



به نام خدا
دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر



درس شبکه‌های عصبی و یادگیری عمیق

تمرین شماره چهار

طراحان تمرین:

سوال یک: علی رمضانی

سوال دو: احسان فروتن

مهلت ارسال: ۳۰ آذر ۱۴۰۴

4	سوال ۱. مدل‌های RNN برای وظایف مشترک درک زبان گفتاری
5	۱-۱. آماده‌سازی داده (۱۲ نمره)
6	۲-۱. مدل‌های پایه (۱۶ نمره)
8	۳-۱. مدل BiLSTM joint (۲۴ نمره)
8	۴-۱. مدل Encoder-Decoder non-aligned joint (۴۸ نمره)
11	سوال ۲. پیش‌بینی قیمت سهام با استفاده از معماری‌های Recursive
11	۱-۲. Data Preprocessing (۲۰ نمره)
12	۲-۲. مدل پایه (۱۰ نمره)
13	۳-۲. مدل‌های Recursive Neural Network (۵۰ نمره)
13	۱-۳-۲. معماری LSTM
13	۲-۳-۲. معماری GRU
14	۴-۲. مقایسه روی داده‌های تست و مقایسه‌های عملکرد (۲۰ نمره)
14	۵-۲. بخش امتیازی (۵ نمره)

جدول‌ها

7	جدول 1. معماری شبکه BiRNN
7	جدول 2. هایپرپارامترهای مدل
8	جدول 3. معماری شبکه BiLSTM
9	جدول 4. معماری انکودر
9	جدول 5. معماری دیکودر
9	جدول 6. هایپرپارامترهای شبکه انکودر-دیکودر

سوال ۱. مدل‌های RNN برای وظایف مشترک درک زبان گفتاری

در این تمرین، به بررسی مسئله درک زبان گفتاری (SLU) می‌پردازیم که شامل دو زیرمسئله‌ی مهم است: تشخیص نیت (Intent Detection) و شناسایی اسلات‌ها (Slot Filling). این دو وظیفه در سیستم‌های گفت‌وگو، مانند دستیارهای صوتی، نقش حیاتی دارند. هدف از این تمرین پیاده‌سازی مدل‌های مبتنی بر شبکه‌های عصبی بازگشتی برای این دو وظیفه است.

در سیستم‌های درک زبان گفتاری دو وظیفه‌ی اصلی وجود دارد که تقریباً تمام سیستم‌های گفت‌وگو، دستیارهای صوتی و چت‌بات‌ها بر اساس آن‌ها کار می‌کنند: تشخیص نیت و شناسایی اسلات‌ها. در ادامه این دو تسک و همچنین چند نمونه از دیتاست ATIS و نحوه‌ی برچسب‌گذاری BIO توضیح داده می‌شود.

تشخیص نیت به این معناست که مدل باید تشخیص دهد کاربر قصد انجام چه کاری دارد. جمله‌ی کاربر می‌تواند پیچیده باشد، اما در نهایت معمولاً یک هدف اصلی دارد. برای مثال، در جمله‌ی «show me flights from Boston to Denver» نیت کاربر درخواست اطلاعات پرواز است و بنابراین لیبل آن `atis_flight` خواهد بود. دیتاست ATIS شامل حدود ۱۷ نیت مختلف است، مانند `atis_flight`, `atis_airline`, `atis_flight_time` و غیره. در کنار تشخیص نیت، وظیفه‌ی دوم شناسایی اسلات‌ها است که شامل استخراج بخش‌های معنایی مهم از جمله است. سیستم باید بفهمد کدام کلمات نماینده‌ی موجودیت‌های مهم هستند؛ مثلاً شهر مبدأ، مقصد، تاریخ، زمان، ایرلاین و غیره. این کار به‌صورت توکن‌به‌توکن انجام می‌شود؛ یعنی برای هر کلمه یک برچسب معنایی اختصاص می‌یابد.

