



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

باسمه تعالی

## فرم پیشنهاد پروژه کارشناسی



دانشکده مهندسی کامپیوتر

**تحويل پیشنهاد پروژه به دانشکده و ثبت نهایی آن در پورتال:** (این قسمت توسط کارشناسان آموزش دانشکده تکمیل می شود).

تاریخ تحويل پیشنهاد پروژه به آموزش دانشکده:

تاریخ ثبت نهایی در پورتال آموزشی دانشگاه:

### مشخصات دانشجو:

نام و نام خانوادگی: محمد چوپان

شماره دانشجویی: ۹۸۳۱۱۲۵

رایانامه (ایمیل) دانشجو: mohamadkhoe@aut.ac.ir

نیمسال و سال تحصیلی ثبت نام پروژه: نیمسال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

**توضیح ۱:** دانشجو موظف است حداکثر دو ماه پس از ثبت نام پروژه فرم تکمیل شده پیشنهاد پروژه را، که به امضای استاد راهنمای او رسیده است، به آموزش دانشکده تحويل دهد. انجام سر وقت این مرحله نشان دهنده بخشی از رعایت زمانبندی انجام پروژه توسط دانشجو است.

**توضیح ۲:** آموزش دانشکده پیشنهاد پروژه دریافتی را جهت تعیین داور و انجام داورى در اختیار گروه آموزشی استاد راهنمای دانشجو قرار می دهد. گروه های آموزشی حداکثر طی دو ماه داورى را انجام داده و در صورت تصویب در گروه، پیشنهاد پروژه را جهت تصویب در دانشکده و ثبت در پورتال آموزشی دانشگاه در اختیار آموزش دانشکده قرار می دهند. دانشجویان موظفند با داور(ان) پیشنهاد پروژه خود در ارتباط بوده و نظرات آنان را، با راهنمایی استاد راهنمای خود و در مهلت مقرر گروه برای تصویب پیشنهاد پروژه، بر روی پیشنهاد پروژه خود اعمال نمایند.

**توضیح ۳:** مهلت درج نمره پروژه دانشجویانی که در نیمسال اول یا در تابستان سال تحصیلی پروژه را اخذ نموده اند، سی ام مهر سال تحصیلی بعد و برای دانشجویانی که در نیمسال دوم پروژه را اخذ نموده اند، سی و یکم ام فروردین سال تحصیلی بعد است.

**توضیح ۴:** فاصله زمانی بین ثبت نهایی پیشنهاد پروژه (تصویب شده) در پورتال آموزشی دانشگاه و دفاع از پروژه حداقل سه ماه است و امکان دفاع قبل از سپری شدن این فاصله زمانی وجود ندارد. همچنین، دفاع از پروژه کارشناسی با اعلان عمومی و با حضور مخاطبان در حضور داوران انجام خواهد شد. لازم است دانشجویان حداقل سه هفته قبل از فرارسیدن مهلت درج نمره پروژه (توضیح ۳)، پایان نامه تایپ شده خود را، که به تأیید استاد راهنما رسیده است، در اختیار آموزش دانشکده و داور(ان) پروژه قرار داده و مقدمات برگزاری جلسه دفاع را، با هماهنگی آموزش دانشکده، فراهم آورند.

**توضیح ۵:** لازم است دانشجویان رویه دانشگاه صنعتی امیرکبیر با عنوان «چگونگی ثبت نام، تصویب، و دفاع از پایان نامه در مقطع کارشناسی» را که با شماره AUT-PR-3210 بر روی سایت معاونت آموزشی دانشگاه قرار گرفته است مطالعه کنند.

تاریخ:

امضای دانشجو:

**استاد راهنمای پروژه:**

تاریخ:

امضا:

نام و نام خانوادگی:

## عنوان پروژه:

عنوان فارسی: ارزیابی ترکیب شبکه‌های عصبی گرافی و مدل پوینت نت برای پردازش داده‌های گرافی و ابرنقاط  
عنوان انگلیسی:

Assessment of the Hybrid of Graph and Point Net Model for Processing Graph Neural networks and Point Cloud Data

## داور(ان) پیشنهاد پروژه:

### داور اول:

نام و نام خانوادگی:

امضا:

تاریخ:

### داور دوم:

نام و نام خانوادگی:

امضا:

تاریخ:

**توضیح:** با امضای این قسمت داور(ان) محترم تأیید می‌کنند که

- ۱- دانشجو، با راهنمایی استاد راهنمای خود، اصلاحات مورد نظر داور(ان) را انجام داده و عنوان و محتوای پیشنهاد پروژه از نظر ایشان قابل قبول است.
- ۲- دانشجو با مفاهیم پیش‌نیاز و مهارت‌های ضروری و پایه انجام این پروژه آشنایی داشته یا کسب آن برای دانشجو در طول انجام پروژه امکان‌پذیر است.
- ۳- موارد زیر در پیشنهاد پروژه مورد توجه قرار گرفته است:
  - عنوان پروژه به طور کامل و دقیق موضوع پروژه را نشان می‌دهد و محتوای پروژه با عنوان پروژه کاملاً مطابقت دارد.
  - پیشنهاد پروژه شامل بخش‌های مقدمه، مرور پیشینه پژوهش، رویکرد پیشنهادی، روش ارزیابی، مراحل و زمان‌بندی انجام پروژه، امکانات لازم و لیست مراجع و منابع است.
  - اجزای سامانه مورد نظر پروژه در یک نمودار بلوکی نشان داده شده و ورودی‌ها و خروجی‌های آن مشخص شده‌اند.
  - تأکید پروژه بر روی مسائل عملی و علمی و مهارت‌های مهندسی کامپیوتر است و پروژه منجر به توسعه نرم‌افزار، سخت‌افزار یا ترکیبی از آن دو و با درجه سختی و حجم مناسب یک پروژه سه واحدی است.
  - پروژه بر مبنای استفاده از دروس کارشناسی تعریف شده است.
  - چنانچه قرار است در پروژه از ابزارها، نرم‌افزارها، یا محیط‌های آماده استفاده شود، این موارد با صراحت بیان شده و مشخص شده است چه بخش‌هایی و با چه مقداری تلاش سهم دانشجو است.
  - پروژه علاوه بر بخش مطالعاتی-نظری، حدود ۱۵۰ ساعت کار عملی لازم داشته و انجام آن حداقل ۳ ماه زمان نیاز دارد.

## تصویب پیشنهاد پروژه:

### تصویب در گروه آموزشی:

نام و نام خانوادگی مدیر گروه:

امضا:

تاریخ:

### تصویب در شورای آموزشی-پژوهشی دانشکده:

نام و نام خانوادگی معاون آموزشی:

امضا:

تاریخ:

**تعریف پروژه:** (دانشجو می‌تواند با اضافه کردن فاصله لازم بر روی فایل قابل ویرایش این سند، توضیحات خود را در هر یک از قسمت‌های زیر تایپ کند).

## ۱- مقدمه (بیان مسئله کاربردی، ضرورت، انگیزه، اهداف، و چالش‌های انجام این پروژه):

در دهه اخیر، پیشرفت‌های چشمگیر در زمینه‌های پردازش داده‌ها و بینایی کامپیوتری فرصت‌های منحصر به فردی را برای تجزیه و تحلیل داده‌ها و ساختارهای پیچیده ایجاد کرده است. این پیشرفت‌ها در حوزه‌های متنوعی از بینایی کامپیوتر تا شبکه‌های اجتماعی و علوم مهندسی اثرات قابل ملاحظه‌ای داشته و تا حد زیادی به بهبود فهم و تفسیر دقیق داده‌ها کمک کرده است.

در این سیاق دو نوع داده‌ای که به طور ویژه نقش مهمی در زمینه‌های مذکور ایفا می‌کنند داده‌های گرافی و ابرنقاط<sup>۱</sup> هستند. داده‌های گرافی به دسته‌ای از داده‌ها گفته می‌شود که بتوان آن‌ها را به صورت ساختمان داده گراف نمایش داد. ساختار داده‌های گرافی به عنوان ابزاری قدرتمند برای نمایش روابط پیچیده میان عناصر مختلف مورد استفاده قرار می‌گیرند. از جمله کاربردهای مهم داده‌های گرافی مدل‌سازی، شبکه‌های اجتماعی، تجزیه و تحلیل توالی‌های ژنی و تحلیل تعاملات در شبکه‌های اطلاعاتی می‌باشد. برای مثال از الگوریتم‌های مرتبط با تئوری گراف‌ها می‌توان برای پیدا کردن کوتاه‌ترین مسیر در ترافیک‌ها و یا روابط بین افراد و سلاقی آن‌ها در شبکه‌های اجتماعی استفاده کرد. زمانبندی کلاس‌ها، رتبه‌بندی در مسابقات ورزشی و... تنها بخشی از کاربرد های ساختمان داده گراف می‌باشد [۱]. علاوه بر این موارد گراف‌ها در زمینه هوش مصنوعی نیز کاربردهای زیادی دارند. شبکه‌های عصبی گرافی<sup>۲</sup> که در سال‌های اخیر توجه زیادی به آن‌ها شده است باعث پیشرفت روزافزون در این زمینه شده‌اند.

