به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس داده کاوی پیشرفته

تمرین اول

نام و نام خانوادگی	عرفان شهابی
شماره دانشجویی	۸۱۰۱۰۳۱۶۶
تاریخ ارسال گزارش	۱۴۰۴.۰۳.۱۹

فهرست

نمودارها

بخش نظری

برای این بخش من این مسئله رو به چند مرحله تقسیم کردم:

مرحله ۱: صورت مسئله

مدل ما باید بر اساس داده های تاریخی جوامع مختلف در شبکه ای مانند Reddit، پیش بینی کند که کدام یک از نتایج زیر از داده های برگرفته شده از جامعه حاصل می شود.

- Stable
- Growing
- Shrinking
- Spliting
- Dissolving

مرحله ۲: ورودی مدل

همانطور که در صورت سوال گفته شده ما سه نوع داده داریم:

1. ویژگی های درون جامعه:
   مانند تعداد پست، تعداد کاربران فعال، میانگین نظرات به ازای هر پست و یا میزان رشد اعضا و انحراف معیار مشارکت.
2. تعاملات بین جوامع:
   مانند درصد مشترک کاربران بین جوامع، تعداد تعاملات متقابل و یا میزان شباهت موضوعی و میانگین تعداد نقل قول ها یا ارجاعات متقابل.
3. برچسب تحولی آینده:
   که یکی از پنج حالت بالا است و از داده های سه ماه آینده استخراج می شود (لیبل برای Supervised Learning). این برچسب فقط در فاز آموزش و ارزیابی استفاده می شود و نه در inference.

مرحله ۳: استخراج Featureها:

در این مرحله من اول فکر کردم که بهتر است هر کاربر یک نود از این گراف باشد و جوامع را با استفاده از Clustering مشخص کنیم. اما از آنجا که در سوال گفته شده هر جامعه یک گره است با این فرض ادامه می دهم که نود گرافی که داریم یک جامعه است که برای هر جامعه یعنی هر نود از گراف:

ما داده هایی داریم که طی ۶ ماه گذشته، به صورت هفتگی ثبت شده اند (یعنی ۲۶ timestep برای هر ویژگی). فرض کنیم برای هر جامعه در هر هفته یک بردار ویژگی داریم شامل مواردی مانند:

- تعداد پست ها
- تعداد کاربران فعال
- نرخ رشد کاربران
- میانگین نظرات به ازای هر پست

ما میخواهیم از این سری زمانی یک بردار ویژگی با طول ثابت استخراج کنیم که خلاصه ای از رفتار آن جامعه باشد. برای این کار روی هر ویژگی آمار های زیر را محاسبه میکنیم:

- میانگین که نشان دهنده میانگین فعالیت جامعه طی ۶ ماه است.
- انحراف معیار جهت سنجش نوسان و پایداری رفتار جامعه.
- نرخ تغییر خطی که بر اساس رگرسیون خطی روی سری زمانی محاسبه می شود و نشان می دهد که آیا ویژگی مورد نظر در حال رشد است یا کاهش.

در مرحله بعد گراف تعامل بین جوامع برای هر ماه استخراج می شود که در آن نود ها جوامع هستند و وزن یال بین دو جامعه به نظرم میتواند ترکیبی وزن دار از درصد کاربران مشترک، شباهت موضوعی و تعاملات بین گروهی باشد

برای تبدیل گراف تعاملات به embedding نود ها می توانیم از مدل های pretrian مانند GraphSage استفاده کنیم.

این مدل قابلیت استفاده برای جوامع جدید را دارد و نسبت به GCN مقیاس پذیر تر است. سپس میتوانیم از گراف ساخته شده یک امبدینگ برای هر جامعه استخراج کنیم. همچنین میتوانیم ابتدا مدل را روی یک تسک پیش فرض مانند link prediction یا node classification، pretrain کنیم و سپس از آن امبدینگ ها استفاده کنیم.

مرحله ۴: طراحی مدل

مدل انتخابی: GNN based multi classifier

مدل ما یک شبکه عصبی است گرافی است که روی گراف تعاملات بین جوامع کار میکند و ویژگی‌های هر نود و ساختار گراف را در نظر میگیرد.

ورودی این مدل بردار ویژگی عددی و embedding گرافی از مدل GraphSage بالا است.

مدل می تواند شامل دو لایه GNN مثلا GraphSage باشد. همچنین به نظرم باید یک لایه برای concatenation فیچر های عددی و گرافی قرار دهیم. همچنین یک تابع softmax هم باید در انتهای معماری قرار دهیم برای تولید احتمال تعلق به ۵ کلاس.

مرحله ۵: آموزش مدل

در این مرحله مدل روی جوامعی آموزش می بیند که داده های گذشته و آینده آن ها موجود است و فقط از اطلاعات گذشته جوامع استفاده می شود تا لیبل آینده شان در مرحله آموزش پیش بینی شود و در حقیقت supervised classification انجام میشود. طبق گفته سوال می توانیم از جوامع جدید و دیده نشده برای تست مدل استفاده کنیم. همچنین به نظرم میتوانیم از CrossEntropy Loss به عنوان تابع هزینه برای این دسته بندی چند کلاسه استفاده کنیم.