در دیتاست ATIS، اسلات‌ها با فرمت رایج BIO برچسب‌گذاری شده‌اند. این روش سه نوع برچسب دارد:

B: آغاز یک اسلات (Begin)

I: ادامه‌ی همان اسلات (Inside)

O: کلمه‌ای که به هیچ اسلاتی تعلق ندارد (Outside)

یک مثال از دیتاست در ادامه آمده است.

Input:

BOS what is the cost of a round trip flight from pittsburgh to
atlanta beginning on april twenty fifth and returning on may sixth
EOS

Slot labels:

O O O O O O O B-round_trip I-round_trip O O B-fromloc.city_name O B-
toloc.city_name O O B-depart_date.month_name
B-depart_date.day_number I-depart_date.day_number O O O
B-return_date.month_name B-return_date.day_number O

Intent label: `atis_airfare`

مبنای این تمرین بر اساس [این مقاله](#) است. ابتدا مقاله را با دقت مطالعه کنید تا با مفاهیم کلیدی، معماری‌ها و اصطلاحات رایج در این حوزه آشنا شوید. توجه داشته باشید که اطلاعات لازم برای پیاده‌سازی، از جمله هایپرپارامترها، در متن تمرین به‌طور دقیق آورده شده‌اند و نیازی به مراجعه به مقاله برای استخراج آن‌ها نیست.

در این تمرین، از یک نسخه ساختاریافته از [دیتاست ATIS](#) استفاده می‌کنیم که شامل پرسش‌هایی درباره‌ی پروازهای هوایی و برجسب‌های معنایی مرتبط با آن‌هاست. با استفاده از این داده، چند معماری مختلف پیاده‌سازی خواهیم کرد، این دیتاست از کگل قابل دریافت است، برای دریافت می‌توانید از کد زیر استفاده کنید.

```
import kagglehub
path = kagglehub.dataset_download("siddhadev/atis-dataset-clean")
print("Path to dataset files:", path)
```

۱-۱. آماده‌سازی داده (۱۲ نمره)

در این بخش ابتدا با دو اصطلاح مهم توکنایز کردن و ساختن واژگان آشنا می‌شوم.

توکنایز کردن دادگان (Tokenization): پس از بارگزاری دادگان ATIS، اولین گام آماده‌سازی داده، توکنایز کردن جملات است. توکنایز کردن یعنی تبدیل یک جمله‌ی متنی به توکن‌ها (معمولاً کلمه‌ها). در این پروژه نیازی به استفاده از توکنایزهای پیشرفته مانند BERT، SentencePiece یا WordPiece نیست. بنابراین ساده‌ترین و رایج‌ترین روش، توکنایز بر اساس فاصله‌گذاری (white space) است.

به مثال زیر توجه کنید.

جمله: show me flights from Boston to Denver

توکن‌ها: ["show", "me", "flights", "from", "Boston", "to", "Denver"]

مزیت این روش برای ATIS این است که داده‌ها از قبل تمیز و یکدست هستند و اغلب کلمات، معنای کاملی دارند.

توکنایز کردن برای اسلات‌ها نیز مشابه است؛ چون تعداد اسلات‌ها با تعداد توکن‌ها کاملاً یکسان است.

ساختن واژگان (Building Vocab): پس از توکنایز کردن، باید یک واژگان (vocab) برای مدل بسازیم. واژگان مجموعه‌ای از تمام توکن‌های منحصر به فردی است که در داده مشاهده می‌کنیم. ما در این پروژه باید سه واژگان بسازیم:

- واژگان کلمات (Word Vocabulary)
- واژگان اسلات‌ها (Slot Vocabulary)
- واژگان نیت‌ها (Intent Vocabulary)

برای هر سه مورد، روندی مشابه انجام می‌شود:

- تمام جملات آموزش را توکنایز می‌کنیم.

