییات بیشتری بر روی داده های توجه کرده است که همین موضوع باعث می شود سخت گیری بیشتری بر روی داده های پرت داشته باشد.