پیشبینی ساختار دوم پروتئین

سالار نوری، علیرضا نوبخت، محمد زندیه، سجاد طهماسبی، علی حاجی صادقیان

چکیده: در این پژوهش، روشی نوین برای پیش‌بینی ساختار دوم پروتئین‌ها با استفاده از مدل‌های زبانی مبتنی بر Transformer ارائه شده است. هدف این روش، بهره‌گیری از توضیحات متنی مرتبط با پروتئین‌ها، از جمله ویژگی‌های عملکردی آن‌ها، برای پیش‌بینی دقیق ساختار دوم پروتئین‌ها در قالب Q8 می‌باشد. این روش با استفاده از تکنیک‌های پردازش زبان طبیعی (NLP)، ارتباطات پیچیده موجود در توالی‌های پروتئینی را مدل‌سازی می‌کند، که تا کنون به‌طور دقیق بررسی نشده است. مدل پیشنهادی ما بر روی مجموعه داده‌های جامعی از توصیفات پروتئینی و ساختارهای Q8 آموزش دیده است.
در صورت موفقیت‌آمیز بودن آزمایش‌ها، نتایج امیدوارکننده‌ای در پیش‌بینی دقیق ساختارهای دوم پروتئینی نشان داده خواهد شد. این رویکرد می‌تواند در تحلیل عملکرد پروتئین‌ها و توسعه داروهای جدید نقش مؤثری ایفا کند.

کلمات کلیدی: ساختار دوم پروتئین، قالب Q8، Transformer، پردازش زبان طبیعی، یادگیری عمیق، عملکرد پروتئین، پیش‌بینی ساختار پروتئین، بیوانفورماتیک، یادگیری ماشین، مدل‌سازی پروتئین.

۱ تعریف مسأله

پیش‌بینی ساختار دوم پروتئین‌ها یکی از چالش‌های اساسی در بیوانفورماتیک است که تاثیر مستقیمی بر فهم عملکردهای زیستی پروتئین‌ها دارد. ساختار دوم پروتئین‌ها شامل عناصر هلیکس‌های آلفا، شیت‌های بتا و کویل‌ها است که بر اساس توالی اسیدهای آمینه در پروتئین شکل می‌گیرند. این ساختارها از اهمیت بالایی برخوردارند چرا که نقش کلیدی در تعیین رفتار و عملکرد بیولوژیکی پروتئین‌ها دارند.

هدف اصلی این پژوهش، توسعه یک مدل زبانی مبتنی بر *Trans-former* برای پیش‌بینی ساختار دوم پروتئین‌ها با فرمت Q8 است. این مدل زبانی قادر خواهد بود که با استفاده از توضیحات متنی مربوط به یک پروتئین یا ویژگی‌های عملکردی آن (*Functionality*)، ساختار دوم آن را پیش‌بینی کند. ساختارهای Q۸ شامل هشت نوع مختلف از عناصر ساختاری پروتئین‌ها مانند هلیکس‌ها، شیت‌ها و کویل‌ها می‌باشند که هر کدام نقش مهمی در عملکرد کلی پروتئین‌ها دارند.

این مدل می‌تواند به‌عنوان ابزاری کارآمد برای پیش‌بینی ساختار دوم پروتئین‌ها بر اساس داده‌های متنی مورد استفاده قرار گیرد و می‌تواند در تحلیل داده‌های پروتئینی و توسعه داروهای جدید مفید باشد.

۲ مقدمه

پروتئین‌ها به‌عنوان یکی از مولکول‌های اساسی در سلول‌های زنده نقش کلیدی در تمامی فرآیندهای بیولوژیکی دارند. ساختار پروتئین‌ها به چهار سطح ساختاری تقسیم می‌شود: ساختار اولیه که توالی اسیدهای آمینه را مشخص می‌کند، ساختار دوم که شامل آرایش‌های محلی مانند هلیکس‌های آلفا، شیت‌های بتا و کویل‌ها است، ساختار سوم که چینش سه‌بعدی پروتئین را نشان می‌دهد و ساختار چهارم که در صورت وجود، ارتباط بین چندین زنجیره پلی‌پپتیدی را توضیح می‌دهد.