- تمام توکن‌ها را در یک مجموعه می‌ریزیم.
- توکن‌های منحصربه‌فرد را استخراج می‌کنیم.
- سپس آن‌ها را به فهرست تبدیل می‌کنیم.

همین کار برای تمام برچسب‌های Slot و Intent انجام می‌شود.

۱-۱-۱. دیتاست را لود کنید و تعداد داده‌های آموزشی و تست را گزارش کنید، ۱۰ درصد از داده آموزش را برای ولیدیشن جدا کنید.

۱-۱-۲. ابتدا دادگان را بررسی کنید و یک نمونه از داده‌های آموزشی را نمایش دهید، این نمایش باید شامل جمله اصلی، برچسب‌های اسلات و همچنین برچسب نیت باشد.

۱-۱-۳. دادگان را توکنایز کنید، در این بخش لازم نیست از توکنایزر خاصی استفاده کنید و تنها جداسازی بر اساس white space کفایت می‌کند.

۱-۱-۴. واژگان (vocab) خود را بسازید و تعداد واژگان برای جملات آموزش، اسلات‌ها و نیت را نمایش دهید. دقت کنید که باید دو توکن <PAD> و <UNK> را نیز به واژگان خود اضافه کنید. بررسی کنید که این دو توکن در چه مواقعی استفاده می‌شوند.

۱-۱-۵. کلاس‌های دیتاست و دیتالودر خود را بسازید، دقت کنید که برای پد کردن جملات یک راه پد کردن جملات به اندازه بزرگترین جمله دیتاست است و راه دیگر پد کردن جملات به اندازه بزرگترین جمله هر Batch، پایتورچ برای چنین کاری از تابعی خاص (collate function) استفاده می‌کند که باید در ساخت دیتالودر از آن استفاده کنید.

۱-۱-۶. از بیچ اول و دوم داده‌های آموزش، ۲ جمله را که پد شده اند به صورت Token id و دیکود شده آن‌ها (با کلمات) نمایش دهید و طول جملات این دو بیچ را با هم مقایسه کنید، آیا تغییری دارند؟ فایده این کار نسبت به پد کردن به اندازه بزرگترین جمله چیست؟

۱-۲. مدل‌های پایه (۱۶ نمره)

۱-۲-۱. آموزش مدل

در این بخش لازم است دو شبکه مجزا BiRNN (Bi-directional RNN) یکی برای تشخیص نیت و یکی برای تشخیص اسلات‌ها طراحی کنید، معماری شبکه‌ها و هایپارامترهای مورد نیاز به صورت جدول ۱ است.

دقت کنید برای تشخیص نیت‌ها، خروجی مدل باید یک بردار به اندازه تعداد نیت‌ها باشد و برای تشخیص اسلات‌ها، خروجی مدل باید برای هر توکن یک برچسب اسلات تولید کند، همچنین تابع هزینه را روی توکن‌های پد شده حساب نکنید.

جدول 1. معماری شبکه BiRNN

Architecture	
Layer	#
Embedding	1
RNN	1
Dropout	1
Linear	1

جدول 2. هایپرپارامتر های مدل

Hyper parameters	
EMBED_DIM	128
HIDDEN_DIM	128
BIDIRECTIONAL	True
DROPOUT	0.5
BATCH_SIZE	32
NUM_EPOCHS	10
LEARNING_RATE	0.001
Loss	Cross Entropy
Optimizer	Adam

۱-۲-۲. ارزیابی مدل

- نمودار تغییرات تابع هزینه و دقت را روی داده آموزش و ولیدیشن را در طول آموزش رسم کنید.
- دقت (Accuracy) تشخیص نیت روی داده تست را چقدر شد. برای گرفتن نمره کامل باید به بالای ۸۰ درصد برسد.
- با استفاده از کتابخانه sequeval، مقادیر F1 score و Classification Report روی تشخیص اسلات ها را گزارش کنید، چرا این کتابخانه نسبت به sklearn در این تسک ترجیح داده میشود. برای دریافت نمره کامل مقدار F1 به بالای ۸۰ برسد.

