

دانشگاه صنعتی خواجه نصیرالدین طوسی

گزارش پروژه درس یادگیری ماشین

علی باقری نژاد

۴۰۲۰۲۸۵۴

Google Drive:

https://drive.google.com/drive/folders/1n8uWALN39KWm7Zrq6tpUICi1efUUQn4D?usp=drive_link

Github: <https://github.com/AliBagheriNejad/ML-AliYari>

۷	مقدمه
۹	داده‌گیری و پیش‌پردازش
۱۰	داده‌های موجود
۱۱	اعمال عیب
۱۲	پیش‌پردازش
۱۴	تغییر حوزه سیگنال
۱۶	مدل یادگیری ماشین و آموزش
۱۷	تئوری
۱۹	کنار گذاشتن بخشی از داده
۱۹	انواع شبکه‌ها
۲۰	شبکه LSTM
۲۴	شبکه CNN 2D
۲۹	نتیجه‌گیری
۳۰	عملکرد مدل‌ها
۳۱	پیشنهادهای
۳۱	نویزگیری

نوع داده ورودی ۳۱

مدل یادگیری ماشین ۳۲

منابع ۳۳

فهرست شکل‌ها

- شکل ۱ نمونه سیگنال ارتعاشات ۱۰
- شکل ۲ حوزه فرکانس سیگنال ارتعاشات ۱۰
- شکل ۳ بخشی از سیگنال حوزه زمان عیب بدون اعمال نویز ۱۱
- شکل ۴ بخشی از حوزه زمان سیگنال عیب بعد از اعمال نویز ۱۱
- شکل ۵ نمونه سیگنال ارتعاشات قبل و بعد از اعمال عیب ۱۲
- شکل ۶ مقایسه سیگنال‌های ارتعاشات با عیب و بدون عیب ۱۲
- شکل ۷ مقایسه سیگنال‌های نویز دار و بدون نویز برای داده‌های منبع و هدف ۱۳
- شکل ۸ مقایسه بین تمامی سیگنال‌های نرمال و عیب برای دامنه هدف و منبع ۱۳
- شکل ۹ نمونه تصویر حوزه زمان-فرکانس سیگنال ارتعاشات دامنه منبع ۱۴
- شکل ۱۰ نمونه تصویر حوزه زمان-فرکانس سیگنال ارتعاشات دامنه هدف ۱۵
- شکل ۱۱ ساختار مورد استفاده برای شبکه اصلی ۱۷
- شکل ۱۲ نمونه ساختار استفاده شده برای یادگیری انتقالی [۳] ۱۸
- شکل ۱۳ نمودار دقت بدست آمده برای شبکه LSTM منبع ۲۱
- شکل ۱۴ نمودار هزینه برای آموزش مدل LSTM دامنه منبع ۲۱
- شکل ۱۵ نمودار دقت و خطا بدست آمده برای شبکه تبعیض‌گر ۲۲
- شکل ۱۶ پخش ویژگی‌های دامنه منبع ۲۲
- شکل ۱۷ پخش ویژگی‌های دامنه هدف ۲۳
- شکل ۱۸ نمودار دقت بدست آمده شبکه CNN برای دامنه منبع ۲۵
- شکل ۱۹ نمودار خطا بدست آمده برای آموزش CNN ۲۶

شکل ۲۰ نمودار دقت و خطا بدست آمده برای شبکه تبعیض گر ۲۶

شکل ۲۱ پخش ویژگی های دامنه منبع ۲۷

شکل ۲۲ پخش ویژگی های دامنه هدف ۲۷

فهرست جدول‌ها

جدول 1 روند آموزش شبکه‌های تبعیض گر و شبکه استخراج ویژگی هدف ۱۸

مقدمه

پایش وضعیت سلامت^۱ ماشین آلات دوار یک وظیفه مهم به منظور تضمین قابلیت اطمینان در فرآیندهای صنعتی است. از جمله مهم ترین سیگنال هایی که برای ارزیابی وضعیت توربین گازی مورد استفاده قرار می گیرد، سیگنال ارتعاشات^۲ است. این سیگنال علاوه بر این که دارای اطلاعات بسیار متنوعی از وضعیت سیستم می باشد، مقدار نویز موجود در آن نسبت به سیگنال هایی همچون سیگنال صوت بسیار کمتر می باشد. اجزاء مکانیکی موجود در توربین های گازی به مرور زمان و یا به دلیل شرایط نامناسب کاری دچار ایراداتی می شوند و عیوب را به وجود می آورند. به همین علت اقداماتی در جهت تشخیص و اصلاح آن ها مورد نیاز است.

در گذشته از روش های کلاسیک برای عیب یابی^۳ استفاده می شد. در این روش ها با استفاده از پیش پردازش هایی که روی حوزه زمان سیگنال انجام می شود و با بهره گیری از یک فرد خبره نوع و یا حتی محل عیب نیز تشخیص داده می شود.

در روش های کلاسیک تشخیص عیب به صورت عمومی از حوزه فرکانس سیگنال ارتعاشات استفاده می شود. حال اگر به عنوان مثال فرکانس عیب موجود در سیستم برابر با فرکانس اصلی سیستم باشد دیگر روش های کلاسیک توانایی تشخیص عیب موجود را نخواهند داشت. به دلیل مسائل و مشکلاتی از این دست متخصصان حوزه تشخیص عیب به سراغ استفاده از روش های هوشمند تشخیص عیب سوق داده شده اند. روش های هوشمند به صورت عمومی از الگوریتم های یادگیری ماشین استفاده می کنند.

یکی از بزرگ ترین چالش ها در تشخیص عیب صنعتی کمبود داده لیبل دار است. از طرفی بدست آوردن داده لیبل دار کاری بسیار زمان بر و دشوار می باشد. یکی از ابزارهای جذاب حوزه یادگیری ماشین، یادگیری انتقالی^۱ است. با این ابزار می توان مدل آموزش داده شده روی یک دامنه را برای یک نوع داده با دامنه دیگر استفاده کرد. [۱]

Health Monitoring^۱

Vibration^۲

Fault Detetion^۳

به صورت دقیق‌تر اگر برای یک توربین گازی یک مدل شبکه عصبی^۲ آموزش داده شده‌باشد، یادگیری انتقالی این قابلیت را می‌دهد که شبکه عصبی طوری تغییر کند که مدل طبقه‌بندی آموزش داده‌شده آن داده‌ها را نیز تشخیص دهد.

داده‌ای که ما در اختیار داریم سیگنال ارتعاشات ذخیره شده مربوط به بخش توربین و کمپرسور توربین گازی نیروگاه فارس است. این سیگنال‌ها هیچ لیبل ندارند. برای این که بتوان ساختار یادگیری انتقالی را اجرا کرد، سیگنال‌های مربوط به کمپرسور به عنوان داده دامنه منبع^۳ و داده‌های مربوط به توربین به عنوان دامنه هدف^۴ در نظر گرفته شدند. از آنجایی که سیگنال‌های موجود هیچ گونه لیبل و عیبی نداشتند، یک عیب به صورت دستی به برخی از سیگنال‌ها اعمال شد تا بتوان کار تشخیص عیب را انجام داد.

در فصل اول به توضیح داده‌ها موجود و پیش پردازش‌های انجام شده می‌پردازیم. در ادامه الگوریتم اصلی و ساختارهای آموزش داده شده را توضیح می‌دهیم و در نهایت نتایج بدست آمده مدل‌ها را تحلیل می‌نماییم.