مرحله ۶: ارزیابی مدل

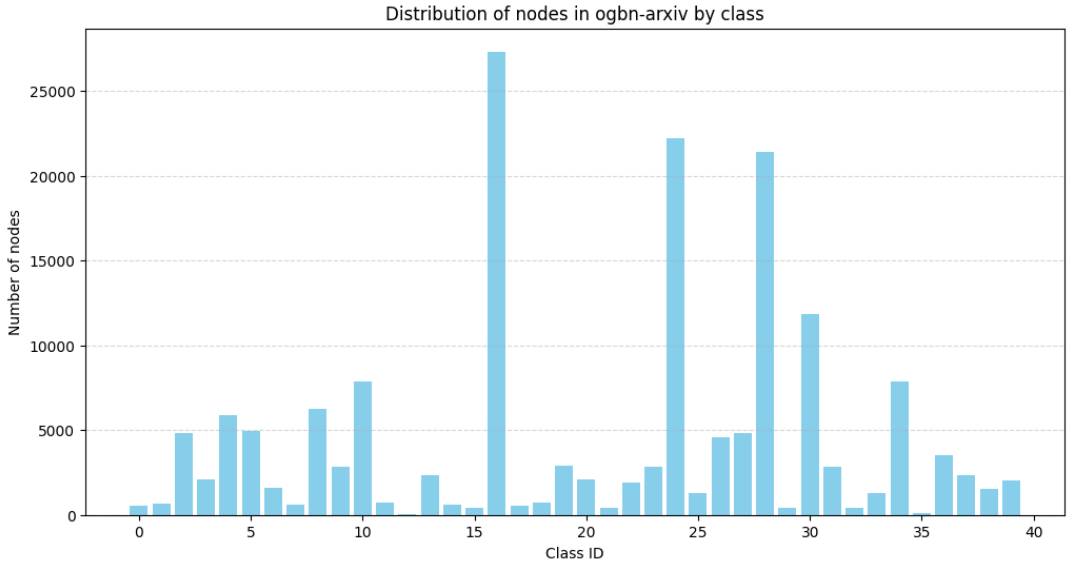
برای ارزیابی مدل میتوانیم از Accuracy، Macro-F1، ماتریس آشفتگی و Recall, Precision استفاده کنیم استفاده کنیم و عملکرد مدل روی هر کلاس را بررسی کنیم. مثلا می توانیم بررسی کنیم که آیا مدل می تواند Dissolving را خوب تشخیص دهد یا نه؟ دلیل استفاده از Macro-F1 این است که داده ها ممکن است نامتوازن باشند. همچنین ماتریس آشفتگی می تواند برای تحلیل نوع خطا مفید باشد. precision, Recall هم برای به دست آوردن درصد کلاس های مثبت واقعی نسبت به همه کلاس ها و درصد کلاس های مثبتی که به درستی لیبل زده شده را مشخص میکنند.

بخش عملی

آشنایی با مجموعه دادگان

بخش ۱

در این بخش با توجه به سوال داده های "ogbn-arxiv" را با استفاده از کتابخانه OGB بارگزاری کردم و نمونه ای از دیتا ست را نمایش دادم. همچنین توزیع کلاس ها در این دیتاست را با استفاده از یک هیستوگرام نمایش دادم که این توزیع به شرح زیر است:



نمودار 1

همانطور که از نمودار مشخص است، توزیع کلاس ها در این دیتاست کاملا نا متوازن است و برخی کلاس ها مانند کلاس های ۱۶، ۲۴ و ۲۸ به تعداد بسیار بیشتری از بقیه توزیع ها تکرار شده‌اند.

در این سناریو با توجه به اینکه داده های ما به صورت گراف هستند، oversampling و undersampling معمولا دشوار و با ریسک بالا هستند چرا که نود های تکراری ممکن است ساختار گراف را خراب کنند. به همین دلیل در این سوال برای متوازن سازی داده ها هنگام آموزش با استفاده از یک تابع وزن معکوس برای هر کلاس را بر اساس توزیع در داده های train محاسبه کردم و در حقیقت Re-weighting انجام دادم که در این راه حل تعداد نود های هر کلاس در داده های آموزش را می شماریم و وزن معکوس هر کلاس را محاسبه میکنیم و داده ها را برای اعمال loss وزن دهی دوباره میکنیم. این کار باعث می شود که کلاس های کم نمونه دارای وزن بیشتری بشود و مدل در حالت آموزش به آن ها بیشتر توجه کند.

بخش ۲

اگر دیتا ست شامل فایل های خامی باشد که در آن ها هر مقاله دارای عنوان، ID و کلاس باشد و روابط ارجاعی بین مقالات را هم داشته باشیم به نظرم برای ساخت گراف مشابه با دیتاست "ogbn-arxive" باید به صورت زیر عمل کنیم:

1. ساخت نود ها:

در این سناریو هر مقاله تبدیل به یک نود در گراف می شود و برای هر مقاله یک node_id یکتا تعریف میکنیم.

2. استخراج ویژگی ها:

برای استخراج ویژگی ها از عنوان مقاله ابتدا باید روی عنوان پیش پردازش انجام شود. این پیش پردازش می توان شامل کوچک کردن حروف، حذف علائم و حذف stop word ها باشد. در مرحله بعد باید این متون را به بردار ویژگی تبدیل کنیم. برای انجام این کار میتوانیم از ساده ترین مدل ها مانند TF-IDF و یا Bag of Words استفاده کنیم و یا اینکه از روش های پیشرفته در مانند Encoderهایی مانند Bert استفاده کنیم. بعد از این مرحله، هر بردار تبدیل به feature هر نود می شود.

3. ساخت یال ها:

در مرحله بعد از فایل ارجاعات استخراج می کنیم که چه مقاله ای به چه مقاله ارجاع داده است و برای هر ارجاع یک یال جهت دار بین نود ها تعریف میکنیم.

4. ساخت گراف:

سپس با استفاده از کتابخانه هایی مانند DGL و یا PyTorch Geometric گراف را با نود ها و یال ها می سازیم همچنین میتوانیم داده ها را نیز تقسیم بندی کنیم. مثلا ۸۰٪ برای اموزش، ۱۰٪ برای اعتبار سنجی و ۱۰٪ هم برای تست.