۱-۳. مدل BiLSTM joint (۲۴ نمره)

۱-۳-۱. آموزش مدل

در این بخش قرار است، یک شبکه BiLSTM را به صورت joint روی هر دو تسک آموزش دهیم، به عبارتی تنها یک شبکه داریم اما دو head مجزا که یکی وظیفه تشخیص نیت و دیگری تشخیص اسلات ها را برعهده دارند، معماری شبکه به صورت زیر است:

جدول 3. معماری شبکه BiLSTM

Architecture	
Layer	#
Embedding	1
LSTM	1
Dropout	1
Intent head: Linear	1
Slot head: Linear	1

برای تشخیص نیت ها، خروجی مدل باید یک بردار به اندازه تعداد نیت ها باشد و برای تشخیص اسلات ها، خروجی مدل باید برای هر توکن یک برچسب اسلات تولید کند. دقت کنید که همه هایپرپارامترها دقیقاً مثل بخش قبلی باشند؛ تابع هزینه مدل را به عنوان مجموع دو تابع هزینه - یکی برای اسلات ها و دیگری برای تشخیص نیت - با ضرایب برابر در نظر بگیرید.

۱-۳-۲. ارزیابی مدل

- نمودار تغییرات تابع هزینه و دقت روی داده آموزش و ولیدیشن را در طول آموزش رسم کنید.
- دقت (Accuracy) تشخیص نیت روی داده تست چقدر شد. برای گرفتن نمره کامل باید به بالای ۸۰ درصد برسد.
- با استفاده از کتابخانه seqeval، مقادیر F1 score و Classification report را روی تشخیص اسلات ها گزارش کنید. برای گرفتن نمره کامل، مقدار F1 به بالای ۸۰ برسد.

۱-۴. مدل Encoder-Decoder non-aligned joint (۴۸ نمره)

۱-۴-۱. ساختار مدل

در این بخش قرار است، یک شبکه Encoder-Decoder را به صورت joint روی هر دو تسک آموزش دهیم، در اینجا نیز یک شبکه داریم اما دو head مجزا که یکی وظیفه تشخیص نیت و دیگری تشخیص اسلات ها را

برعهده دارند، اساسا چه تفاوتی بین چنین شبکه ای با یک شبکه BiLSTM وجود دارد؟ به عبارتی وجود دیکورد چه قابلیت به شبکه افزوده است؟

این شبکه اسلات ها را مانند یک جمله میبیند که باید آن را تولید کند، به همین منظور لازم است که از توکن های <BOS> و <EOS> برای اسلات ها نیز استفاده کنید، دادگان را برای این شبکه متناسب سازی کنید و بگویید چرا این توکن ها لازم هستند.

۱-۴-۲. آموزش مدل

معماری شبکه به صورت زیر است:

جدول 4. معماری انکودر

Encoder Architecture	
Layer	#
Embedding	1
LSTM	1

جدول 5. معماری دیکودر

Decoder Architecture	
Layer	#
Embedding	1
LSTM	1
Dropout	1
Intent head: Linear	1
Slot head: Linear	1

هایپرپارامتر های آموزش شبکه نیز تماما مانند قسمت قبل است به جز مواردی که در جدول ۶ ذکر شده است.