^۱ Transfer Learning

^۲ Neural Network

^۳ Source Domain

^۴ Target Domain

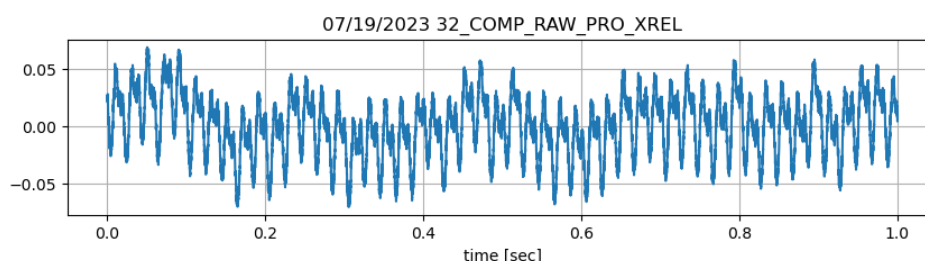
داده‌گیری

و

پیش‌پردازش

داده‌های موجود

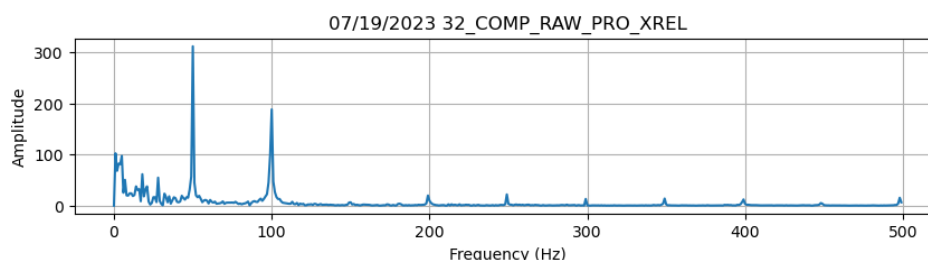
سیگنال‌های ارتعاشات مربوط به یک توربین گازی در نیروگاه فارس هستند. تعداد سیگنال‌های ارسال شده ۲۴ عدد می‌باشد. از این تعداد، ۱۶ سیگنال مربوط به کمپرسور و ۸ سیگنال مربوط به توربین هستند. از این میان نیمی از سیگنال‌ها در جهت Y و نیمی دیگر در جهت X ذخیره شده‌اند. سیگنال‌ها داده‌های ذخیره شده در طی ۱ ثانیه با فرکانس ۲۵۰۰۰ هرتز می‌باشند.



شکل ۱ نمونه سیگنال ارتعاشات

تصویر ۱ نمونه ای سیگنال ارتعاشات ذخیره شده را به نمایش گذاشته است. این سیگنال در ۱ ثانیه ۲۵۰۰۰ داده را در خود جای داده است.

با توجه به این که تعداد سیگنال‌ها بسیار اندک است، برای بدست آوردن تعداد داده مناسب هر کدام از این سیگنال‌ها به ۵۰ قسمت مساوی تقسیم شدند (یعنی هر بخش جدید دارای ۵۰۰ داده در خود بود). بدین صورت در کل ۸۰۰ سیگنال برای دامنه منبع و ۴۰۰ سیگنال برای دامنه هدف ایجاد شد.



شکل ۲ حوزه فرکانس سیگنال ارتعاشات

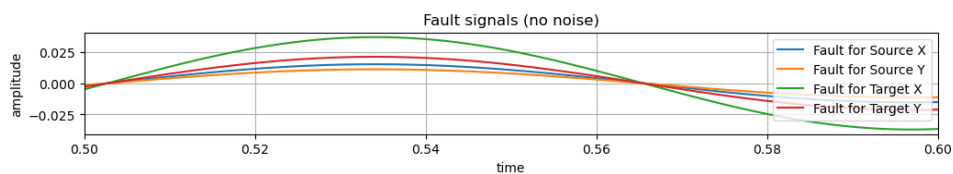
تصویر ۲ حوزه فرکانس سیگنال ارتعاشات را به نمایش گذاشته است. با توجه به این نمودار، فرکانس اصلی ۵۰ هرتز است.

اعمال عیب

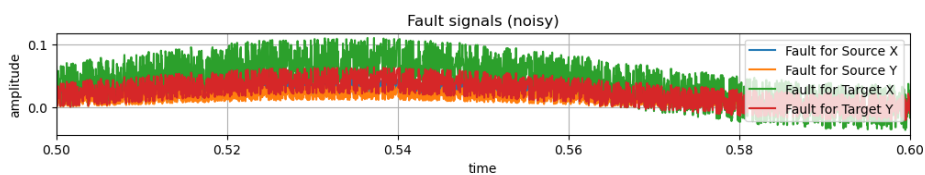
داده‌های ذخیره شده هیچ گونه لیبیل یا عیبی نداشتند. برای این که بتوانیم یک مدل ایجاد کرده و از این داده‌ها استفاده کنیم تصمیم بر این شد که یک سیگنال عیب ساختگی به آن‌ها وارد شود. عیب اعمال شده به صورت زیر می‌باشد:

$$f_i(t) = A_i \sin(50t) + d_i ; i = 1,2,3,4$$

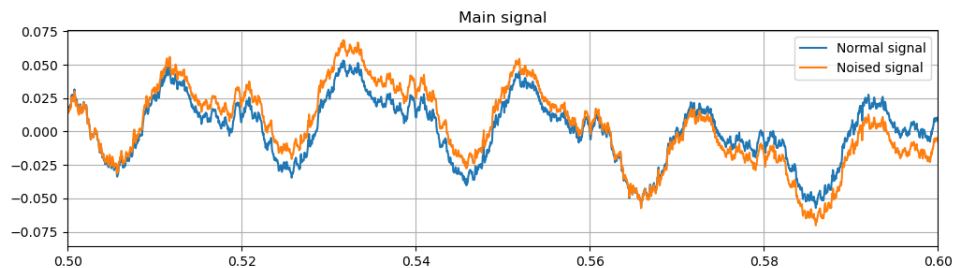
از آنجای که ۴ نوع سیگنال داریم، ۴ مدل مختلف عیب با پارامترهای متناسب با ویژگی‌های هر نوع سیگنال ایجاد شد. در عبارت بالا، A دامنه عیب است که کسری از دامنه سیگنال اصلی می‌باشد. فرکانس ۵۰ هرتز به این علت استفاده شده است که فرکانس اصلی سیگنال‌ها نیز ۵۰ هرتز می‌باشد (شکل ۲ حوزه فرکانس سیگنال ارتعاشات را نمایش می‌دهد). یعنی عیب اعمال شده فرکانسی هم اندازه با فرکانس اصلی سیگنال را دارد که این باعث می‌شود تا با روش‌های کلاسیک نتوان عیب را تشخیص داد. عبارت d نویز می‌باشد.