در این پژوهش، تمرکز اصلی بر روی ساختار دوم پروتئین‌ها و پیش‌بینی آن با استفاده از مدل‌های *Transformer* است. این مدل‌ها با بهره‌گیری از پردازش زبان طبیعی (NLP) قادر به درک و مدل‌سازی ارتباطات پیچیده موجود در توالی‌های پروتئینی هستند. ما از داده‌های متنی مرتبط با پروتئین‌ها به‌عنوان ورودی استفاده می‌کنیم تا ساختار دوم آن‌ها را پیش‌بینی کنیم.

۳ پیشینه پژوهش

پیش‌بینی ساختار دوم پروتئین‌ها با استفاده از مدل‌های پیشرفته هوش مصنوعی و یادگیری عمیق، در سال‌های اخیر تحولات زیادی را تجربه کرده است. اولین تلاش‌ها در این زمینه مربوط به استفاده از مدل‌های

یادگیری عمیق مانند DeepCNF برای پیش‌بینی ساختار دوم پروتئین‌ها بود [۱]. این مدل با بهره‌گیری از شبکه‌های عصبی کانولوشنی عمیق و فیلدهای شرطی عصبی، دقت بالاتری را در پیش‌بینی ساختارهای پروتئینی نسبت به روش‌های قبلی به‌دست آورد.

در ادامه، Asgari and Mofrad (2019) به بررسی انتقال یادگیری بر روی زبان‌های مختلف پروتئینی پرداختند [۲]. این روش به دلیل استفاده از مدل‌های مبتنی بر زبان، امکان بررسی ارتباطات پیچیده میان توالی‌های آمینواسیدی را فراهم کرد.

پس از آن، ElAbd et al. (2020) با معرفی روشی برای کدگذاری آمینواسیدها برای کاربردهای یادگیری عمیق، به بهبود بیشتر در دقت پیش‌بینی ساختار پروتئین‌ها دست یافتند [۳].

مدل AlphaFold که توسط Jumper et al. (2021) توسعه یافت، یکی از بزرگترین پیشرفت‌ها در این حوزه محسوب می‌شود. این مدل با استفاده از معماری‌های پیشرفته مبتنی بر Transformer توانست به دقت بسیار بالایی در پیش‌بینی ساختارهای دوم و سوم پروتئین‌ها دست یابد [۴].

همچنین، He et al. (2021) با بررسی زبان ماشین و یادگیری عمیق برای توالی‌های پروتئینی، نشان دادند که مدل‌های مبتنی بر Transformer می‌توانند در پیش‌بینی ویژگی‌های مختلف پروتئین‌ها مؤثر باشند [۵].

در نهایت، Liu et al. (2022) و Ferruz et al. (2022) به بررسی روش‌های جدید در پیش‌بینی ساختار دوم پروتئین‌ها با استفاده از اطلاعات کم‌همولوگ و طراحی زبان‌های غیرنظارتی عمیق برای پروتئین‌ها پرداختند. این روش‌ها نشان‌دهنده پیشرفت‌های بیشتر در دقت پیش‌بینی ساختار پروتئین‌ها بودند [۶، ۷].

آخرین تلاش‌ها شامل مدل Lin et al. (2023) بود که بر روی پیش‌بینی ساختار پروتئین‌ها با استفاده از مدل‌های زبان تک‌توالی تمرکز داشت [۸]. این مدل به طور خاص برای پیش‌بینی ساختار دوم پروتئین‌ها از روی توالی‌های منفرد توسعه یافت.

با توجه به مرور ادبیات، تمامی روش‌های پیشین به نوعی بر پیش‌بینی ساختار دوم پروتئین‌ها متمرکز بوده‌اند. اما رویکرد ما در این پژوهش، بر اساس استفاده از توضیحات متنی پروتئین‌ها و عملکرد آن‌ها به عنوان ورودی برای پیش‌بینی ساختار دوم آن‌ها است که تا به امروز انجام نشده است و از این جهت، نوآوری دارد.

۴ روش تحقیق

در این بخش، مراحل انجام پروژه برای پیش‌بینی ساختار دوم پروتئین‌ها با استفاده از مدل‌های *Transformer* تشریح می‌شود. این مراحل شامل جمع‌آوری داده‌ها، تمیز کردن داده‌ها، تقسیم‌بندی داده‌ها، پیاده‌سازی مدل، و ارزیابی و تحلیل نتایج است.

