



هدف این تمرین آشنایی با شبکه‌های بازگشتی و مقایسه عملکرد آن‌ها با شبکه‌های پیشرو در مدل‌سازی پدیده‌های وابسته به زمان است. برای این کار از مجموعه داده C-MAPSS استفاده خواهیم کرد که یکی از مجموعه داده‌های پرکاربرد ناسا در حوزه پایش وضعیت موتور توربوفن هواپیما است. این مجموعه داده با استفاده از یک شبیه‌ساز پیشرفته تولید شده است و شامل شرایط کاری موتور در طول چندین پرواز پیاپی و کیفیت عملکرد موتور در این شرایط است. در هر پرواز، حسگرها پارامترهای مختلفی مانند دما، فشار و شرایط کاری موتور را با نرخ نمونه‌برداری ۱ هرتز ثبت می‌کنند. همچنین در بخشی از پروازها، به‌صورت شبیه‌سازی شده خرابی‌هایی در اجزای مختلف موتور مانند کمپرسور، توربین و فن ایجاد می‌شود تا بتوان مدل‌ها را برای پیش‌بینی خرابی‌های احتمالی و زمان وقوع آن‌ها آموزش داد.

داده‌ها به‌صورت سری زمانی هستند و هر داده شامل ۳۰ پارامتر مختلف از موتور و شرایط پرواز است. زیرمجموعه FD001 ساده‌ترین حالت این داده‌ها است که تنها شامل یک نوع خرابی است. برای کاهش پیچیدگی مساله، در ادامه این تمرین از همین زیرمجموعه استفاده خواهیم کرد. توصیه می‌شود برای انجام تمرین از محیط Google Colab یا Jupyter Notebook استفاده کنید. مجموعه داده CMAPSS و مقاله‌ای که آن را معرفی می‌کند در فایل ZIP تمرین قرار داده شده است.

[دانلود مجموعه داده C-MAPSS \(NASA Turbofan Engine\)](#)

بخش الف: پیاده‌سازی مدل‌ها

مرحله ۱: آماده‌سازی داده‌ها

بارگذاری و انتخاب داده‌ها

ابتدا زیرمجموعه FD001 را بارگذاری کنید. سپس ویژگی‌های حسگر مرتبط را انتخاب کنید و ویژگی‌هایی را که اطلاعات مفیدی ارائه نمی‌دهند حذف نمایید. برای این کار:

- ویژگی‌هایی با مقدار ثابت یا تغییرات بسیار کم (Low Variance Features) را حذف کنید (ویژگی‌هایی که تقریباً همیشه مقدار ثابتی دارند).
- ویژگی‌های تکراری یا بسیار مشابه (Duplicate or Highly Similar Features) را شناسایی و یکی از آن‌ها را نگه دارید.
- ارتباط ویژگی‌ها با متغیر هدف (Correlation with Target, Correlation Matrix) را بررسی کنید و ویژگی‌هایی با همبستگی بسیار پایین را کنار بگذارید.
- با استفاده از اهمیت ویژگی‌ها بر اساس مدل (Feature Importance, e.g., Random Forest Feature Importance)، ویژگی‌های با اهمیت پایین را حذف نمایید. (امتیازی)

تقسیم داده‌ها به آموزش و آزمون (Train/Test Split)

داده‌ها را به دو بخش ۷۰ درصدی (آموزش) و ۳۰ درصدی (آزمون) تقسیم کنید.