بخش ۳

- دیتاست های آماده بسیار سریع هستند در صورتی که آماده سازی دستی از فایل خام نیازمند زمان زیاد برای آماده سازی و تمیز کردن داده هاست.
- دیتاست های آماده انتزاع شده و آسان هستند در صورتی که آماده سازی دستی نیازمند درک عمیق از ساختار داده و مراحل پردازش است.
- دیتا ست آماده معمولا استاندارد است اما آماده سازی دستب ممکن است موجب ایجاد خطا و یا ابهام در پردازش داده ها شود.
- دیتاست های آماده محدود به ساختار از پیش تعیین شده هستند در صورتی که در آماده سازی دستی میتوان طراحی گراف، ویژگی ها و لیبل ها را به دلخواه انجام دهیم.

ایجاد ساختار مدل

در این قسمت من ابتدا کلاس GCNEncoder را تعریف کردم که یک مدل پایه GCN است که برای استخراج امبدینگ ها از نود ها در یک گراف استفاده می شود. در این قطعه کد بعد از ایمپورت کردن پکیج ها و ماژول های مورد نیاز، معماری را تعریف کردم که در این مدل شامل ۲ لایه GCN است. لایه اول ویژگی های ورودی گراف نود ها با ابعاد in_dim را به hidden State، hidden_dim نگاشت میکند. لایه دوم نیز امبدینگ های hidden را به فضای خروجی out_dim تبدیل می کند. ابتدا من برای این مدل از لایه Dropout استفاده نکرده بودم که نتایج مورد نیاز حاصل نشد به همین دلیل من از یک لایه Dropout با احتمال 0.5 برای جلوگیری از overfitting در حین آموزش استفاده کردم.

سپس مسیر عبور داده forward رو تعریف کردم که ابتدا گره ها با لایه اول GCN پردازش می شوند و سپس تابع ReLU روی خروجی اعمال می شود. در مرحله بعد Dropout روی خروجی لایه اول اعمال می شود تا به مدل کمک کند از یادگیری بیش از حد جلوگیری کند و سپس خروجی Dropout وارد لایه دوم GCN میشود. در نهایت خروجی x یک امبدینگ برای هر نود در گراف است که میتواند مستقیما وارد classifier شود.

در مرحله بعد کلاس MultiTaskModel را تعریف کردم که یک معماری یادگیری مالتی تسک برای گراف ها است و به صورت همزمان دو وظیفه زیر را انجام می دهد.

- Node Classification
- Link Predition

این مدل با استفاده از یک Encoder گرافی مشترک مبتنی بر GCNEncoder که در بالا تعریف شد ابتدا امبدینگ نود ها را استخراج می کند و سپس از این امبدینگ ها برای انجام دو وظیفه بالا استفاده میکند.

اجزای این معماری به این صورت است که ابتدا ماژول GCNEncoder که در بالا توضیح داده شد برای استخراج بردار امبدینگ ها از هر نود گراف قرار دارد و سپس دو headهای مربوط به Node Classification و Link Prediction قرار دارد. وظیفه اول برای این مدل طبقه بندی نود ها است، برای هر نود، خروجی مدل یک بردار از طول num_classes است که احتمال تعلق آن نود به هر کلاس را نشان می دهد. وظیفه دوم نیز پیش بینی لینک هاست. در این تسک اگر جفت نود ها وجود داشته باشد ابتدا امبدینگ مبدا و مقصد هر یال انتخاب میشود و سپس این دو بردار با هم concate میشود و در نهایت با استفاده از یک لایه linear یعنی sigmoid پیش بینی می شود که آیا لینک واقعا بین آن ها وجود دارد یا نه. در نهایت خروجی مدل Loss برای محاسبه Classification نودها و همجنین loss برای پیش بینی لینک ها به همراه امبدینگ نهایی گره ها است.

آموزش مدل

در این بخش طبق گفته سوال، ابتدا طبق گفته تابعی برای برای وزنی دهی مجدد به کلاس ها نوشتم که این تابع وزن معکوس کلاس ها را برای loss طبقه بندی نود ها محاسبه می کند. که با این روش مدل به loss کلاس هایی که تعداد کمی فراونی دارد بیشتر از حالت قبل توجه میکند. در این تابع ابتدا فقط برچسب های نود های آموزشی گرفته می شود. سپس تعداد نمونه ها در هر کلاس شمرده می شود و وزن معکوس نیز محاسبه میشود و در مرحله آخر وزن ها نرمال سازی می شوند.

از آنجا که از torch ارور میگرفتم که بعضی داده ها روی CPU قرار دارند، همه داده ها را به GPU ارسال کردم. در مرحله بعد lossها را تعریف کردم که طبق گفته سوال برای نود ها از Cross Entropy Loss و برای لینک ها از BCE With Loss استفاده کردم. در نهایت تابع محاسبه loss مالتی تسک را نوشتم که در آن مجموع وزن دار دو loss برای آموزش مدل محاسبه می شود.

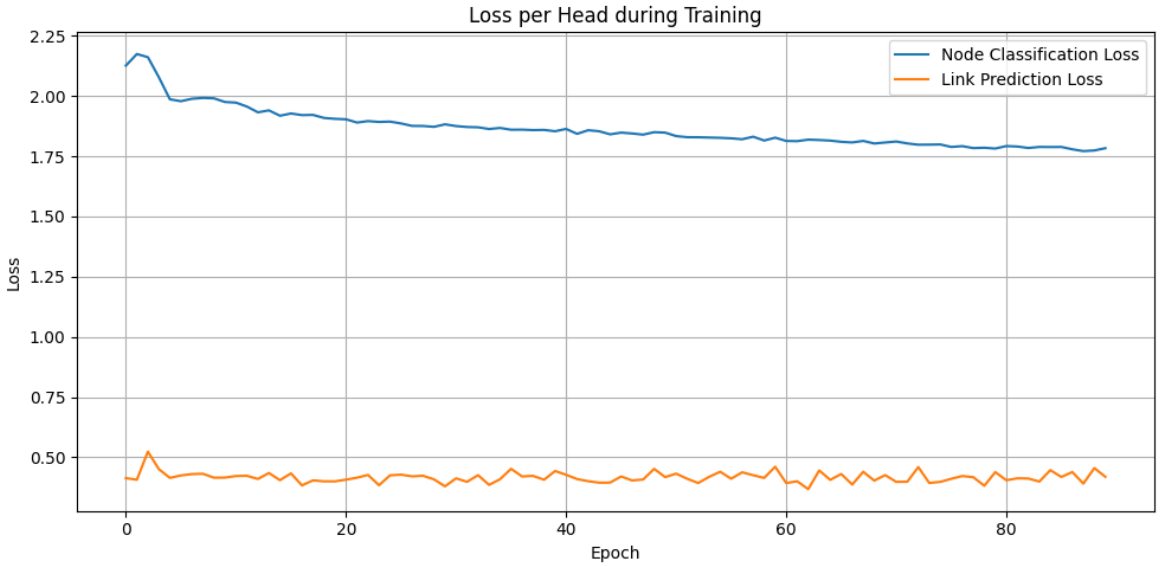
در مرحله بعدی تابع سمپل گیری از لینک ها نوشته شد که وظیفه این تابع ساختن مجموعه‌ای از جفت نودها برای تسک پیشبینی وجود و یا عدم وجود یال بین آن هاست که شامل تعداد برابر نمونه های مثبت و منفی می باشد.