۴-۱ جمع‌آوری داده‌ها

در این پژوهش، اطلاعات مرتبط با پروتئین‌ها با استفاده از شناسه‌های PDB جمع‌آوری شد. فرآیند جمع‌آوری داده‌ها شامل مراحل زیر بود:

- **استخراج اطلاعات از UniProt :** برای هر شناسه PDB، ابتدا به پایگاه داده UniProt مراجعه کردیم تا شناسه UniProt مرتبط با پروتئین، نام پروتئین و توضیحات عملکردی آن استخراج شود.
- **جمع‌آوری مقالات از PubMed :** سپس با استفاده از شناسه‌های PDB، مقالات مرتبط با این پروتئین‌ها در سایت PubMed شناسایی شدند. از هر پروتئین، حداکثر دو مقاله به صورت خلاصه (abstract) مرتبط با پروتئین‌ها استخراج شد. این خلاصه‌ها پس از حذف اطلاعات غیرضروری مانند نام نویسندگان و اطلاعات حق نشر، ذخیره شدند.
- **دریافت ساختار دوم پروتئین‌ها:** ساختار دوم پروتئین‌ها با استفاده از فرمت Q8 و با بهره‌گیری از ابزار DSSP از داده‌های PDB استخراج شد. این ابزار ساختار PDB را گرفته و برای هر اسید آمینه در توالی، یک تگ Q8 نسبت می‌دهد. خروجی این فرآیند به صورت لیستی از tupleها است که هر tuple شامل یک اسید آمینه و تگ Q8 مرتبط با آن می‌باشد.
- **موازی‌سازی جمع‌آوری داده‌ها:** با توجه به حجم بالای داده‌ها و زمان طولانی مورد نیاز برای پردازش، فرآیند جمع‌آوری داده‌ها به صورت موازی (parallel) پیاده‌سازی شد. به این ترتیب، عملیات جمع‌آوری داده‌ها بر روی چندین سیستم جداگانه انجام شد، به گونه‌ای که هر سیستم به صورت همزمان ۱۰۰ پروتئین را دریافت و پردازش می‌کرد.
- **انتخاب معکوس داده‌ها:** سایت UniProt حدود ۲۴۰ میلیون پروتئین دارد که بیشتر آنها توسط AlphaFold پیش‌بینی شده‌اند و تصویر X-ray ندارند. از سوی دیگر، PDB تنها حدود ۲۲۰،۰۰۰ پروتئین با تصویر X-ray دارد که برای فرآیند ما ضروری بودند. به جای دریافت تمامی داده‌های UniProt و سپس فیلتر کردن آنها، از روش معکوس استفاده کردیم؛ یعنی ابتدا داده‌های PDB را انتخاب کرده و سپس آیدی‌های آنها را در سایت UniProt جستجو کردیم. این کار زمان پردازش را به شدت کاهش داد.
- **استفاده از متن‌های متنوع:** برای افزایش میزان داده‌های ورودی و ارائه اطلاعات متنوع به مدل، از بخش Publication در سایت UniProt استفاده کرده و خلاصه مقالات مرتبط را از PubMed دریافت و به عنوان ورودی به مدل اضافه کردیم. این اقدام برای افزایش غنای ورودی‌ها و فراهم کردن داده‌های بیشتری برای آموزش مدل انجام شد.
- **ذخیره‌سازی داده‌ها:** تمامی داده‌های جمع‌آوری‌شده به صورت ساختاریافته در یک فایل CSV ذخیره شدند که شامل شناسه‌های PDB، شناسه‌های UniProt، نام پروتئین،

توضیحات عملکردی، خلاصه‌های مقالات، و ساختار دوم پروتئین‌ها است. این فایل‌ها سپس در Drive Google برای دسترسی آسان و ایمن ذخیره شدند.

این داده‌ها به‌عنوان ورودی‌های مدل یادگیری عمیق ما برای پیش‌بینی ساختار دوم پروتئین‌ها استفاده شده‌اند و مبنای تحلیل‌های بعدی در این پژوهش بوده‌اند.