جدول 6. هایپرپارامتر های شبکه انکودر-دیکودر

Hyper parameters	
Encoder EMBED_DIM	128
Encoder HIDDEN_DIM	128
Decoder EMBED_DIM	64
Decoder HIDDEN_DIM	256

۱-۴-۳. ارزیابی مدل

- نمودار تغییرات تابع هزینه و دقت روی داده آموزش و ولیدیشن را در طول آموزش رسم کنید.
 - دقت (Accuracy) تشخیص نیت روی داده تست چقدر شد. برای گرفتن نمره کامل باید به بالای 80 درصد برسد.
 - با استفاده از کتابخانه sequeval، مقادیر F1 score و Classification Report روی تشخیص اسلات ها را گزارش کنید. برای دریافت نمره کامل، مقدار F1 به بالای ۵۰ درصد برسد.
- چرا عملکرد این شبکه روی تشخیص اسلات ها ضعیف تر از شبکه های قبلی است؟ ۵ جمله از داده های تست را به شبکه بدهید و خروجی دیکودر با روش greedy دیکود کرده و با برچسب های واقعی مقایسه کنید.
- در مقاله برای رفع این مشکل یک روش (به غیر از مکانیزم توجه) پیشنهاد شده است، آن روش چیست و چگونه کار میکند؟

سوال ۲. پیش بینی قیمت سهام با استفاده از معماری های Recursive

در این تمرین، هدف بررسی و مقایسه عملکرد معماری های مختلف یادگیری ماشین و یادگیری عمیق در مسئله پیش بینی سری های زمانی چند بعدی است. سری زمانی انتخاب شده برای این تمرین را می توانید با کمک کتابخانه yahoo finance و با دستور زیر به سادگی وارد کنید:

```
import yfinance as yf
df = yf.download("^GSPC", start="2000-01-01", auto_adjust=False)
```

با این کار داده ها از سال ۲۰۰۰ به بعد را وارد مدل می کنید. نام این شاخص سهام، S&P500 بوده که شاخص بورس برای ۵۰۰ شرکت مانند Google، Amazon، Apple، Fox Cooperation و دیگر کمپانی های بزرگ می باشد. همچنین این داده ها به صورت روزانه استخراج شده اند.

ساختار این تمرین بر اساس رویکرد مطرح شده در [مقاله](#) پیوست شده است. در این کار ابتدا یک خط مبنا با مدل های کلاسیک ایجاد می شود و سپس چندین معماری مبتنی بر recursion با یکدیگر مقایسه می شوند. به طور کلی، پیش بینی عملکرد سهام اولین مرحله از Quantitative Analysis در کاربردهای Finance است و با کمک آن به حل مسائل کلی تر مانند Portfolio optimization و یا risk management در trading پرداخته می شود. در ادامه به حل سوال پیش بینی سهام S&P 500 پرداخته شده است.

۲-۱. Data Preprocessing (۲۰ نمره)

۲-۱-۱. در مرحله نخست، مجموعه داده را بارگذاری کنید و ویژگی های روز و ماه را با تبدیلات سینوسی و کسینوسی به صورت چرخه ای دریاورید. همچنین گزارش کنید چه ویژگی هایی در این مجموعه داده وجود دارد و برای این ویژگی ها نمودار زمانی در کل بازه را رسم کنید. همچنین مقدار سال ها را با استفاده از MinMax Normalization به صورت نرمال دریاورید. توضیح دهید چرا باید این گونه عمل شود.

۲-۱-۲. با استفاده از کتابخانه statsmodels، نمودارهای Autocorrelation Function (ACF) و Partial Autocorrelation Function (PACF) را برای یکی از ویژگی های قیمت مانند Open Price یا Close Price رسم کنید. براساس این دو نمودار بررسی کنید که چگونه می توان طول پنجره ی ورودی و افق پیش بینی را تعیین کرد. سپس یک بار سیگنال را تفاضل گیری کنید و دوباره نمودارهای ACF و PACF را رسم نمایید. نتایج را تحلیل کنید و توضیح دهید تفاضل گیری چه تغییری در رفتار سری زمانی ایجاد می کند و این تغییر از نظر مدل سازی چه مفهومی دارد. در ادامه، درباره ی مفهوم ایستاکردن (stationarizing) سیگنال تحقیق کنید و با انجام آزمون Augmented Dickey-Fuller (ADF) تعیین کنید که سری زمانی شما در کدام مرتبه از تفاضل گیری، stationary می شود.