از آنجا که طبق معماری گفته شده در سوال ورودی انکودر شامل بردارد TF-IDFو سال انتشار مقاله و یال ها بود،تابع پیش پردازش فیچر ها را نوشتم که وظیف آن تبدیل سال انتشار به یک سلول و اضافه کردن آن به بردار tf-idf است. برای قسمت یال ها هم من یال ها رو من از edge indexبرای message passing در انکودر استفاده کردم و در مرحله بعد آرگومان های مدل رو تعریف کردم.

در مرحله پایانی لیست هایی را برای ذخیره معیار های ارزیابی در هر ایپاک تعریف کردم و مدل را به تعداد ۹۰ ایپاک روی داده ها آموزش دادم. اپتیمایزر استفاده شده Adam است که با lr = 0.1 مقدار دهی شده است.

ارزیابی مدل

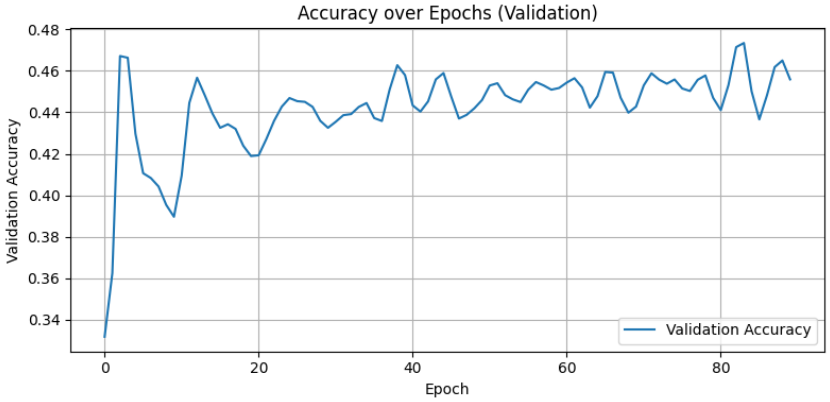
نمودار تغییرات Loss هر head:



نمودار 2

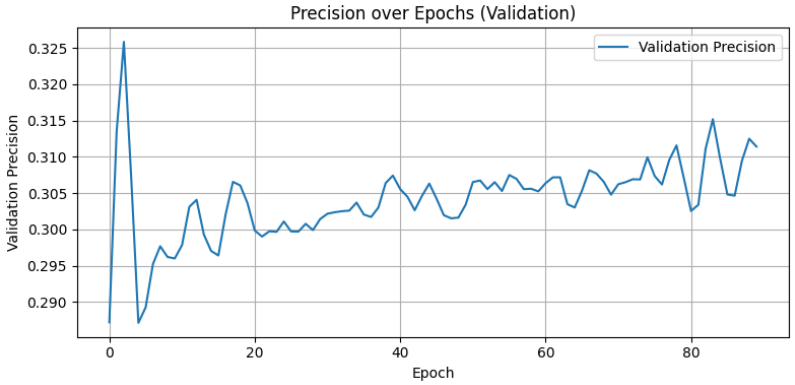
همانطور که از نمودار ها مشخص است، مقدار loss برای node Classification به وضوح کاهشی بوده است. اگر چه که این کاهش در ایپاک های انتهایی شیبی نسبتا کمتر داشته است. به نظرم این به این معناست که مدل در حال یادگیری است اما با توجه به نمودار در ایپاک های انتهایی به مرحله اشباع در یادگیری باشد. برای تسک Link Prediction همانطور که از نمودار مشخص است، مقدار loss از ابتدا بسیار پایین تر از تسک classification است و با اینکه نوسانات کوچکی دارد ولی در مجموع خیلی پایدار و نزدیک به مقدار ثابت باقی می ماند. به نظرم دلیل این موضوع این است که احتمالا چون داده های مث بت و منفی واضح هستند مدل از همان ابتدا در کار پیش بینی یال ها نسبتا خوب عملکرده است.