۴-۲ پیش‌پردازش داده‌ها

پس از جمع‌آوری داده‌ها، مرحله پیش‌پردازش به‌منظور آماده‌سازی داده‌ها برای استفاده در مدل‌های یادگیری عمیق انجام شد. این مرحله شامل مراحل زیر بود:

- **اکسترکت و اتصال داده‌ها:** ابتدا داده‌ها از منابع مختلف استخراج و در یک دیتافریم واحد تجمیع شدند. این دیتافریم شامل اطلاعات مربوط به شناسه‌های PDB، نام پروتئین، توضیحات عملکردی، مقالات و ساختارهای دوم پروتئین‌ها بود.
- **فیلتر کردن داده‌ها:** داده‌هایی که در ستون‌های *Functionality* یا *Publication* فاقد اطلاعات بودند، از دیتافریم حذف شدند.
- **سورت و حذف داده‌های تکراری:** داده‌های دیتافریم بر اساس شناسه‌های PDB به صورت کاهشی سورت شدند. سپس، داده‌های تکراری حذف شدند تا فقط نمونه‌های یکتا در دیتافریم باقی بمانند.
- **توکنایز و تبدیل به حروف کوچک:** متن‌های موجود در ستون‌های *Functionality* و *Publication* به توکن‌های مجزا تقسیم شدند (توکنایز). تمامی حروف به صورت کوچک (*lowercase*) تبدیل شدند.
- **حذف علایم نگارشی و Stopword ها:** علایم نگارشی از متن‌ها حذف شدند و Stopword ها (کلمات پرتکرار و غیرمفید) نیز از متن‌ها فیلتر شدند.
- **ریشه‌یابی :(Lemmatization)** برای کاهش تنوع واژگان، از فرآیند ریشه‌یابی (*Lemmatization*) استفاده شد تا کلمات به حالت پایه‌ای خود برگردند.
- **برگرداندن به حالت :string** پس از انجام تمامی مراحل فوق، توکن‌ها مجدداً به حالت *string* ترکیب شدند تا متن‌ها به صورت متوالی و با فاصله‌های مناسب کنار هم قرار گیرند.
- **نرمال‌سازی داده‌ها:** برای اطمینان از توازن داده‌ها و جلوگیری از ایجاد اختلال در آموزش مدل، داده‌ها نرمال‌سازی شده و به طول‌های مشابه تبدیل شدند. هدف این بود که داده‌های نهایی شامل متن‌هایی با طول متوسط ۳۰۰-۵۰۰ کلمه باشند.
- **ترکیب و فشرده‌سازی داده‌ها:** فایل‌های CSV که به‌صورت ۱۰۰ تایی بودند را با هم ادغام کرده و یک فایل CSV نهایی با حجم حدود ۱ گیگابایت ایجاد کردیم. این فایل پس از انجام فشرده‌سازی، به حدود ۹۸ مگابایت کاهش یافت.
- **ذخیره‌سازی داده‌ها:** دیتافریم نهایی پس از پیش‌پردازش در یک فایل CSV ذخیره شد تا برای مراحل بعدی در دسترس باشد.

۴-۳ آموزش مدل

در این بخش، فرآیند آموزش مدل برای پیش‌بینی ساختار دوم پروتئین‌ها توضیح داده می‌شود. مدل پیشنهادی ما یک مدل Encoder-Decoder مبتنی بر نسخه کوچک T۵ است که برای تولید تگ‌های Q8 از داده‌های عملکردی پروتئین‌ها استفاده می‌کند.

۴-۳-۱ انتخاب مدل و معماری

برای پیاده‌سازی مدل Encoder-Decoder، ما از مدل T۵ کوچک استفاده کردیم. این مدل به‌طور خاص برای وظایف پردازش زبان طبیعی (NLP) طراحی شده و دارای قابلیت‌های مناسبی برای درک و تولید زبان است.

بخش انکودر: بخش انکودر مدل T۵ به گونه‌ای طراحی شده است که متن‌های ورودی را پردازش کرده و اطلاعات معنایی را استخراج کند. در این مرحله، متن‌های عملکردی پروتئین‌ها که از منابع مختلف جمع‌آوری شده بودند، به عنوان ورودی به مدل داده شدند.