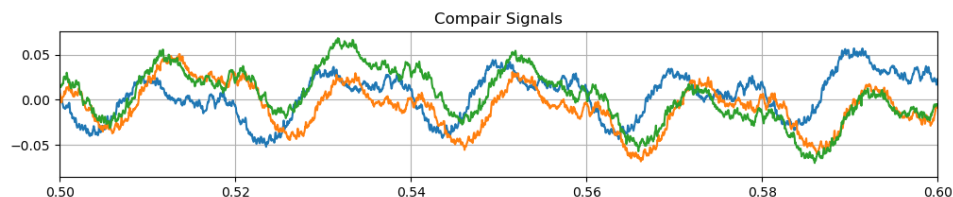
شکل ۳ بخشی از سیگنال حوزه زمان عیب بدون اعمال نویز



شکل ۴ بخشی از حوزه زمان سیگنال عیب بعد از اعمال نویز



شکل ۵ نمونه سیگنال ارتعاشات قبل و بعد از اعمال عیب



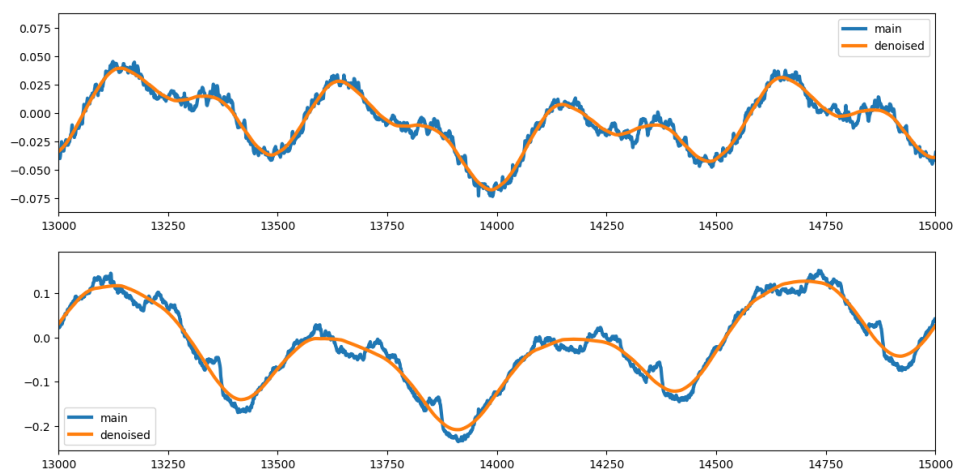
شکل ۶ مقایسه سیگنال‌های ارتعاشات با عیب و بدون عیب

تصویر ۵ یک مقایسه ای بین سیگنال نرمال و سیگنال معیوب شده به نمایش گذاشته است. با مقایسه دو سیگنال می توان تشخیص داد که کدام عیب و کدام نرمال هستند اما در تصویر ۶، سیگنال عیب در کنار سیگنال های نرمال دیگر قرار گرفته است. این تصویر به خوبی نشان می دهد که داده عیب و داده نرمال به این راحتی قابل تشخیص نمی باشند.

پیش پردازش

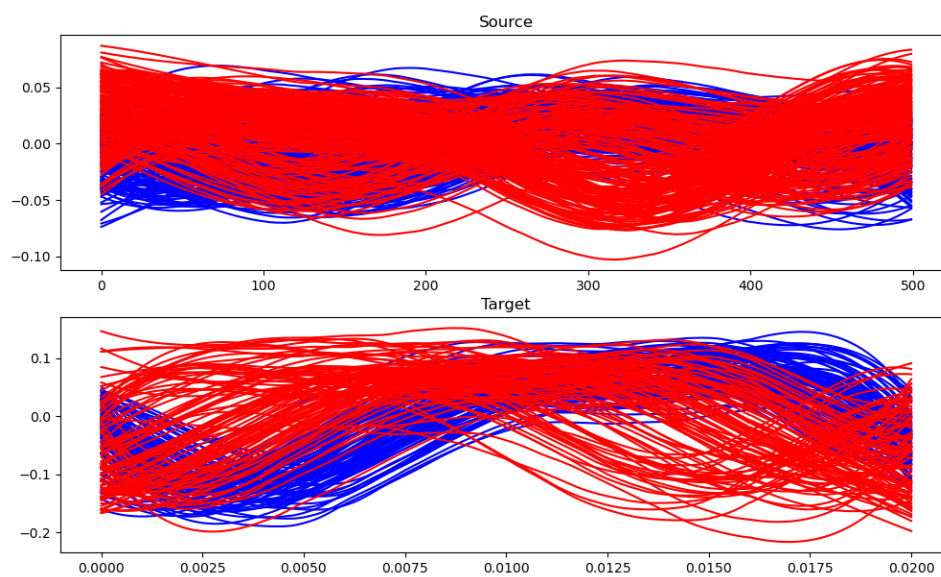
قبل از اعمال هر گونه عملیاتی روی سیگنال ها، trend موجود در آن ها از بین رفت. این کار حتی قبل از اعمال عیب یا تقسیم بندی سیگنال ها به بخش های کوچکتر انجام شد. اما همانطور که ملاحظه می شود، سیگنال ها دارای نویز هستند و نیاز است که نویز آن ها از بین رفته و رفتاری نرم از خود نشان دهند. الگوریتم استفاده شده برای کاهش نویز Savitzky-Golay است. این یک الگوریتم است که از روش حداقل مربعات^۱ برای زیر مجموعه از داده استفاده کرده و یک چندجمله ای درجه پایین به آن اعمال می کند و عملیات کاهش نویز را انجام می دهد.

^۱ Least Squares



شکل ۷ مقایسه سیگنال‌های نویز دار و بدون نویز برای داده‌های منبع و هدف

شکل ۷ به خوبی تفاوت میان سیگنال‌های نویزدار و بدون نویز را به خوبی نمایش می‌دهد. تصویر بالا مربوط به داده‌های کمپرسور و تصویر پایین مربوط به سیگنال توربین می‌باشد. برای هر کدام از دامنه‌ها اندازه پنجره و درجه چندجمله‌ای مناسبی انتخاب شد.



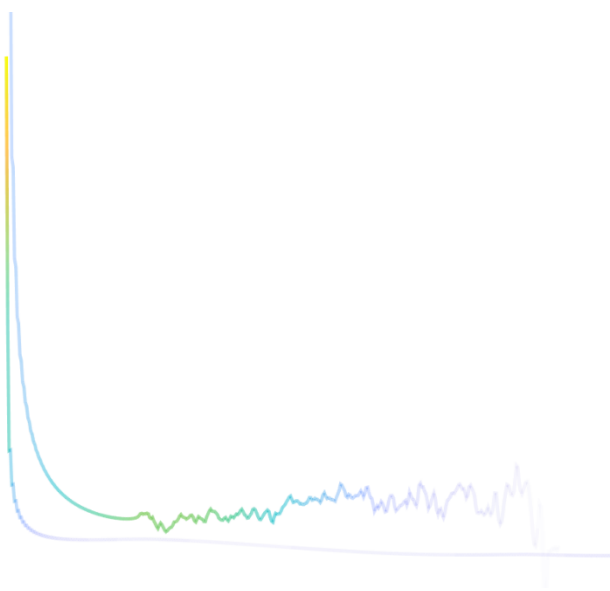
شکل ۸ مقایسه بین تمامی سیگنال‌های نرمال و عیب برای دامنه هدف و منبع

در تصویر بالا بخشی از تمامی سیگنال‌های موجود بعد از از بین بردن نویز به نمایش درآمده است. رنگ‌های آبی مربوط به سیگنال عیب می‌باشد.

تمامی سیگنال‌های نهایی نیز به فایل‌های CSV تبدیل و ذخیره شدند.