نمودار تغییرات معیار های ارزیابی (Node Classification Validation):



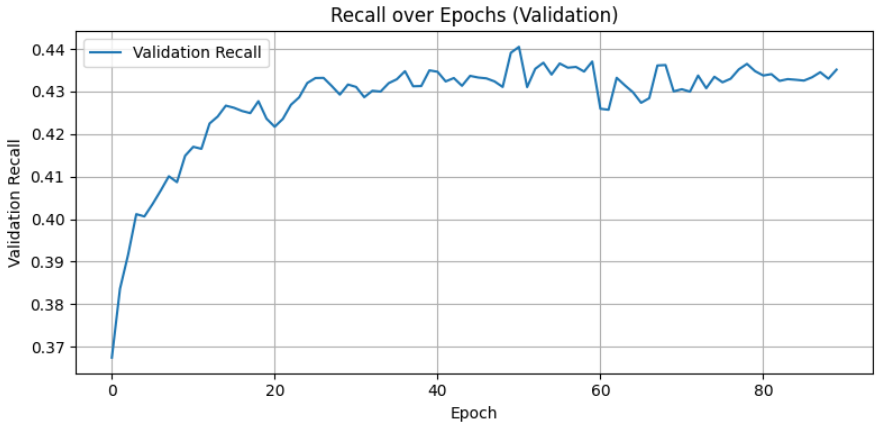
نمودار 3

روند کلی تغییرات Accuracy به این گونه بوده است که ابتدای آموزش جهشی سریع داشته است که نشان دهنده آن است که مدل به سرعت از وضعیت پیش بینی تصادفی به یک سطح قابل قبولی میشود و نشان دهنده یادگیری موفق اولیه است. از ایپاک ۱۰ به بعد مدل دچار نوعی نوسان و تثبیت شده است اما overfitting و underfitting دیده نمیشود و به نظرم مدل یادگیری جدیدی روی داده های validation ندارد.



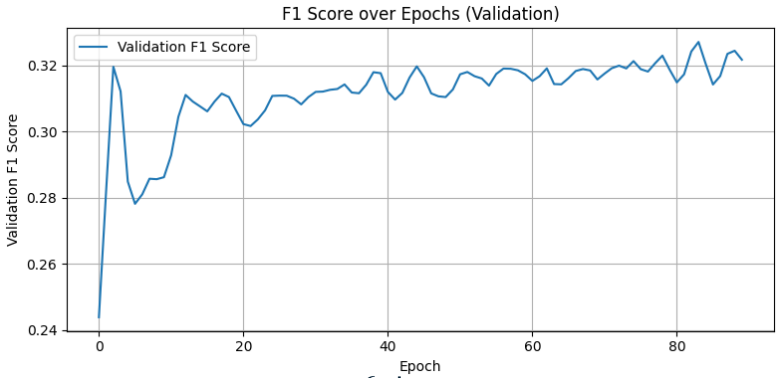
نمودار 4

برای تغییرات Precision ابتدا این معیار پایین است و ناگهان وارد جهش می شود و سپس افت میکند. میتوان گفت این رفتار طبیعی است چون مدل هنوز ناپایدار است و تازه یادگیری را شروع کرده است. بین ایپاک های ۱۰ تا ۶۰ precision در یک بازه بسیار کوچکی نوسان پیدا میکند و میتوان گفت در این بازه مدل در حال یادگیری جزئیات بیشتری است اما رشد آهسته دارد و در ایپاک های پایانی هم این روند رشد آهسته دیده می شود. تفاوت این معیار با Acccuracy نشان میدهد که احتمالا مدل برای کلاس های نادر و کم precision کمی دارد و پیشبینی های درست ان بیشتر از روی کلاس های پر تکرار نتیجه شده اند.



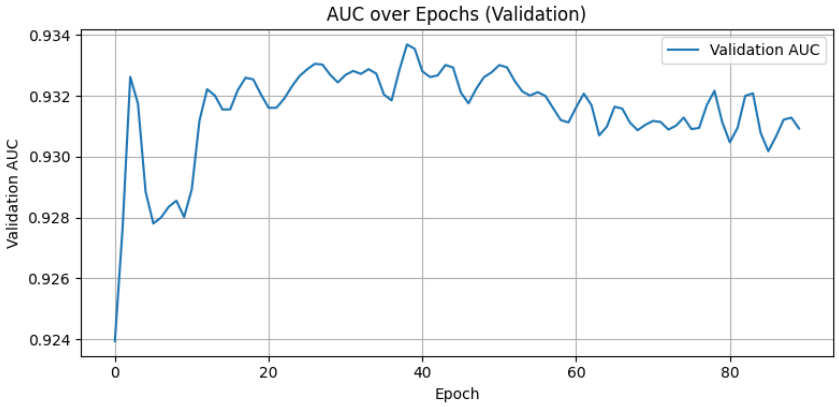
نمودار 5

در نمودار Recall برای هر ایپاک همانطور که مشخص است این معیار در شروع یادگیری بسیار پایین است و بین ایپاک های ۱۰ تا ۱۵ خیلی سریع رشد میکند. از ایپاک ۲۰ به بعد رشد بسیار اهسته است و نوسان کمی دیده می شود. در ایپاک ۶۰ یک سقوط موقتی دیده می شود که به سرعت بازیابی می شود. این تغییرات به نظرم نشان می دهد که مدل خیلی زود توانایی پیشبینی کلاس های مثبت را به دست می آورد اما بعد از ایپاک ۳۰ یادگیری تقریبا به تثبیت می رسد و مدل دیگر بهبود چشم گیری ندارد. اموزش بر روی داده های validation پایدار است و overfitting در مدل دیده نمی شود. ترکیب precision نسبتا پایین با Recall نسبتا بالا احتمالا نشان دهنده false positive های زیاد است.



نمودار 6

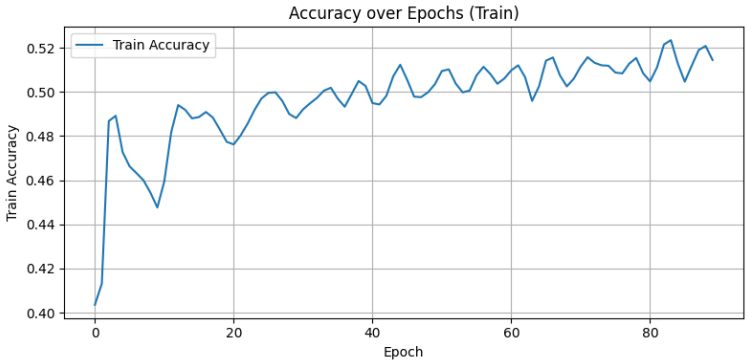
از انجا که معیار F1 ترکیبی از دو معیار precision و Recall است، یکی از مهم ترین متریک ها برای ارزیابی مدل های چندکلاسه می باشد. در ایپاک اول این معیار بسیار پایین است اما در ایپاک های اولیه رشد بسیار سریعی دارد که نشان دهنده این است ه مدل یادگیری پایه ای را به خوبی انجام داده است. سپس افت و نوسان دیده می شود که به نظرم طبیعی است چرا که مدل در مراحل اولیه آموزش است و در حال تنظیم وزن ها می باشد. از ایپاک ۲۰ به بعد مدل از نظر generalization روی داده های validation تقریبا به سقف توانایی خودش رسیده است.