شبکه‌های عصبی گرافی یکی از روش‌های مبتنی بر یادگیری عمیق<sup>۳</sup> است که بر حول محور گراف‌ها بررسی می‌شود. در این نوع شبکه‌ها ما ابتدا نیاز داریم که داده‌های ورودی خود را بررسی کرده و با توجه به ساختارمند و یا غیر ساختارمند بودن آن‌ها ارتباطی بین داده‌ها برای ساختن گراف پیدا کنیم. در مرحله بعدی نوع گراف خود را انتخاب کرده و در نهایت یک تابع برای دقت خود تعریف می‌کنیم که هرکدام از این موارد با توجه به نوع داده‌های ما و هدف ما قابل تغییر است. محاسبات لایه‌های شبکه تعریف شده نیز با توجه به نوع شبکه گرافی مورد استفاده قابل تغییر است. در شکل ۱ انواعی از شبکه‌های عصبی گرافی را مشاهده می‌کنید که بر اساس معیارهای گفته‌شده دسته‌بندی شده‌اند. این شبکه‌ها در بسیاری از زمینه‌ها کاربرد دارند برای مثال به کاربردهای آن در زمینه پردازش تصویر، دسته‌بندی تصاویر، سیستم‌های توصیه و تخمین ساختار مولکولی می‌توان اشاره کرد. در شکل ۱ انواعی از شبکه‌های عصبی گرافی را مشاهده می‌کنید که به ۳ دسته اصلی تقسیم می‌شوند که در ادامه به توضیح آن‌ها می‌پردازیم.

- ۱- پودمان انتشار<sup>۴</sup>: در این نوع از شبکه‌های عصبی گرافی داده‌ها بین برگ‌ها انتشار پیدا کرده تا هم اطلاعات ویژگی‌ها و هم اطلاعات توپولوژیکی انتشار پیدا کنند. در این نوع پودمان‌ها از عملگرهای پیچشی و بازگشتی برای تجمیع داده‌ها از همسایه‌های یک برگ استفاده می‌شود.
  - ۲- پودمان نمونه‌برداری<sup>۵</sup>: هنگامی که گراف‌ها بزرگ هستند از پودمان‌های نمونه‌برداری معمولاً برای انتشار گراف مورد نیاز استفاده می‌شود.
  - ۳- پودمان تجمیعی<sup>۶</sup>: این پودمان‌ها معمولاً زمانی استفاده می‌شوند که ما نیاز به استخراج ویژگی‌های از برگ‌ها داریم و یا نیاز به نمایش زیرگراف‌ها داریم.
- در انجام این پروژه پودمان‌های تجمیعی بیشتر مورد توجه ما هستند. پودمان‌های تجمیعی خود به دو دسته اصلی تقسیم می‌شوند که عبارتند از:

- ۱- پودمان‌های تجمیعی مستقیم<sup>۷</sup>: این نوع پودمان‌های تجمیعی نمایش در سطح گراف را به طور مستقیم از برگ‌ها با استراتژی‌های انتخاب برگ همسایه یاد می‌گیرند.

برای به دست آوردن ویژگی‌های سراسری در این گراف‌ها عملیاتی مانند جمع، ضرب و تقسیم با برگ‌ها انجام می‌شود تا ویژگی‌های سراسری گراف‌ها را استخراج کند. یکی دیگر از روش‌ها مرتب کردن برگ‌ها با معیاری<sup>۸</sup> مشخص است که ابتدا کل گراف را مرتب می‌کنیم و سپس با استفاده از یک شبکه‌عصبی پیچشی ویژگی‌ها را استخراج می‌کنیم.

- ۲- پودمان‌های تجمیعی سلسله مراتبی<sup>۸</sup>: در قسمت قبلی مدل ساختار گراف را با استفاده مستقیم از برگ‌ها یاد می‌گرفت اما در این نوع از پودمان تجمیعی ساختار را از لایه‌های مختلف گراف یاد می‌گیرد. روش‌های مختلفی برای انجام اینکار وجود دارد. یکی از این‌ها روش خودتوجهی است که ما از آن در پروژه خود استفاده خواهیم کرد [۲].