نرمال‌سازی داده‌ها (Normalization)

در این مرحله، همه مقادیر عددی به یک بازه یا توزیع مشخص تبدیل می‌شوند تا مدل بهتر و سریع‌تر آموزش ببیند. با استفاده از یکی از دو روش زیر، داده‌ها را نرمال‌سازی کنید:

- Min-Max: انتقال داده‌ها به بازه‌ای مشخص مثل ۰ تا ۱
- Z-score: استانداردسازی به‌طوری‌که داده‌ها میانگین صفر و انحراف معیار یک داشته باشند

تقسیم‌بندی داده‌ها با روش پنجره لغزان (Sliding Window)

روش «پنجره لغزان» برای تقسیم داده‌های سری زمانی به قسمت‌های کوچک‌تر و هم‌اندازه به کار می‌رود. یک پنجره (مثلاً شامل ۳۰ چرخه پیاپی) را به تدریج روی داده‌ها حرکت دهید و در هر مرحله، داده‌های درون پنجره را استخراج کنید. این روش:

- داده‌ها را به نمونه‌هایی با طول یکسان تبدیل می‌کند که برای مدل‌های CNN و LSTM ضروری است.
- به مدل امکان می‌دهد الگوها و تغییرات کوتاه‌مدت در داده‌ها را بهتر تشخیص دهد.
- تعداد نمونه‌های قابل استفاده را افزایش می‌دهد و یادگیری مدل را بهبود می‌بخشد.

برچسب‌گذاری پنجره‌ها با مقدار RUL

برای هر پنجره استخراج شده، یک برچسب "عمر باقی‌مانده RUL: Remaining Useful Life" تعیین کنید. برای این کار:

- برای هر لحظه (چرخه) در سری زمانی، حساب کنید که چند چرخه تا پایان عمر موتور باقی مانده است.
- برای هر پنجره، مقدار RUL مربوط به آخرین چرخه آن پنجره را انتخاب کرده و به کل پنجره نسبت دهید. به این ترتیب مدل یاد می‌گیرد که با مشاهده کل یک پنجره پیش‌بینی کند که در انتهای آن پنجره چقدر از عمر موتور باقی مانده است.
- اگر مقادیر RUL خیلی بزرگ باشد ممکن است آموزش مدل با چالش روبرو شود. به همین دلیل می‌توانید یک حد بالا (مثلاً ۱۳۰ چرخه) برای RUL تعیین کنید و همه مقادیر بزرگ‌تر از آن را برابر با این حد قرار دهید. این کار به مدل کمک می‌کند روی پیش‌بینی‌های واقع‌بینانه‌تر تمرکز کند و دقت پیش‌بینی افزایش یابد.

مرحله ۲: پیاده‌سازی شبکه عصبی پیچشی (CNN)

برای پیش‌بینی باقی‌مانده عمر مفید بر اساس داده‌های سری‌زمانی از یک CNN یک بعدی با معماری زیر استفاده کنید:

ورودی با شکل (اندازه پنجره، تعداد ویژگی‌ها):

Conv1D با ۶۴ فیلتر، کرنل ۳، فعال‌سازی ReLU، پدینگ

MaxPooling1D با اندازه پنجره ۲

Conv1D با ۱۲۸ فیلتر، کرنل ۳، فعال‌سازی ReLU، پدینگ

GlobalAveragePooling1D
Dense با ۶۴ نورون، فعال سازی ReLU
Dropout با نرخ ۰/۲
Dense با یک نورون برای خروجی RUL

آموزش:

- تابع خطا: میانگین مربعات خطا (MSE)
- بهینه ساز: Adam با نرخ یادگیری ۰/۰۰۱
- اندازه دسته (batch size): ۳۲ یا ۶۴
- تعداد دوره (epochs): ۵۰ تا ۱۰۰
- معیار ارزیابی: خطای میانگین قدر مطلق (MAE) و به صورت اختیاری log RMSE

مرحله ۳: پیاده سازی شبکه LSTM

برای پیش بینی باقی مانده عمر مفید بر اساس داده های سری زمانی از یک شبکه استاندارد LSTM با معماری زیر استفاده کنید:
ورودی با شکل (اندازه پنجره، تعداد ویژگی ها):

LSTM با ۱۰۰ واحد، بازگرداندن توالی
Dropout با نرخ ۰/۲
LSTM با ۵۰ واحد، فقط خروجی آخرین گام
Dense با ۶۴ نورون، فعال سازی ReLU
Dropout با نرخ ۰/۲
Dense با یک نورون برای مقدار RUL

کامپایل و آموزش:

- تابع خطا: MSE
- بهینه ساز: Adam با نرخ یادگیری ۰/۰۰۱
- اندازه دسته و تعداد دوره مشابه مدل CNN برای فراهم شدن امکان مقایسه

مرحله ۴: تنظیم ابر پارامترها (Hyperparameter Tuning)

پارامترهای زیر را با جستجوی دستی یا با استفاده از ابزارهایی مانند KerasTuner یا Optuna تنظیم کنید:

- نرخ یادگیری
- اندازه دسته (batch size)
- تعداد فیلترها یا واحدها
- Dropout نرخ

- اندازه پنجره
- نوع بهینه‌ساز (مقایسه Adam و RMSprop)

گزارش:

- کدام پارامترها بیشترین تأثیر را بر عملکرد مدل داشتند؟
- جدولی یا گزارشی از اجرای مدل با تنظیمات مختلف و نتایج ارائه دهید.
- درباره تعادل بین زمان آموزش، دقت و پیچیدگی مدل بحث کنید.

مرحله ۵: پیاده‌سازی مدل ترکیبی CNN + LSTM

این مدل ترکیبی با معماری زیر از توانایی CNN در استخراج ویژگی‌ها و توانایی LSTM در یادگیری زمانی بهره می‌برد.

ورودی با شکل (اندازه پنجره، تعداد ویژگی‌ها):

Conv1D با ۶۴ فیلتر، کرنل ۳، فعال‌سازی ReLU، پدینگ مساوی

MaxPooling1D با اندازه ۲

LSTM با ۱۰۰ واحد، بدون بازگرداندن توالی

Dropout با نرخ ۰/۳

Dense با ۶۴ نورون، فعال‌سازی ReLU

Dense با یک نورون برای خروجی RUL

این مدل را مشابه سایر مدل‌ها آموزش داده و ارزیابی کنید. می‌توانید ترتیب CNN و LSTM را نیز عوض کنید و نتیجه را بررسی نمایید. (امتیازی)

مرحله ۶: پیاده‌سازی معماری LSTM + ATTENTION

برای آشنایی با سازوکار توجه (Attention)، یک مدل شبکه عصبی LSTM با سازوکار توجه طراحی و پیاده‌سازی کنید. سپس این مدل را روی همان مجموعه داده آموزش دهید. شبکه LSTM قادر است وابستگی‌های زمانی و الگوهای بلندمدت را به خوبی یاد بگیرد، اما ممکن است به تنهایی تمام بخش‌های سری زمانی را به طور یکسان مورد توجه قرار دهد. سازوکار توجه به مدل اجازه می‌دهد که روی بخش‌های مهم‌تر داده‌های سری زمانی تمرکز کند و وزن کمتری به بخش‌های کم‌اهمیت اختصاص دهد.

این معماری ویژگی‌های جالبی دارد، از جمله:

- افزایش دقت پیش‌بینی: به دلیل توجه هدفمند به بخش‌های حیاتی و تعیین‌کننده در سری زمانی.
- تفسیرپذیری بیشتر: به کمک وزن‌هایی که مکانیزم توجه تولید می‌کند، می‌توان مشخص کرد کدام قسمت از داده‌ها بیشترین تأثیر را در پیش‌بینی دارد.
- بهبود یادگیری وابستگی‌های بلندمدت و ظریف که ممکن است در معماری‌های ساده‌تر به خوبی یاد گرفته نشوند.
- عملکرد بهتر در داده‌های پیچیده‌ای که بخش‌هایی با اهمیت متفاوت دارند و نیازمند توجه گزینشی هستند.