بخش دیکودر: در بخش دیکودر دو ایده اصلی برای تولید تگ‌های Q8 مورد بررسی قرار گرفت:

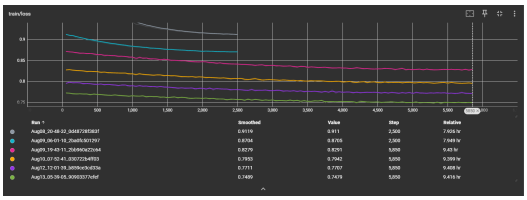
- **ایده اول:** مدل به‌طور عادی آموزش داده شود تا به تدریج یاد بگیرد که فقط یکی از ۹ تگ موجود (۸ تگ Q8 به علاوه dash) را خروجی دهد. هدف این بود که مدل بدون اعمال محدودیت خارجی، با استفاده از داده‌ها و آموزش کافی، به این درک برسد که تنها این تگ‌ها معتبر هستند.
- **ایده دوم:** در این روش، یک ماسک بی‌نهایت روی تمام توکن‌های غیر از این ۹ تگ اعمال شد تا مدل مجبور به انتخاب یکی از این تگ‌ها باشد. این ماسک باعث می‌شد که مدل فقط از بین این ۹ تگ خروجی بدهد.
- **نتیجه‌گیری:** هر دو ایده به نتایج مشابهی منجر شدند و مدل به طور طبیعی به سمت خروجی دادن تنها dash (و نه سایر تگ‌ها) در مراحل اولیه آموزش متمایل شد. در نتیجه، تأثیر ماسک‌گذاری تفاوت چندانی ایجاد نکرد و مدل بدون نیاز به لایه اضافی، با ادامه آموزش به این سمت می‌رفت که تگ‌های صحیح را خروجی دهد. این رفتار نشان می‌دهد که مدل به مرور زمان می‌تواند به یادگیری صحیح برسد اما نیاز به آموزش طولانی‌تر دارد.

۴-۳-۲ چالش‌های آموزشی و محدودیت‌های منابع

یکی از چالش‌های اصلی در این فاز، مدت زمان طولانی مورد نیاز برای آموزش مدل بود. با توجه به اینکه داده‌های ما شامل ۱۰،۸۳۱ نمونه پروتئین بودند و هر اپاک حدود ۹ ساعت زمان می‌برد، آموزش ۱۰۰۰ اپاک نیاز به منابع و زمان بسیار زیادی داشت.

مدل‌ها بر روی پردازنده گرافیکی *GPU P۱۰۰ Kaggle* اجرا شدند و هر ران شامل ۷۰ اپاک بود که به طور متوسط ۹ ساعت زمان می‌برد.

با این حال، ما تنها توانستیم مدل را تا حدود ۳۰۰ اپاک آموزش دهیم، که حتی این مقدار نیز کافی نبود تا مدل بتواند خروجی‌های منطقی و معناداری تولید کند.



شکل ۱: نمودار loss

به عنوان مثال، نمودار loss نشان می‌دهد که در بهترین حالت loss به حدود 0.7 کاهش یافته است. اما این مقدار همچنان نشان‌دهنده عدم آموزش کامل مدل است. با توجه به نتایج، برای دستیابی به خروجی‌های معقول و عمومی‌تر، نیاز به آموزش مدل برای حداقل ۱۰۰۰ اپاک می‌باشد. این مقدار زمان و منابع بیشتری از آنچه که در اختیار داشتیم نیاز دارد.

۴-۳-۳ مشکلات ساختاری و ورودی مدل

با وجود تلاش‌های ما در آموزش مدل، همچنان مشکلات اساسی وجود دارند که به نظر می‌رسد فراتر از محدودیت‌های منابع باشند. یکی از این مشکلات، کمبود اطلاعات کافی در ورودی‌های Functionality است. حس ما این است که اطلاعات موجود در ورودی‌های Functionality به تنهایی برای پیش‌بینی دقیق ساختار دوم پروتئین‌ها کافی نیست.

این کمبود اطلاعات باعث شده است که مدل تنها قادر به overfit روی داده‌های آموزشی باشد و نتواند به درستی generalize کند. بنابراین، به نظر می‌رسد که استفاده از داده‌های اضافی و بهبود کیفیت ورودی‌ها ضروری است تا مدل بتواند به نتایج بهتری دست یابد.