۲-۱-۳. تحقیق کنید پروسه ergodic چیست. آیا برای مدل های یادگیری عمیق خوب است سیگنال را ergodic و یا stationary کنیم؟ آیا قیمت های Open و یا Close پروسه ergodic می شوند؟

۲-۱-۴. داده ها را با استفاده از نمودار های بالا و با توجه به تعداد lag ها با مقدار قابل توجه، به دو پنجره ورودی و خروجی تقسیم بندی بکنید و در نظر بگیرید که برای پنجره ورودی، سائز پنجره حداکثر برابر با تعداد تفاضل برای ایستا کردن سیگنال ضرب شده در تعداد lag های با مقدار قابل توجه در حالت ایستایی خواهد بود و برای خروجی، حداکثر به تعداد lag های قابل توجه در حالت ایستا می توان طول پنجره را در نظر گرفت. توزیع های ورودی ها و خروجی ها را در یک ویژگی با کمک رسم histogram نشان بدهید. داده ها را به سه قسمت ۷۰ درصدی آموزش، ۲۰ درصدی Validation و ۱۰ درصدی test تقسیم بندی کنید. سپس با کمک Z-score Normalization و با توجه به اینکه نباید Data leakage داشته باشید، نرمال سازی هر سه مجموعه را انجام دهید. توضیح دهید چرا اینگونه عمل کردید. همچنین histogram پنجره های خروجی ۳ مجموعه را در یک شکل نمایش دهید. در آخر برای آموزش، در هنگام تعریف Loader ها، Shuffling داشته باشید. هایپرپارامتر ها مانند batch size را از مقاله و یا مقدار منطقی برای سرعت خوب آموزش انتخاب کنید. دقت داشته باشید که داده های ابتدای ۲۰۲۴ تا انتهای ۲۰۲۵ را حتما در تست خود در نظر بگیرید و باقی تست را تا ۱۰ درصد از کل باقی مانده سال ها به سمت عقب انتخاب کنید.

۲-۲. مدل پایه (۱۰ نمره)

در کار های finance، برای ارزیابی توانایی تعمیم پذیری، همواره نیازمند یک baseline برای مقایسه هستیم تا توجیه پردازش و پیچیدگی ساختار را داشته باشیم. مدل baseline باید مدلی باشد که در عین سادگی، بتواند به حداقل دقت قابل قبول و بهتر از پیش بینی میانگین دست بیابد تا در ادامه وقتی مدل های پیچیده با نیاز پردازشی بیشتر معرفی و تست می شوند، بتوان نیاز به سطح بالاتر از پردازش برای دقت بهتر را توجیه نمود. به شبکه ای که در این قسمت طراحی خواهید کرد یک NARMAX Network گفته می شود.

برای یکپارچگی در کل این تمرین از MSE Loss به عنوان معیار بهینه سازی و با بهینه ساز Adam Optimizer با Weight decay استفاده کنید. در طول آموزش نیز مدلی را ذخیره کنید که کمترین Loss را روی Validation دارد و در قسمت Testing باید از این مدل برای ارزیابی استفاده کنید. نکته قابل توجه این است که مدل های شما همگی باید ۴ ویژگی اصلی در پنجره خروجی یعنی قیمت های Open، Close، High و Low را پیش بینی کند. در پنجره ورودی از هر تعداد ویژگی که به نظر شما درست است استفاده کنید.

۲-۲-۱. یک شبکه MLP با دو لایه مخفی با Batch Normalization و Dropout و با تنظیم هایپرپارامترها، درست کنید و روی داده ها آموزش دهید. معیار های MAPE، RMSE، MAE و R2 را پیاده سازی کنید و با استفاده از آن تاریخچه های Metric ها و Loss را برای مدل در طول آموزش روی مجموعه آموزش و Validation رسم کنید. نمودار ها را تحلیل کرده و درباره پایداری آموزش بحث کنید (برای گرفتن نمره کامل، Validation Loss شما باید کمتر از 0.70 باشد).