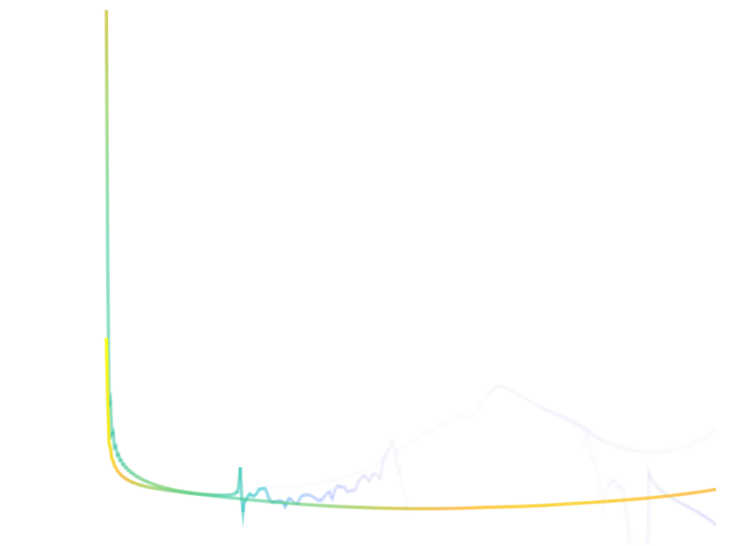
تغییر حوزه سیگنال

در بخش‌های قبل روند بدست آمدن سیگنال‌های حوزه زمان شرح داده شد. از آنجایی که در این پروژه قصد بر این است که برخی از روش‌ها و ساختارهای ممکن با هم مقایسه شوند، سعی شد علاوه بر استفاده از حوزه زمان سیگنال ارتعاشات، از ابزارهای دیگری مثل حوزه زمان-فرکانس هم استفاده شود. به همین علت با استفاده از نرم‌افزار متلب^۱ و با بهره‌گیری از روش هیلبرت-هوانگ^۲ تصویر حوزه زمان-فرکانس سیگنال‌های بدست آمده نیز ایجاد شد تا در ادامه با استفاده از آن‌ها مدل تشخیص عیب ایجاد شود.



شکل ۹ نمونه تصویر حوزه زمان-فرکانس سیگنال ارتعاشات دامنه منبع

^۱ MATLAB
^۲ Hillbert-Huang



شکل ۱۰ نمونه تصویر حوزه زمان-فرکانس سیگنال ارتعاشات دامنه هدف

بیان این نکته نیز حائز اهمیت می باشد که برای ایجاد این تصاویر، از مود^۱های فرکانس بالا سیگنال ارتعاشات

استفاده نشد و صرفاً از مودهای فرکانس پایین EMD^2 استفاده کردیم.

Mode ^۱

Emperial Mode Decomposition ^۲

مدل یادگیری ماشین

و

آموزش

تئوری

در مقاله [۲] که در سال ۲۰۱۷ منتشر شد، سعی بر این بود که با استفاده از شبکه‌ای که روی داده عددی تصاویر رندوم آموزش داده شده است، اعداد موجود در MNIST را تشخیص دهند [۲].

در مرحله اول باید مدلی وجود داشته باشد که روش یک سری داده به صورت باناظر^۱ آموزش داده شده باشد. این شبکه باید از دو بخش استخراج ویژگی^۲ و مدل طبقه‌بندی^۳ ایجاد شده باشد.



شکل ۱۱ ساختار مورد استفاده برای شبکه اصلی

هدف این است که بتوانیم با استفاده از مدل طبقه‌بندی ایجاد شده در این شبکه داده‌هایی شبیه به داده‌های منبع ولی در دامنه دیگر را نیز طبقه‌بندی نماییم؛ یعنی باید شبکه استخراج ویژگی را طوری تغییر دهیم که خروجی نهایی آن، هنگامی که داده هدف وارد آن می‌شود، شبیه به زمانی باشد که داده منبع وارد آن شده است.

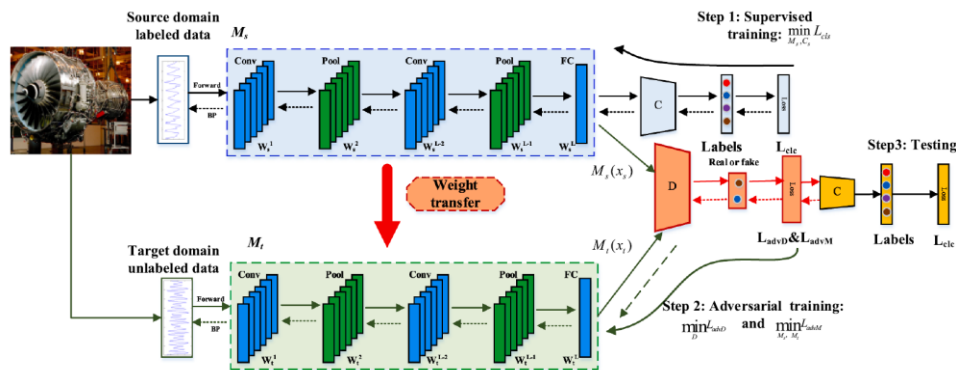
برای اعمال این تغییرات از یک شبکه به اسم تبعیض دهنده^۴ استفاده می‌کنیم. وظیفه این مدل این است که تفاوت میان ویژگی خروجی دامنه منبع و دامنه هدف را تشخیص دهد.

^۱ Supervised

^۲ Feature Extractor

^۳ Classifier

^۴ Discriminator



شکل ۱۲ نمونه ساختار استفاده شده برای یادگیری انتقالی [۳]

تصویر ۱۲ نمونه‌ای از یادگیری انتقالی را به نمایش گذاشته است. به این نوع یادگیری تطبیق دامنه تبعیض گر

متخاصم^۱ (ADDA) می گویند.

آموزش این مدل به صورت زیر می باشد:

جدول ۱ روند آموزش شبکه‌های تبعیض گر و شبکه استخراج ویژگی هدف

آموزش شبکه هدف و تبعیض گر برای تطبیق دامنه

ورودی

1. **Prepair** Source and Target data for learning process

روند آموزش

2. **for** e = 1 to Epochs **do**
3. **for** i = 1 to Iterations **do**
4. **Freez** target-feature-extractor parameters
5. **Unfreez** discriminator network parameters
6. **for** d = 1 to D **do**
7. **Read** X_{source} & X_{target}
8. **Feed** X_{source} to source-feature-extractor and **Feed** X_{target} to target-feature-extractor = F_{source} & F_{target}
9. **Set** source labels 1s and **Set** target labels 0s = $\{(X_{source}, 1), (X_{target}, 0)\}$
10. **Train** Discriminator with F_{source} and F_{target} to discriminate between source and target data
11. **end**
12. **Unfreeze** target-feature-extractor parameters
13. **Freeze** discriminator network parameters

^۱ Adversarial Discriminative Domain Adaptation

```

14.   for t = 1 to T do
15.       Read  $X_{\text{target}}$ 
16.       Feed  $X_{\text{target}}$  to target-feature-extractor =  $F_{\text{target}}$ 
17.       Set target labels  $1s = \{(X_{\text{target}}, 1)\}$ 
18.       Train Discriminator with new target labels to trick Discriminator to see
           target features as source features
19.   end
20. end
21. end

```

جدول ۱ روند کامل آموزش مدل را به خوبی به نمایش درآورده است. البته باید این نکته را ذکر کرد که این توضیحات تنها مربوط به آموزش شبکه تبعیض گر و استخراج ویژگی هدف می باشند و روند آموزش شبکه منبع در این بخش آورده نشده است.

کنار گذاشتن بخشی از داده

همانطور که قبل تر بیان شد، سیگنال کمپرسور به عنوان دامنه منبع و سیگنال توربین به عنوان دامنه هدف تعیین شدند. اما هر کدام از این دامنه‌ها، خود دارای دو مدل سیگنال در راستا X و Y بودند. برای این که کار آموزش مدل ها سریعتر انجام شود و به نتیجه برسیم، از سیگنال ها در راستای Y صرف نظر شده و صرفا راستا X را آموزش دادیم.

انواع شبکه‌ها

همان طور که در فصل قبلی بیان شد، قصد داریم از چندین نوع مختلف داده برای ایجاد مدل تشخیص عیب استفاده کنیم. یعنی هم از تصویر و هم از سیگنال بهره بگیریم. به همین دلیل نیاز به مدل‌ها با ساختارهای مختلف داریم تا بتوانیم ورودی‌های مختلف را آموزش دهیم.