نمودار 7

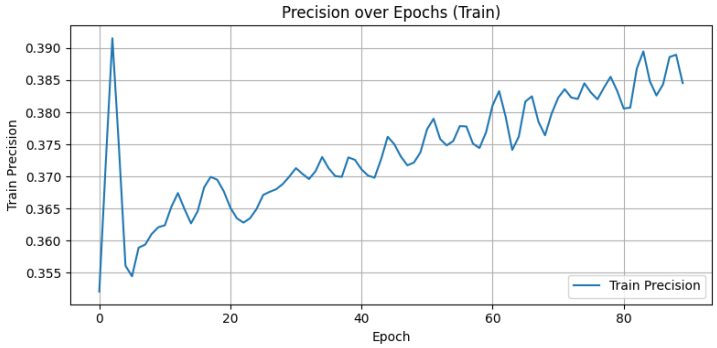
در این نمودار همانطور که مشخص است مقدار AUC بسیار بالا و پایدار بوده است و نشان دهنده قدرت مدل در تمایز بین کلاس ها حتی در شرایط عدم توازن داده هاست. طبق نمودار آموزش پایدار بوده است و هیچ افت ناگهانی و یا overfitting شدیدی مشاهده نمیشود و مدل در یک سطح یادگیری خوب تثبیت شده است. متریک های قبلی همه متوسط بوده اند اما به نرم مقدار AUC بالا نشان دهنده این است که اگر صرفا بخواهیم تصمیم گیری به صورت one vs all انجام دهیم مدل توانایی تشخیص بالایی دارد.

نمودار تغییرات معیار های ارزیابی (Node Classification Train):



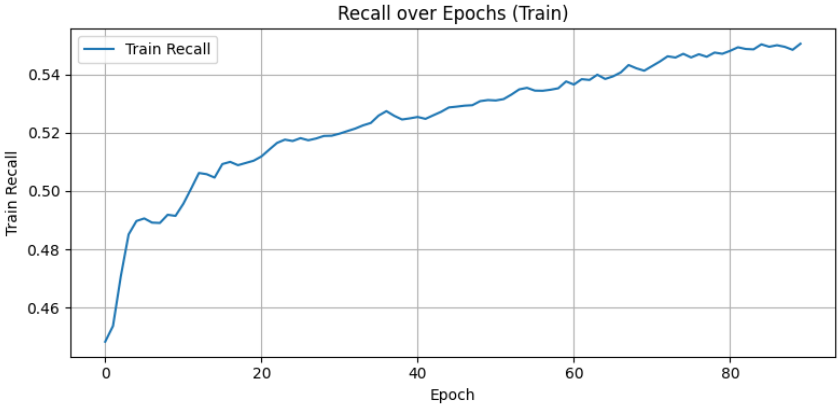
نمودار 8

رشد سریع اولیه در این نمودار به نظرم نشان دهنده این است که مدل روی داده های ترین خیلی زود الگوهای ساده را یاد میگیرد. در میانه نمودار نوسان دیده میشود اما بهبود تدریجی نشان دهنده یادگیری فعال مدل می باشد.



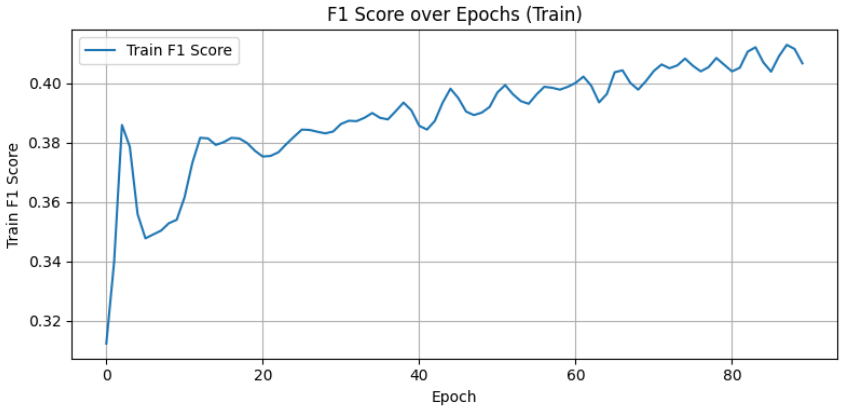
نمودار 9

نمودار Precision نشان میدهد که مدل رشد سریعی دارد که نشان میدهد مدل روی داده ها ترین خیلی زود الگوهای ساده را یاد گرفت است. اگر چه کاهش شدیدی داشته اما مدل رشد پایداری داشته و مدل overfit نکرده و یادگیری مدل واقع بینانه است. این معیار هنوز مقدار پایینی دارد که نشان دهنده این است که مقدار false positive ها نسبتا زیاد است. تفاوت قابل توجه با validation precision نشان دهنده generalization gap است.