۲-۲-۲. برای بازه 2024 تا 2025 که بخشی از داده تست بوده و آن را جدا کرده بوده اید، پیش بینی های مدل در مقابل قیمت واقعی را رسم و تحلیل کنید. همچنین ساختار شبکه و تعداد پارامتر های آن را گزارش کنید.

۲-۳. مدل‌های Recursive Neural Network (۵۰ نمره)

در این بخش باید معماری‌های مبتنی بر Recursion را برای پیش‌بینی سری زمانی پیاده‌سازی و ارزیابی کنید. این ساختارها اجازه خواهند داد تا پروسه‌های رندوم داخل پیش‌بینی قیمت دقیق تر از یک شبکه کاملاً متصل مدل شده و همچنین اثرات نویز و اغتشاش بهتری فیلتر و حذف شوند. ساختار کلی این بخش بر اساس معماری‌های مقایسه‌شده در مقاله است و از ۲ ساختار اصلی آن استفاده می‌شود. به شبکه‌هایی که در این قسمت طراحی خواهید کرد، Non-Linear State-Space Based Networks گفته می‌شود.

۲-۳-۱. معماری LSTM

مدل اولی که انتخاب می‌کنید، مدل CNN + LSTM خواهد بود. ساختار کلی این مدل به صورت تعدادی لایه کانولوشنی ۱ بعدی و در ادامه آن، لایه‌های LSTM برای یادگیری ساختار زمانی هستند و در انتها تعدادی لایه کاملاً متصل برای پیش‌بینی از روی مقادیر حالت LSTM است.

۲-۳-۱-۱. مدل را همانند baseline و با هایپرپارامترهای مشابه مقاله، طراحی کنید و با توجه به اندازه پنجره ورودی و خروجی، توضیح دهید لایه‌های کانولوشنی دقیقاً چه بخشی از ورودی را رؤیت می‌کنند و روی آن چه پردازشی انجام می‌دهند و در مقابل، لایه‌های LSTM چگونه به feature map های استخراج شده دست پیدا می‌کنند. همچنین توضیح دهید آیا بهتر است که hidden feature های لایه آخر LSTM به شبکه کاملاً متصل انتهایی داده شود و یا آخرین خروجی حاصل از LSTM؟

۲-۳-۱-۲. آموزش مدل را انجام بدهید و معیارهایی که در بخش قبل برای baseline مطرح شد را به همراه loss در طول آموزش برای Training و Validation رسم کنید. تعداد پارامترها و شکل ساختاری که طراحی کردید را با کمک torchsummary و یا خروجی keras از شکل شبکه، گزارش کنید. آیا شبکه این قسمت بهتر از baseline عمل کرده است؟ (برای گرفتن نمره کامل از این بخش Validation loss شما باید کمتر از 0.02 شود)

۲-۳-۲. معماری GRU

مدل CNN + GRU در این قسمت پیاده‌سازی می‌شود. ساختار این شبکه همانند LSTM است، با این تفاوت که به جای لایه‌های LSTM، از GRU استفاده می‌شود. در نظر داشته باشید برای شبکه با GRU که بتواند تعداد پارامتر مشابه شبکه LSTM داشته باشد، به تعداد لایه‌ها و تعداد سلول‌های بیشتر نیاز است ولی CNN ابتدایی باید اندازه و عمق یکسانی با بخش قبل داشته باشد.