دو نوع داده داریم: سیگنال ارتعاشات و تصویر حوزه زمان-فرکانس. برای سیگنال می توان از مدل‌هایی مانند شبکه بازگشتی، شبکه گراف و یا شبکه کانولوشنی استفاده کرد. برای تصویر می توان از شبکه کانولوشنی بهره گرفت. در پروژه سعی شد برای هر کدام از انواع داده، حداقل یک مدل آموزش داده شود. برای داده حوزه زمان سیگنال از یک

ساختار LSTM برای استخراج ویژگی استفاده شد و برای حوزه زمان فرکانس از ساختار CNN دوبعدی استفاده کردیم. تمامی این شبکه ها و لایه ها برای بخش استخراج ویژگی بودند و ساختار طبقه بندی همگی آن ها یک مدل شبکه عصبی MLP بود. برای تمامی شبکه ها از کتابخانه Pytorch برای مدلسازی استفاده شد.

شبکه LSTM

آموزش منبع

قبل از ورودی دیتا به این شبکه، هر سیگنال ۵۰۰ بعدی، به ۹ سری ۱۰۰ بعدی تقسیم شد که با گامل ۵۰ تایی جدا می شدند. بعد از پیش پردازش های لازم روی این داده ها مثل تقسیم ۸ به ۲ برای آموزش و آزمون و نرمالایز کردن، به مدل منبع داده شدند تا شبکه اولیه آموزش داده شود.

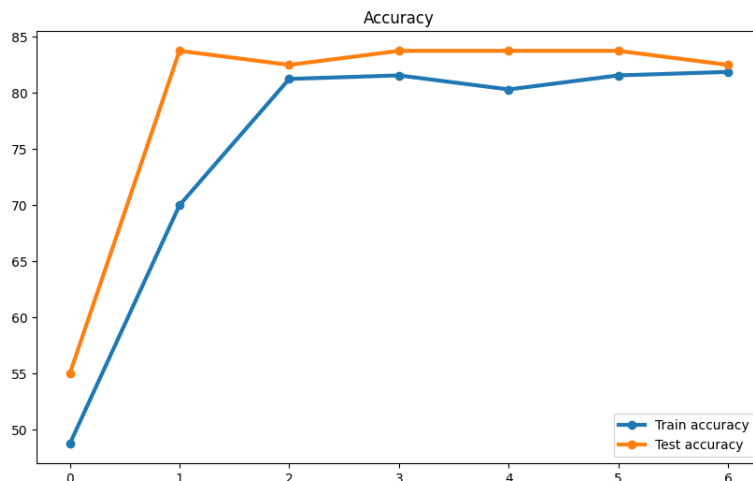
```
Properties of scaled data:
x_train shape: (320, 100, 9)
x_test shape: (80, 100, 9)

Scaled max: 2.9023424580100015
Non-Scaled max: 0.0796803573286535
```

ساختار شبکه به صورت زیر می باشد:

```
LSTMModel(
  (feature_extraction): Sequential(
    (0): LSTM(9, 128, batch_first=True)
    (1): LSTM(128, 128, batch_first=True)
    (2): LSTM(128, 64, batch_first=True)
    (3): LSTM(64, 16, batch_first=True)
  )
  (classifier): Sequential(
    (0): Linear(in_features=16, out_features=8, bias=True)
    (1): ReLU()
    (2): Linear(in_features=8, out_features=1, bias=True)
    (3): Sigmoid()
  )
)
```

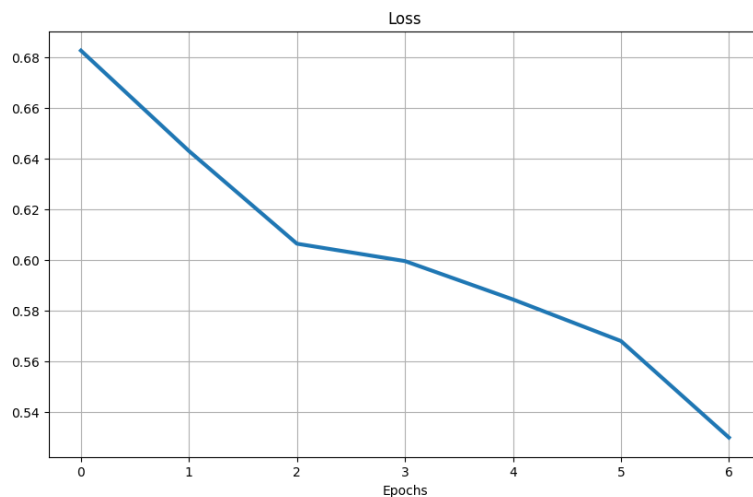
یک استخراج ویژگی ۴ لایه به همراه یک مدل طبقه بندی ۲ لایه. آموزش این شبکه چیزی حدود ۲ دقیقه به طول انجامید.



شکل ۳ نمودار دقت بدست آمده برای شبکه LSTM منبع

شکل بالا نمودار دقت بدست آمده برای آموزش مدل منبع را به نمایش گذاشته است. این آموزش برای ۳۰

ایپاک تنظیم شده بود که با استفاده از early-stopping در ایپاک ۱۵ به اتمام رسید.



شکل ۴ نمودار هزینه برای آموزش مدل LSTM دامنه منبع

آموزش تبعیض‌گر و هدف

در این مرحله پیش پردازش هایی مانند منبع روی داده های هدف انجام گرفت.

Properties of scaled data:

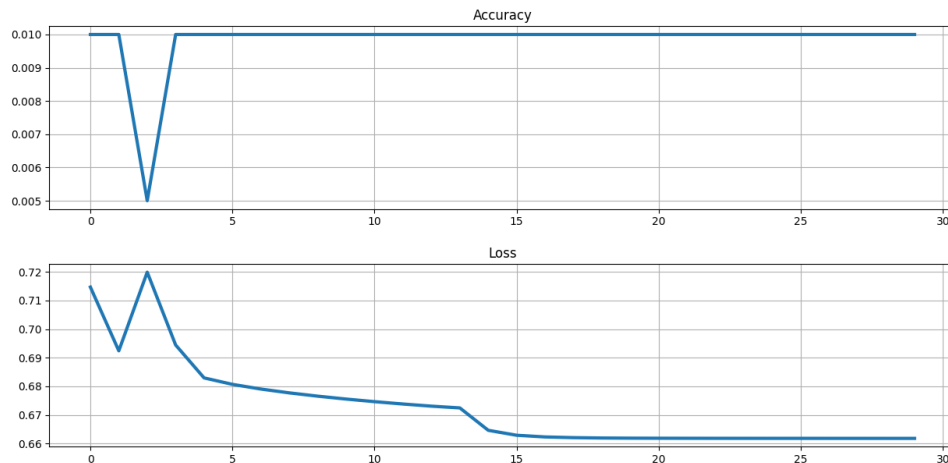
X_train shape: (320, 100, 9)

x_test shape: (80, 100, 9)