نمودار 10

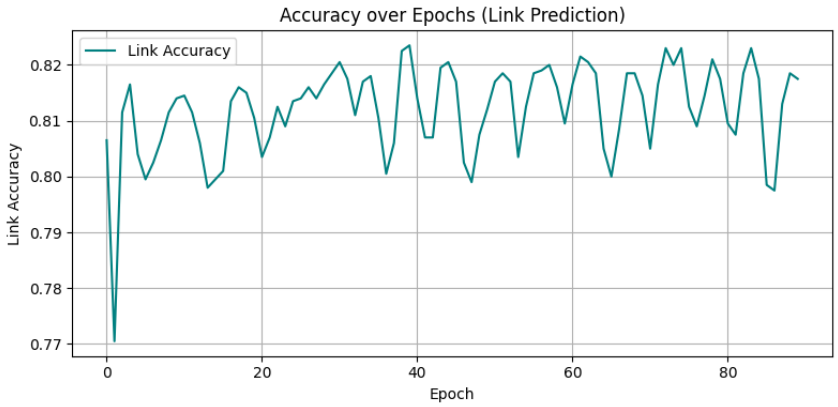
این نمودار نشان می دهد که مدل در طول آموزش توانسته به طور پیوسته توانایی خود را در شناسایی صحیح نمونه های مثبت افزایش دهد. رشد در ابتدا رشد بسیار زیادی داشته است اما به تدریج پایدار تر شده است و همچنان به رشد خود اما با شیب پایین ادامه داده است. روند صعودی و بدون نوسان شدید این نمودار نشان دهنده پایداری یادگیری و عدم وجود overgitting محسوس در این معیار است.



نمودار 11

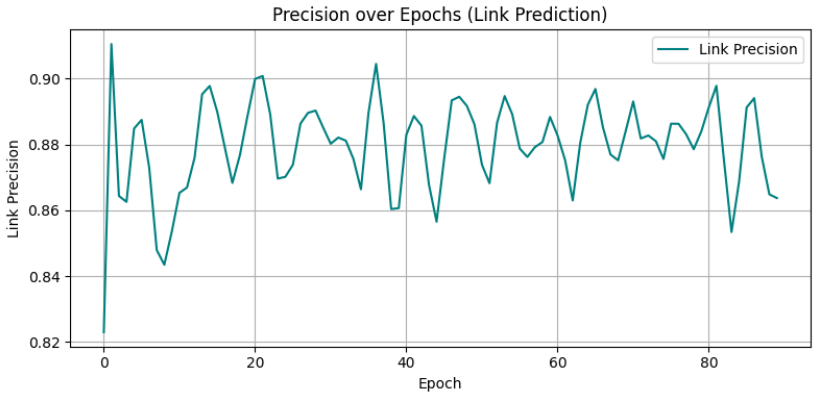
نمودار F1 روی داده های آموزش نشان میدهد که مدل به مرور زمان توانسته است تعادل بهتری بین Precision و Recall برقرار کند.مدل به مرور پایدار شده است و همچنین نبود نوسانات شدید بیانگر ثبات در آموزش و عدم Overfitting در هنگام ترین است.

نمودار تغییرات معیار های ارزیابی (Link Prediction):



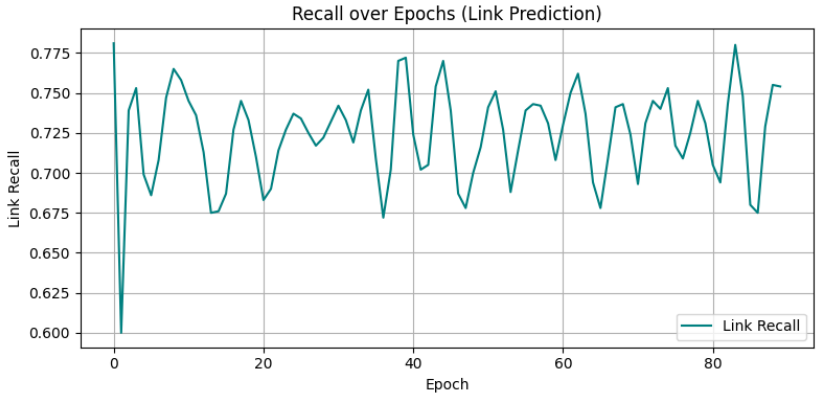
نمودار 12

این نمودار نشان می دهد که مدل از همان ابتدای آموزش عملکرد نسبتا خوبی در پیش بینی وجود و یا عدم وجود یال بین نود ها داشته است. دقت مدل در پیش بینی لینک ها حدود ۰.۸ تا ۰.۸۲ بوده است و همچنین در برخی ایپاک ها افت های کوتاه مدت دیده می شود اما به طور کلی روند مدل پایدار و قابل اعتماد است. این موضوع می تواند به سادگی نسبی وظیفه Link Prediction در مقایسه با Node Classification یا به تعادل بهتر داده های مثبت و منفی در نمونه سازی یال ها بر می گردد.



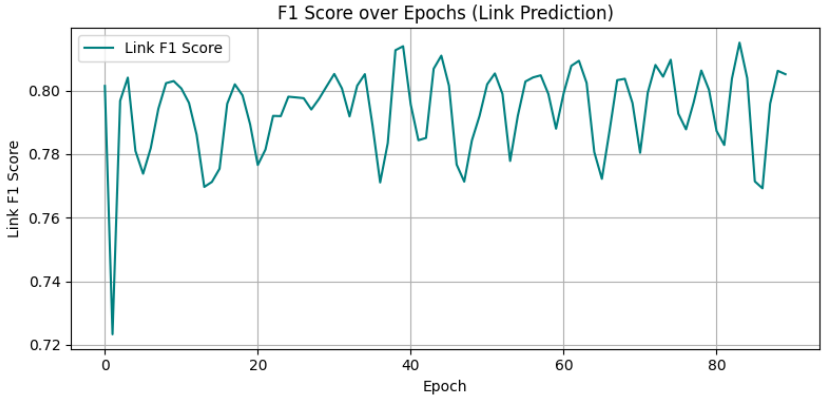
نمودار 13

نمودار Precision نشان می دهد که مدل در طول آموزش توانسته دقت بالایی در پیش بینی لینک های مثبت واقعی حفظ کند. مقدار بالای این معیار نشان دهنده این است که مدل در تشخیص لینک های واقعی عملکرد خوبی دارد و میزان false positive ها بسیار پایین است.