۲-۳-۲-۱. در این قسمت مانند بخش LSTM عمل کرده و ساختار شبکه و هایپرپارامترها را تعیین بکنید. همچنین تعیین کنید آیا استفاده از hidden feature ها در لایه آخر مناسب تر است و یا آخرین خروجی GRU؟

۲-۳-۲-۲. برای مدل CNN + GRU آموزش را انجام دهید و metric ها و loss را برای داده validation و داده آموزش رسم کنید. همچنین تعداد پارامترهای شبکه و هایپرپارامترهای استفاده شده را گزارش کنید. (برای گرفتن نمره کامل از این بخش Validation loss شما باید کمتر از 0.02 شود)

۲-۴. مقایسه روی داده های تست و مقایسه های عملکرد (۲۰ نمره)

۲-۴-۱. برای مقایسه عملکرد کلیه مدل ها، روی داده های تست که قبلا از داده های دیگر جدا کرده بودید و دیده نشده اند، معیار های R^2 ، MAPE، MAE و RMSE را بدست آورید. همه این نتایج را در یک جدول بیاورید و گزارش دهید. جدول را حتما در کد خود نیز خروجی دهید. در این جدول تعداد پارامتر های هر مدل را نیز بیاورید تا تا اثر تعداد پارمتر و ساختار در کنار هم دیده شوند.

۲-۴-۲. برای تمامی مدل ها، گزارشی از زمان های آموزش بیاورید. همچنین برای هر مدل به صورت تقریبی تعداد Flops (تعداد Operation های ضرب و جمع بر اساس تعداد پارامتر های شبکه) را به صورت تقریبی بیاورید. بر روی Colab و یا Kaggle آموزش انجام می دهید، اسم GPU استفاده شده را ذکر کنید. همچنین اگر به صورت Local در Run میگیرید، اسم GPU و CPU سیستم خود را ذکر کنید. (در نظر داشته باشید که تمامی مدل ها بر روی MPS برای این سوال آموزش داده شده اند و طولانی ترین زمان آموزش برای 50 epoch برابر با ۲ دقیقه و با کتابخانه Pytorch است).

۲-۴-۳. نمودار پیش بینی قیمت و قیمت واقعی در حالت Open و یا Close را برای کل بازه ۲۰۲۴ تا ۲۰۲۵ را رسم کنید. تحلیل کنید استفاده از کدام مدل برای پیش بینی قیمت از ابتدای ۲۰۲۴ تا انتهای ۲۰۲۵ بهتر است. تمام خروجی های این مدل ها را در یک نمودار برای این بازه بیاورید و تحلیل خود را ارائه بدهید. چرا نمی توان از این مدل ها برای پیش بینی های خیلی بلند مدت (برای مثال ۱ ساله تا ۲۰۲۶) استفاده کرد؟ با استفاده از مدل و با توجه به اینکه می توان پیش بینی های خود مدل ها را به عنوان ورودی بازه بعد در نظر گرفت، پیش بینی قیمت را برای بازه ۲۰۲۶ تا ۲۰۲۷ برای تمامی مدل ها انجام داده جواب خود را اثبات کنید.

۲-۵. بخش امتیازی (۵ نمره)

شبکه های کانولوشنی همانند فیلترها عمل می کنند. در بسیاری از کاربردها می توان این فیلترها را به گونه ای طراحی کرد که با استفاده از اطلاعات گذشته، ویژگی های مربوط به آینده سیگنال را استخراج یا پیش بینی کنند. این بخش امکان این موضوع را می سنجد.

۲-۵-۱. یک شبکه CNN-Only که تنها از لایه های کانولوشنی تشکیل شده است، طراحی کنید و توضیح دهید چگونه می توان با استفاده از strid و سایز kernel از ابعاد ورودی به ابعاد خروجی رسید. تعداد لایه ها و هایپر پارامترها را گزارش کنید. آیا شبکه حاصل حتما از baseline نیز سبک تر است؟

۲-۵-۲. شبکه را با تنظیمات مشابه، روی داده های آموزش، آموزش داده، نمودار های Loss و متریک ها در طول آموزش همچنین پیش بینی قیمت برای بازه ۲۰۲۴ تا ۲۰۲۵ را رسم کرده و تحلیل کنید آیا استفاده از فقط لایه های کانولوشنی می تواند مناسب باشد یا خیر.