Scaled max: 2.9023424580100015

Non-Scaled max: 0.0796803573286535

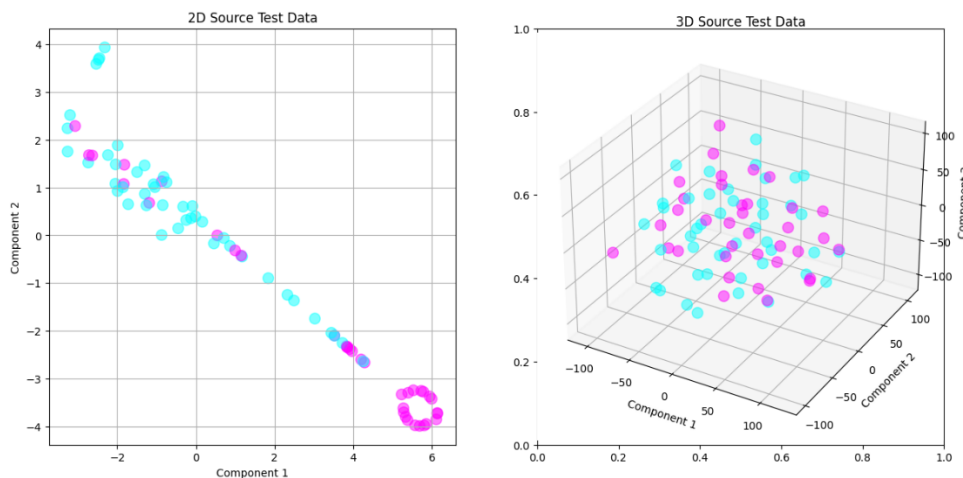
سپس به روش توضیح داده شده، شبکه های تبعیض گر و اسخراج ویژگی هدف آموزش داده شدند.



شکل ۱۵ نمودار دقت و خطا بدست آمده برای شبکه تبعیض گر

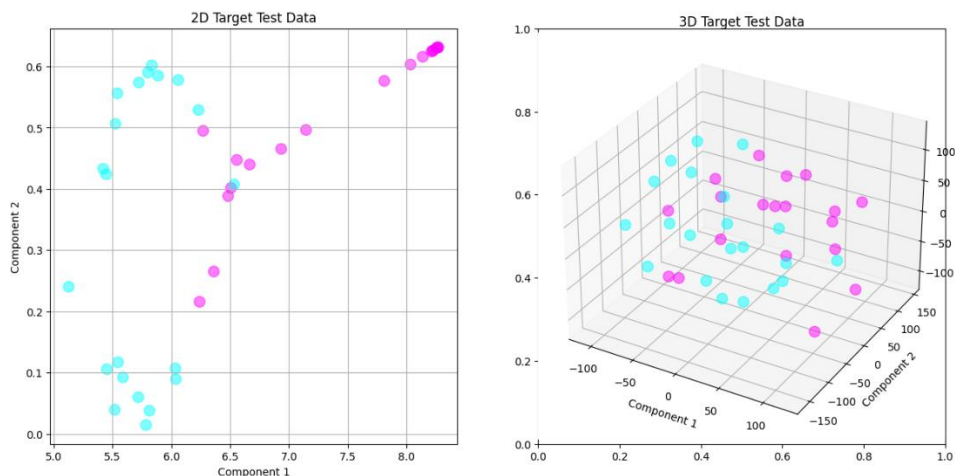
تصویر بالا نمودار خطا و دقت بدست آمده برای شبکه تبعیض گر را به نمایش گذاشته است. با این که هزینه به حالت *rest* رسیده است ولی هنوز هیچ دقت خوبی نداریم. این به این دلیل است که دامنه هدف و منبع اختلاف زیادی با هم دارند و تبعیض گر به راحتی آن ها را از یکدیگر تشخیص می دهد.

آمورش این بخش برای ۳۰ اپیاک بعد از زمان تقریبی ۳ دقیقه به اتمام رسید.



شکل ۱۶ پخش ویژگی های دامنه منبع

شکل ۱۶ پخش ویژگی های منبع را به نمایش گذاشته است. یعنی خروجی شبکه استخراج ویژگی که ۱۶ بعد دارد بدین صورت به نمایش درآمده است.



شکل ۱۷ پخش ویژگی های دامنه هدف

شکل ۱۷ نیز پخش ویژگی های دامنه هدف را به نمایش درآورده است.

این پخش ویژگی ها برای داده های آزمون می باشند و بر بروی شبکه های استخراج ویژگی هر دامنه اعمال شده اند. برای کاهش ابعاد نیز از الگوریتم t-SNE استفاده شد.

عملکرد دیتاهای منبع روی مدل های مختلف

MODEL ACCURACIES ON SOURCE DATA
 Source model & Train data: 81.9%
 Source model & Test data: 83.8%
 Target model & Train data: 51.2%
 Target model & Test data: 45.0%

عملکرد دیتاهای هدف روی مدل های مختلف

MODEL ACCURACIES ON TARGET DATA
 Source model & Train data: 75.6%
 Source model & Test data: 80.0%
 Target model & Train data: 50.6%
 Target model & Test data: 47.5%

تا به اینجای کار با تغییر مدل استخراج ویژگی هدف، هیچ بهبودی در عملکرد مدل حاصل نشد و داده های

هدف با مدل منبع و وزن های آن مدل بهتر دسته بندی می شوند.

شبکه CNN 2D

آموزش منبع

تصاویر ورودی دارای ابعاد تقریبی ۸۰۰ در ۸۰۰ پیکسل بودند. با استفاده از تابع `compose` از کتابخانه

`pytorch` تبدیل های زیر روی تصاویر انجام شد:

```
resize -> 224x224
```

```
normalization
```

که پس از اعمال این پیش پردازش ها داده ها به صورت زیر در آمدند:

```
There are 320 batches of 8 images in the source training set.
There are 80 images in the source test set.
There are 160 batches of 8 images in the target training set
There are 40 images in the target test set
>>> All images are 224x224 pixels <<<
```

ساختار شبکه به صورت زیر می باشد:

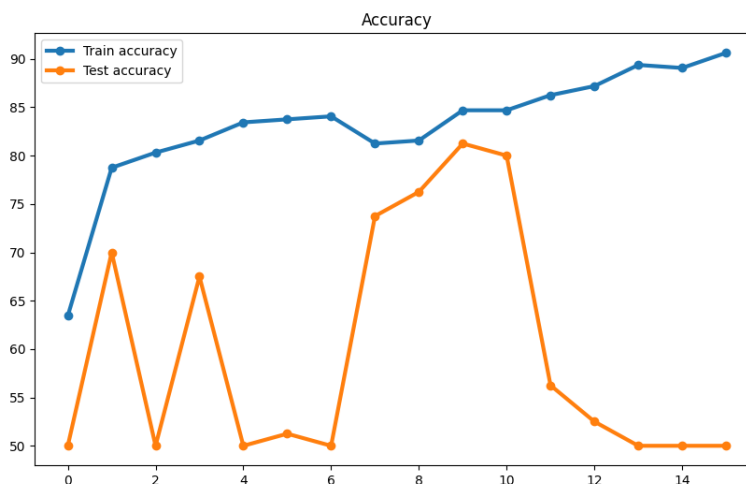
Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 8, 224, 224]	4,064
BatchNorm2d-2	[-1, 8, 224, 224]	16
MaxPool2d-3	[-1, 8, 112, 112]	0
Dropout-4	[-1, 8, 112, 112]	0
ConvLayer-5	[-1, 8, 112, 112]	0
Conv2d-6	[-1, 16, 112, 112]	6,288
BatchNorm2d-7	[-1, 16, 112, 112]	32
MaxPool2d-8	[-1, 16, 56, 56]	0
Dropout-9	[-1, 16, 56, 56]	0
ConvLayer-10	[-1, 16, 56, 56]	0
Conv2d-11	[-1, 32, 52, 52]	12,832
BatchNorm2d-12	[-1, 32, 52, 52]	64
MaxPool2d-13	[-1, 32, 26, 26]	0
Dropout-14	[-1, 32, 26, 26]	0
ConvLayer-15	[-1, 32, 26, 26]	0
Conv2d-16	[-1, 32, 24, 24]	9,248
BatchNorm2d-17	[-1, 32, 24, 24]	64
MaxPool2d-18	[-1, 32, 12, 12]	0
Dropout-19	[-1, 32, 12, 12]	0
ConvLayer-20	[-1, 32, 12, 12]	0
Linear-21	[-1, 128]	589,952
ReLU-22	[-1, 128]	0
Linear-23	[-1, 1]	129
Sigmoid-24	[-1, 1]	0