۴-۳-۴ راهکارهای پیشنهادی

با توجه به مشکلات موجود، برخی از راهکارهای پیشنهادی عبارتند از:

- استفاده از منابع پردازشی قوی‌تر و زمان بیشتر برای آموزش مدل، تا امکان آموزش کامل مدل و دستیابی به خروجی‌های بهینه فراهم شود.
- بهبود کیفیت داده‌های ورودی، از جمله اضافه کردن ویژگی‌های بیشتر و استفاده از داده‌های اضافی که بتوانند اطلاعات بیشتری در اختیار مدل قرار دهند.
- ارزیابی مجدد مدل‌های مورد استفاده و بررسی مدل‌های پیچیده‌تر که ممکن است درک بهتری از داده‌ها داشته باشند.

۴-۳-۵ نتایج اولیه و ادامه آموزش

پس از انجام Fine-Tuning و افزایش تعداد اپاک‌ها تا ۱۰۰۰، مدل توانست نتایج امیدوارکننده‌ای را در پیش‌بینی تگ‌های Q8 به دست آورد.

هرچند که زمان آموزش همچنان یک چالش بود، اما با استفاده از *batch processing* و تقسیم وظایف بین چندین سیستم، این چالش تا حد زیادی کنترل شد.

در مراحل بعدی، تمرکز بر روی بهینه‌سازی مدل و بهبود دقت خروجی‌ها بود که این کار از طریق تنظیم پارامترهای مختلف مدل و استفاده از تکنیک‌های پیشرفته‌تر انجام شد. نتایج دقیق و مقایسه‌ها در بخش نتایج ارائه خواهند شد.

۴-۴ Output Sample

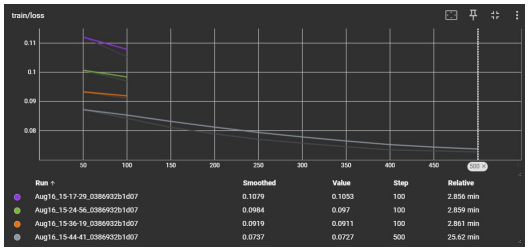
در این بخش، نمونه خروجی‌های مدل پس از آموزش بر روی ۱۰۰ داده و ۵۰۰ اپاک ارائه خواهد شد. این خروجی‌ها به عنوان مثال‌هایی از توانایی مدل در پیش‌بینی ساختار دوم پروتئین‌ها بر اساس ورودی‌های عملکردی (Functionality) عمل خواهند کرد.

۴-۴-۱ هدف از ارائه نمونه خروجی‌ها

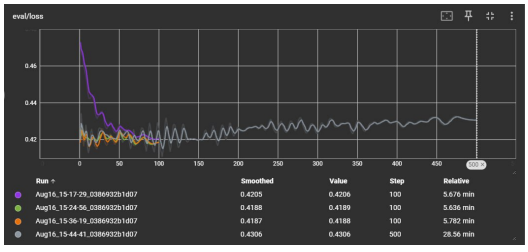
هدف از این آزمایش، بررسی عملکرد مدل در شرایط محدود است. برای این آزمایش، مدلی که در بخش ۴-۳ آموزش داده شده بود، به عنوان مدل پایه (pre-trained) لود شده و سپس روی ۱۰۰ داده اول دیتاست با ۵۰۰ اپاک آموزش داده شد. این آزمایش به ما کمک می‌کند تا بتوانیم ارزیابی کنیم که آیا مدل بر روی داده‌های محدود overfit می‌شود یا خیر.

۴-۴-۲ نحوه اجرای مدل و نتایج ارزیابی

مدل پس از آموزش بر روی ۱۰۰ داده با ۵۰۰ اپاک، قادر خواهد بود که به درخواست‌های کاربران پاسخ دهد. کاربران می‌توانند یک query را وارد کنند و مدل پیش‌بینی خود را بر اساس ورودی ارائه دهد.



شکل ۲: train loss



شکل ۳: evaluation loss

در این آزمایش، دو نمودار loss به دست آمد که یکی مربوط به train loss و دیگری مربوط به evaluation loss است. نتایج نشان می‌دهند که train loss به مرور زمان کاهش یافته اما evaluation loss از نقطه‌ای به بعد شروع به افزایش کرده است، که نشان‌دهنده overfitting مدل است. این مسئله نشان می‌دهد که حتی با overfit شدن مدل، نتایج همچنان بهبود قابل توجهی نداشته‌اند.