بخش ب: ارزیابی عملکرد مدل‌ها

۱- نمودارها و مقایسه بصری

- منحنی‌های خطای آموزش و اعتبارسنجی مثلاً MSE را برای هر مدل رسم کنید.
- مقدار واقعی در برابر مقدار پیش‌بینی شده RUL را روی داده‌های تست برای مدل‌ها رسم کنید

۲- معیارهای ارزیابی مدل

برای هر مدلی که آموزش داده‌اید معیارهای زیر را محاسبه کنید و نتایج را به صورت جدول ارائه دهید تا مقایسه مدل‌ها ساده‌تر انجام شود:

- ریشه میانگین مربعات خطا: (RMSE)
- ارزیابی عملکرد مدل با تأکید بر خطاهای بزرگ‌تر.
- میانگین قدر مطلق خطا: (MAE)
- نشان‌دهنده میانگین خطای مطلق پیش‌بینی‌ها است.
- R^2 ضریب تعیین (Coefficient of Determination)
- ارزیابی میزان تطابق بین مقادیر واقعی و پیش‌بینی شده (هرچه نزدیک‌تر به ۱ بهتر است).
- زمان آموزش مدل:
- زمان صرف‌شده برای آموزش مدل (بر حسب ثانیه یا دقیقه).
- معیارهای تکمیلی (اختیاری برای یادگیری عمیق‌تر دانشجویان):

- خطای درصد مطلق میانگین: (MAPE)
- معیاری جهت مقایسه خطا به صورت درصدی که برای ارزیابی دقیق‌تر کاربرد دارد.
- منحنی خطای تجمعی: (Cumulative Error Curve)
- تحلیل نحوه توزیع خطاها و بررسی این که چه تعداد پیش‌بینی‌ها خطای کمتری نسبت به یک حد معین دارند.
- نتایج را به صورت جدولی برای مقایسه راحت‌تر ارائه دهید.

۳. تحلیل عملکرد (خلاصه)

به سوالات زیر پاسخ دهید:

- کدام مدل دقیق‌ترین پیش‌بینی را ارائه داد؟ به نظر شما دلیل عملکرد بهتر آن چه بود؟
- آیا هیچ یک از مدل‌ها نشانه‌هایی از بیش‌برازش (Overfitting) داشت؟ چگونه متوجه شدید؟
- درباره تعادل بین دقت (Accuracy) و زمان آموزش بحث کنید.

چند تذکر:

- برای یادگیری مفاهیمی که در تمرین مطرح شده و در کلاس تدریس نشده‌اند از منابع موجود در اینترنت استفاده کنید.

- برای انجام بخش‌های مختلف تمرین می‌توانید از کتابخانه‌های آماده پایتون استفاده کنید.
- تحویل گزارش این تمرین ضروری است و به تمرین بدون گزارش نمره‌ای تعلق نمی‌گیرد. حجم گزارش معیاری برای ارزیابی نخواهد بود و لزومی به توضیحات کد نیست؛ اما تحلیل نتایج الزامی است، حتی اگر در صورت پرسش اشاره‌ای به آن نشده باشد.
- در فرآیند ارزیابی گزارش، کدهای شما لزوماً اجرا نخواهد شد. بنابراین همه نتایج و تحلیل‌های خود را به طور کامل ارائه کنید.
- شباهت بیش از حد گزارش‌ها و کدها باعث از دست دادن نمره تمرین خواهد شد. همچنین گزارش‌هایی که در آن‌ها از کدهای آماده استفاده شده باشد پذیرفته نخواهد شد.
- تنها زبان برنامه نویسی مجاز Python است.
- کدها حتماً باید در قالب نوت‌بوک با پسوند ipynb تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتماً در فایل ارسالی ذخیره شود. برای مثال اگر خروجی یک سلول یک نمودار است، این نمودار باید هم در گزارش و هم در نوت‌بوک کدها وجود داشته باشد.
- لطفاً گزارش، کدها و سایر ضمایم را در یک پوشه با نام زیر قرار داده و پس از فشردن دکمه بارگذاری Elearn بارگذاری کنید.

HW5_Last Name_Student Number.zip

- پرسش‌های خود را از طریق ایمیل یا تلگرام از دستیار آموزشی مربوطه بپرسید:

تلگرام	ایمیل	
@arshiaeft	arshiaeftekhari@ut.ac.ir	ارشیا افتخاری زاده