=====

Total params: 622,689
Trainable params: 622,689
Non-trainable params: 0

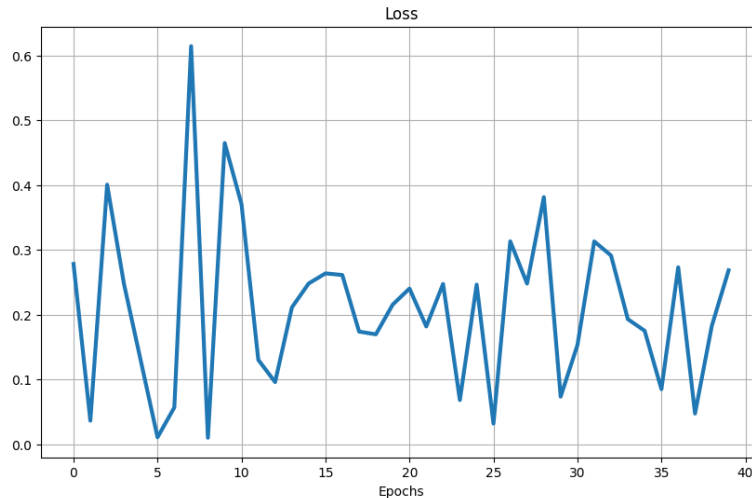
Input size (MB): 0.57
Forward/backward pass size (MB): 14.84
Params size (MB): 2.38
Estimated Total Size (MB): 17.79

شبکه استخراج ویژگی دارای ۴ لایه کانوولوشنی بود که هر کدام از این لایه ها توسط یک batchnorm و maxpool و dropout ادامه پیدا می کردند. بعد از flatten کردن آخرین لایه مدل استخراج ویژگی تعداد 4608 وارد شبکه طبقه بندی می شد. شبکه طبقه بندی هم دارای ۲ لایه بود. آموزش این مدل با ۳۰ اپیاک حدود ۴ دقیقه به طول انجامید.



شکل ۱۸ نمودار دقت بدست آمده شبکه CNN برای دامنه منبع

شکل بالا نمودار دقت بدست آمده برای آموزش مدل منبع را به نمایش گذاشته است. در این شبکه نسبت به شبکه LSTM دارای دقت بالاتری برای داده آموزش هستیم ولی از نظر دقت بدست آمده برای داده آزمون، دقت کمتر است.



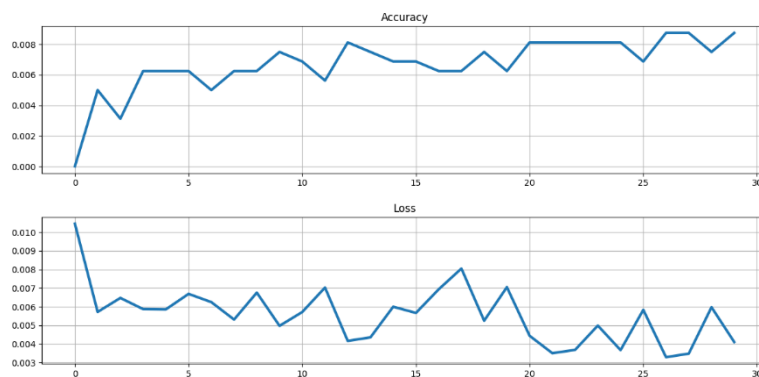
شکل ۱۹ نمودار خطا بدست آمده برای آموزش CNN

با استناد به خطا بدست آمده، خطا بسیار نا منظم در حال تغییر می باشد، شاید با تغییر ساختار شبکه بتوان این روند را اصلاح کرد.

البته باید بیان کرد که به هنگام آموزش یک شبکه ای که با تصویر کار می کند، به regularization نیاز داریم که تنها regularization استفاده شده در این آموزش dropout با نسبت 0.2 بود.

آموزش تبعیض‌گر و هدف

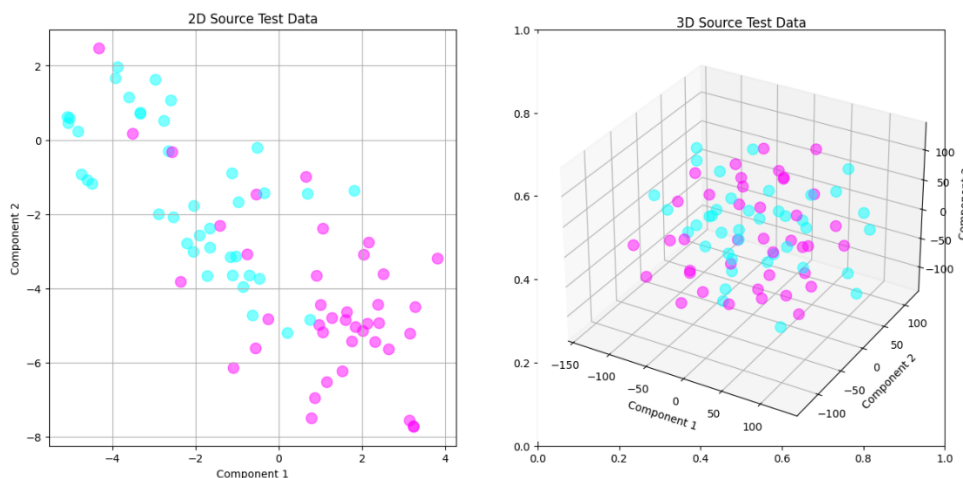
در این مرحله شبکه های تبعیض گر و اسختراج ویژگی هدف آموزش داده شدند.



شکل ۲۰ نمودار دقت و خطا بدست آمده برای شبکه تبعیض‌گر

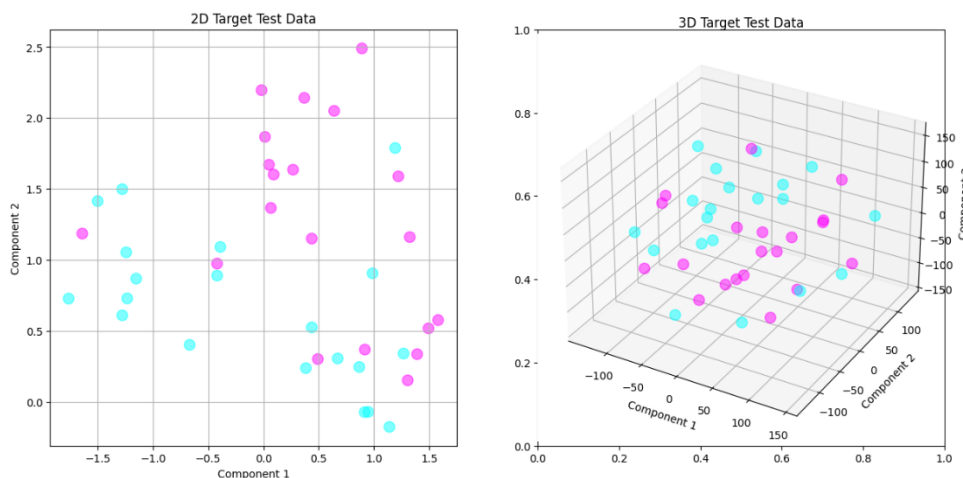
آموزش تبعیض‌گر و شبکه استخراج ویژگی هدف برای حالتی که از شبکه CNN استفاده می‌کنیم روند بهتری را از خود نشان می‌دهد. در این حالت می‌توان امیدوار بود که با آموزش بیشتر مدل بتوان به نتایج بهتری دست پیدا کرد.