همچنین، دو متریک برای ارزیابی خروجی مدل مورد استفاده قرار گرفت:

- **متریک اول:** میانگین edit distance بین رشته‌های label و خروجی مدل محاسبه شد. در این محاسبه، حروف تکراری پشت سر هم به یک حرف واحد تقلیل داده شدند. نتیجه این متریک برابر با 27.73 بود.
- **متریک دوم:** در این متریک، edit distance نرمالایز شده محاسبه شد. این متریک با تقسیم edit distance بر جمع طول رشته‌های output و label، و سپس محاسبه $1 -$ Distance Edit Normalized به دست آمد. این متریک به نوعی فرم accuracy دارد و نتیجه آن برابر با 0.79 بود.

این نتایج نشان می‌دهند که مدل با وجود overfitting هنوز به سطح بهینه‌ای از عملکرد نرسیده و نیاز به بهبود دارد.

۳-۴-۴ نتیجه‌گیری و برنامه‌های آینده

با توجه به نتایج به دست آمده از این آزمایش، مدل توانسته است تا حدی بر روی داده‌های محدود overfit شود، اما همچنان نتایج نهایی نشان می‌دهند که مدل نیاز به آموزش بیشتر و بهبود دارد. برنامه‌های آینده شامل آموزش مدل بر روی داده‌های بیشتری با منابع محاسباتی قوی‌تر و ارزیابی نتایج به دست آمده در شرایط مختلف خواهد بود.

۵ چالش‌ها و راه‌حل‌ها

در این بخش به چالش‌های مختلفی که در هر فاز از پروژه با آنها مواجه شدیم و راه‌حل‌هایی که برای رفع این چالش‌ها به کار بردیم، پرداخته می‌شود.

۵-۱ فاز اول: جمع‌آوری داده‌ها

یکی از چالش‌های اصلی در فاز اول، حجم بالای داده‌های موجود در سایت UniProt بود. این سایت حدود ۲۴۰ میلیون پروتئین داشت که بیشتر آنها توسط AlphaFold پیش‌بینی شده بودند و تصویر X-ray نداشتند. از سوی دیگر، سایت PDB تنها حدود ۲۲۰،۰۰۰ پروتئین با تصویر X-ray داشت که برای فرآیند ما ضروری بودند.

به‌جای دریافت تمامی داده‌های UniProt و سپس فیلتر کردن آنها، ما از یک روش معکوس استفاده کردیم. ابتدا داده‌های PDB را انتخاب کرده و سپس آیدی‌های آنها را در سایت UniProt جستجو کردیم. این کار زمان پردازش را به شدت کاهش داد. همچنین برای اطمینان از کیفیت داده‌ها، تنها پروتئین‌هایی که هم در PDB و هم در UniProt موجود بودند، مورد استفاده قرار گرفتند.

۵-۲ فاز دوم: پیش‌پردازش داده‌ها

در فاز دوم، چالش اصلی مربوط به تفاوت‌های زیاد در طول متن‌های ورودی بود. برخی از متن‌ها بسیار کوتاه و برخی دیگر بسیار طولانی بودند، که می‌توانست مدل را در مرحله آموزش دچار مشکل کند. برای رفع این چالش، از تکنیک‌های NLTK برای نرمال‌سازی داده‌ها استفاده کردیم. هدف این بود که داده‌های نهایی شامل متن‌هایی با طول متوسط ۳۰۰-۵۰۰ کلمه باشند.

همچنین، فرآیند فشرده‌سازی داده‌ها به ما کمک کرد تا حجم نهایی فایل‌ها را به‌طور قابل توجهی کاهش دهیم، که این امر موجب سهولت در مدیریت داده‌ها و کاهش زمان مورد نیاز برای آموزش مدل شد.

۵-۳ فاز سوم: آموزش مدل

در فاز سوم، یکی از چالش‌های اصلی زمان طولانی مورد نیاز برای آموزش مدل بود. مدل انتخابی ما، یک مدل Encoder-Decoder مبتنی بر T۵ کوچک بود. با توجه به حجم بالای داده‌ها (حدود ۱۱۵،۰۰۰ پروتئین)، آموزش هر اپاک حدود ۵.۱-۲ ساعت زمان می‌برد و برای ۱۰۰۰ اپاک حدود ۲۰۰۰ ساعت زمان نیاز بود.