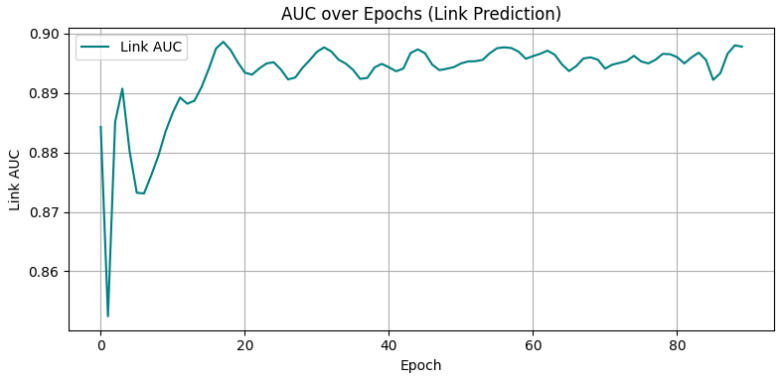
نمودار 14

این نمودار نشان دهنده این است مدل در طول آموزش توانسته است بخش قابل توجهی از لینک های واقعی را شناسایی کند اما با نوسانات نسبتا زیاد. بر خلاف نمودار Precision این نمودار بی ثبات تر است. این نوسان نشان می دهد که مدل توانسته در بعضی ایپاک ها لینک های واقعی بیشتری را شناسایی کند ولی در برخی دیگه نمونه های false negative را از دست می دهد.



نمودار 15

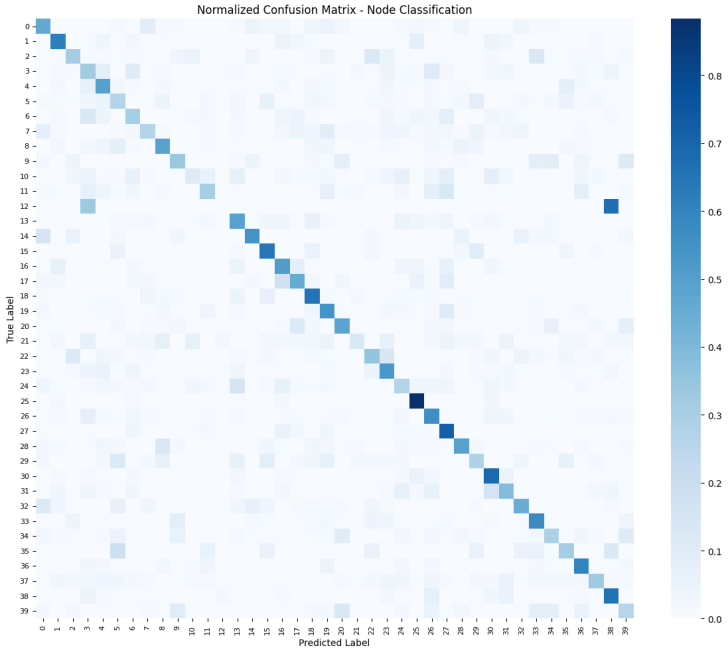
نمودار F1 از آنجا که میانگینی وزن دار بین precision و recall است میتواند معیار بسیار خوبی باشد. همانطور که از این نمودار مشخص است مدل توانسته در تسک پیش بینی لینک ها عملکرد کلی خوبی داشته باشد و توانسته است تعادلی نسبتا مناسب بین این دو معیار برقرار کند. مقدار F1 در بیشتر ایپاک ها بین 0.78 تا 0.82 نوسان دارد که بیانگر قدرت مدل در تشخیص لینک های مثبت واقعی با نرخ خطای پایین است. هر چند نوساناتی در طول آموزش دیده می شود اما عملکرد پایدار و بدون افت ناگهانی باقی مانده است. در ایپاک پایانی مقدار F1 برای این تسک بیشتر از ۰.۸ بوده است.



نمودار 16

این نمودار نشان میدهد که مدل از همان ابتدای آموزش توانسته به سطح بالایی از تمایز میان لینک های مثبت و منفی دست پیدا کند. این معیار به خدود ۰.۸۹۸ رسیده و در طول آموزش با نوسانات بسیار کم تقریبا نزدیک همین مقدار باقی میماند. این پایداری بالا و سطح عملکرد خوب نشان می دهد که مدل در تشخیص احتمال درست بودن لینک ها بسیار قوی عمل کرده است.

ماتریس در هم ریختگی:



نمودار 17

این نمودار ماتریس درهم ریختی نرمال شده مربوط به تسک Node Classification است. این ماتریس ۴۰ کلاس مختلف زا نشان میدهد که مقادیر درون ماتریس نزمال شده بر اساس True lable ها هستند بنابراین هر ردیف مجموعش تقریبا ۱ است. محور افقی برچسب های پیش بینی است و محور عمودی نیز برچسب های واقعی است. چیزی که از این ماتریس برداشت می شود این است که قطر این ماتریس نشان می دهد مدل توانسته عملکرد قابل قبولی را برای این تسک داشته باشد. هر چند که برخی برچسب ها مانند لیبل ۳۹ را به اشتباه پیش بینی کرده اما عملکرد کلی مدل قابل قبول است

نتیجه:

همانطور که در بالا گفته شد می توان عملکرد مدل را قابل قبول ارزیابی کرد. با توجه به معیار F1 این معیار برای تسک Node Classification برابر با 0.3 و برای تسک Link Prediction برابر با 0.8051 بوده است.

اظهارنامه استفاده از هوش مصنوعی

تأیید می‌کنم که از ابزارهای هوش مصنوعی مطابق با دستورالعمل‌های بارگذاری شده در سامانه Elearn درس به طور مسئولانه استفاده کرده‌ام. تمام اجزای کار خود را درک می‌کنم و آماده بحث شفاهی درباره آنها هستم.