آموزش این بخش برای ۳۰ اپاک بعد از زمان تقریبی ۱۰ دقیقه به اتمام رسید.



شکل ۲۱ پخش ویژگی‌های دامنه منبع

شکل ۲۱ پخش ویژگی‌های منبع را به نمایش گذاشته است. یعنی خروجی شبکه استخراج ویژگی که ۴۶۰۸ بعد دارد بدین صورت به نمایش درآمده است.



شکل ۲۲ پخش ویژگی‌های دامنه هدف

شکل ۲۲ نیز پخش ویژگی های دامنه هدف را به نمایش درآورده است.

این پخش ویژگی ها برای داده های آزمون می باشند و بر بروی شبکه های استخراج ویژگی هر دامنه اعمال شده اند. برای کاهش ابعاد نیز از الگوریتم t-SNE استفاده شد.

اگر تصاویر ۲۲ و ۱۷ را با هم مقایسه کنیم، درمیابیم که از لحاظ پخش داده برای شبکه هدف ، LSTM عملکرد بهتری دارد.

عملکرد دیتاهای منبع روی مدل های مختلف

```
MODEL ACCURACIES ON SOURCE DATA
Source model & Train data: 85.3%
Source model & Testn data: 81.2%
Target model & Train data: 85.3%
Target model & Test data: 81.2%
```

عملکرد دیتاهای هدف روی مدل های مختلف

```
MODEL ACCURACIES ON TARGET DATA
Source model & Train data: 83.1%
Source model & Testn data: 80.0%
Target model & Train data: 83.1%
Target model & Test data: 80.0%
```

با مقایسه عملکرد مدل ها روی تمامی دامنه ها می توان به این نتیجه رسید که داده ها در حوزه زمان-فرکانس بسیار به هم شباهت دارند چراکه عملکرد همه آن ها به هم شبیه می باشد.

نتیجہ گیری

عملکرد مدل‌ها

مقادیر زیر خروجی نهایی discriminator ها می باشند:

Output for LSTM model

The output of the discriminator for target data:

```
[0.00024258 0.00024259 0.00024264 0.0002426 0.0002426 0.00024263 0.00024264 0.00024263  
0.00024269 0.00024261 0.00024261 0.00024265 0.00024261 0.00024261 0.00024261 0.00024263  
0.00024258 0.00024263 0.00024256 0.00024263]
```

Output for CNN model

The output of the discriminator for target data:

```
[0.08952896 0.07534419 0.35254815 0.07778192 0.4134319 0.24437957 0.5216184 0.7009291 ]
```

همانطور که از خروجی ها مشخص می باشد، در شبکه LSTM، مدل تبعیض گر به خوبی توانسته است که داده

هدف را تشخیص دهد که این بسیار مضر است؛ چرا که شبکه استخراج ویژگی هدف نتوانسته مدل تبعیض گر را گمراه

کند. اما در حالت شبکه CNN ملاحظه می شود که برخی از داده های هدف به عنوان داده داده منبع شناخته شده اند

که این مطلوب می باشد.

دو تفاوت مهم میان آموزش مدل‌ها این است که به هنگام آموزش discriminator و feature extractor

هدف با ساختار LSTM، از وزن های رندوم برای شروع استفاده شد در حالی که برای آموزش حالت CNN، برای

شروع آموزش از وزن های مدل قبلی شروع کردیم. با توجه به عملکرد هایی که برای داده ها در هر دو روش مشاهده

شده است به نتایج زیر می رسیم:

- با شروع آموزش با وزن های رندوم، عملکرد مدل بسیار ضعیف می شود و اگر از انتقال وزن استفاده

کنیم باز هم میتوانیم عملکرد بهتری داشته باشیم. (اختلاف میان عملکرد مدل هدف با منبع در حالت

LSTM خیلی بیشتر از حالت CNN بود)

- حتی اگر شبکه تبعیض گر بتواند تفاوت میان داده هدف و منبع را تشخیص دهد، مدل باز هم می تواند

عملکرد قابل قبولی داشته باشد. (همان طور که برای هر دو نوع مشاهده شد، عملکرد داده هدف روی

شبکه منبع بهتر از عملکرد آن داده ها روی شبکه هدف بود)

- در حالت استفاده از شبکه CNN، مدل تبعیض گر سخت تر توانست بود تفاوت میان داده های منبع و هدف را تشخیص دهد. می توان از این مشاهده نتیجه گرفت که در حالت استفاده از حوزه زمان-فرکانس، داده ها شباهت بیشتری با هم دارند.
- با توجه به رفتاری که دو روش از خود نشان دادند، می توان گفت که یک شبکه CNN که از حوزه زمان-فرکانس به عنوان ورودی استفاده می کند بهتر از یک شبکه LSTM که از حوزه زمان به عنوان داده ورودی استفاده می کند عمل می کند. پس میتوان استفاده از LSTM را در اولویت پایین تری قرار داد.

پیشنهادهات

نویزگیری

برای ازبین بردن نویز و نرم تر کردن سیگنال ورودی از الگوریتم Savitzky-Golay استفاده شد. به جز این روش می توان از پیشنهادهات زیر هم استفاده نمود:

- استفاده از فیلتر فرکانسی. یعنی یک بار سیگنال را به حوزه فرکانس برده، فرکانس های با اندازه کم را • کرده و دوباره سیگنال حوزه زمان را ایجاد کنیم
- از بین بردن مودهای اولیه در حالت EMD. یعنی EMD را روی سیگنال اعمال کرده و مودهایی اولیه آن که فرکانس بالایی دارند را حذف کنیم.

نوع داده ورودی

در این پروژه از حوزه زمان و حوزه زمان-فرکانس سیگنال ارتعاشات استفاده شد. علاوه برا این ها می توان از سیگنال حوزه فرکانس هم بهره گرفت ولی از آنجایی که عیب اعمال شده دارای فرکانس ۵۰ هرتز بود در این پروژه از این حوزه استفاده نشد.

اما می توان از روش های دیگر بدست آوردن حوزه زمان-فرکانس سیگنال ارتعاشات نیز استفاده کرد. به عنوان مثال میتوان از wavelet decomposition برای ساختن حوزه زمان-فرکانس سیگنال ارتعاشات بهره برد و این نوع داده را نیز امتحان کرد.

مدل یادگیری ماشین

شاید مهمترین و مؤثرترین فعالیت ها و تغییرات را میتوان در این بخش به سرانجام رسانید. در ادامه به خلاقیت هایی که میتوان برای هر یک حوزه های سیگنال استفاده کرد، اشاره می کنیم:

حوزه زمان (فرکانس)

- استفاده از CNN ۱ بعدی
- استفاده از Graph Convolutional Neural Network
- امتحان کردن ساختار های مختلف و متفاوت

حوزه زمان-فرکانس

- استفاده از ساختار های مختلف و متفاوت

- [1] S. Liu, H. Wang, J. Tang, and X. Zhang, "Research on fault diagnosis of gas turbine rotor based on adversarial discriminative domain adaption transfer learning," *Measurement*, vol. 196, p. 111174, 2022.
- [2] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167-7176.
- [3] S. Liu, et al., "Research on fault diagnosis of gas turbine rotor based on adversarial discriminative domain adaption transfer learning," *Measurement*, 2022.