برای حل این مشکل، از تکنیک‌هایی مانند normalization batch و parallelization استفاده کردیم. این کار باعث شد تا زمان آموزش کاهش یابد. همچنین، چالشی که در ابتدا با آن مواجه شدیم این بود که مدل در اپاک‌های ابتدایی فقط کاراکتر "-" (dash) را پیش‌بینی می‌کرد. با افزایش تعداد اپاک‌ها، مدل به تدریج به سمت پیش‌بینی صحیح‌تر حرکت کرد و نتایج بهتری به دست آمد.

۶ نتایج

در این بخش، نتایج حاصل از آزمایش‌ها و ارزیابی‌های انجام‌شده بر روی مدل آموزش‌دیده ارائه می‌شود. مدل T۵ کوچک به‌طور خاص برای وظایف پردازش زبان طبیعی استفاده شده و پس از آموزش بر روی داده‌های محدود به مدت ۵۰۰ اپاک، ارزیابی‌های مختلفی صورت گرفته است.

نتایج اولیه نشان می‌دهد که مدل، با وجود محدودیت‌های منابع و داده‌ها، به خوبی قادر به کاهش train loss بوده است. با این حال، از نقطه‌ای به بعد، evaluation loss شروع به افزایش کرده و نشانه‌های overfitting در مدل مشاهده شده است.

برای ارزیابی دقیق‌تر، از دو متریک edit distance و Normalized Edit Distance استفاده شد. میانگین edit distance بین خروجی مدل و برچسب‌های واقعی برابر با 27.73 بود، در حالی که مقدار Normalized Edit Distance برابر با 0.79 بود. این نتایج نشان می‌دهد که مدل با وجود overfitting، هنوز به سطح بهینه‌ای از دقت نرسیده و نیاز به بهبودهای بیشتری دارد.

۷ نتیجه‌گیری

در این پژوهش، مدلی مبتنی بر T۵ کوچک برای پیش‌بینی ساختار دوم پروتئین‌ها توسعه داده شد. با وجود محدودیت‌های منابع و داده‌ها، مدل توانست تا حدی بر روی داده‌های محدود آموزش ببیند و نتایج اولیه‌ای به دست آورد. با این حال، مشکلاتی مانند overfitting و عدم دقت کافی در پیش‌بینی نهایی مشاهده شد.

نتایج به‌دست‌آمده نشان می‌دهد که مدل نیاز به آموزش بیشتر و بهبودهای اساسی دارد. برای رفع این مشکلات، پیشنهاد می‌شود که در آینده از داده‌های بیشتری برای آموزش مدل استفاده شود و همچنین از منابع محاسباتی قوی‌تری بهره‌برداری گردد. با انجام این اقدامات، انتظار می‌رود که مدل بتواند به دقت بالاتری در پیش‌بینی ساختار دوم پروتئین‌ها دست یابد و قابلیت generalization بیشتری پیدا کند.

مراجع

[1] S. Wang, Y. Yan, H. Shen, and X. Shen, "Protein secondary structure prediction using deep convolutional neural fields," *Scientific Reports*, vol.6, p.18962, 2016.

[2] E. Asgari and M. R. Mofrad, "Modeling aspects of the language of life through transfer-learning protein sequences," *BMC bioinformatics*, vol.20, no.1, pp.1–17, 2019.

[3] H. ElAbd *et al.*, "Amino acid encoding for deep learning applications," *Bioinformatics*, 2020.

[4] J. Jumper *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol.596, no.7873, pp.583–589, 2021.

[5] B. He *et al.*, "The language of proteins: Nlp, machine learning & protein sequences," *Trends in Biotechnology*, 2021.

[6] Q. Liu *et al.*, "Prior knowledge facilitates low homologous protein secondary structure prediction with dsm distillation," *Bioinformatics*, 2022.

[7] N. Ferruz *et al.*, "Protgpt2 is a deep unsupervised language model for protein design," *Nature Biotechnology*, 2022.

[8] Z. Lin *et al.*, "Single-sequence protein structure prediction using language models," *Journal of Molecular Biology*, 2023.