<sup>1</sup> Point cloud data

<sup>2</sup> Graph Neural Networks (GNN's)

<sup>3</sup> Deep learning

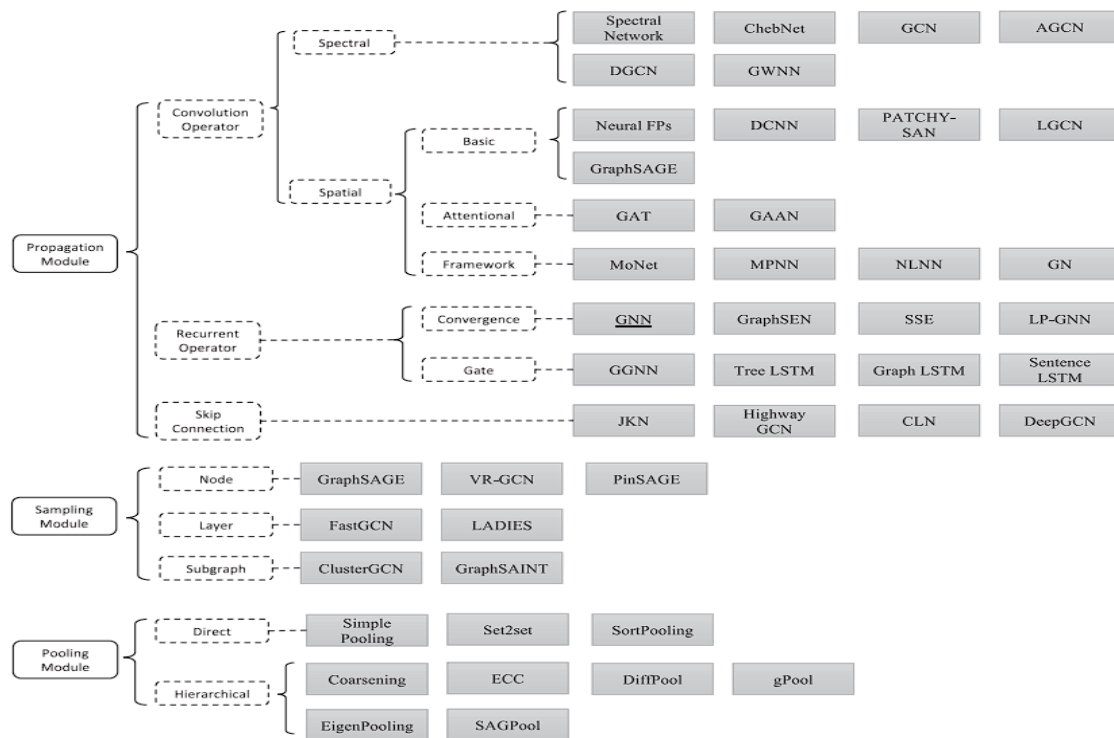
<sup>4</sup> Propagation Module

<sup>5</sup> Sampling Module

<sup>6</sup> Pooling Module

<sup>7</sup> Direct pooling

<sup>8</sup> Hierarchical pooling



شکل ۱: انواع شبکه‌های گرافی [۲]

از طرفی دیگر داده‌های ابرنقاط که با استفاده از سنسورهایی مانند سنسور لایدار<sup>۹</sup> به دست می‌آیند و نمایش سه بعدی از نقاط فضای اطراف هستند با ثبت دقیق و هندسی اشیاء و محیط‌های آن‌ها به ما این امکان را می‌دهند تا دقیق‌ترین تصاویر از محیط اطراف خود را تولید کنیم. این داده‌ها در زمینه‌های ژئوماتیک، مهندسی سه‌بعدی و بینایی کامپیوتر به کار می‌روند. داده‌های ابرنقاط در زمینه خودروهای خودران نیز کاربردهای فراوانی دارد. با استفاده از داده‌های ابرنقاط و پردازش بر روی آن‌ها می‌توان این داده‌ها را دسته‌بندی کرد که این موضوع در تشخیص موانع هنگام رانندگی خودروهای خودران بسیار کاربردی می‌باشد. دسته‌بندی داده‌های ابرنقاط به روش‌های مختلفی قابل انجام است. برای مثال یکی از این روش‌های دسته‌بندی با استفاده از معماری پوینت‌نت<sup>۱۰</sup> است که با استفاده از روش‌های یادگیری عمیق به دسته‌بندی داده‌های ابرنقاط می‌پردازد [۳].

در این پروژه، ما قصد داریم تا با استفاده از ترکیب معماری‌های تجمیع گراف خودتوجه<sup>۱۱</sup> و مدل پوینت‌نت، داده‌های مرتبط با خودروهای خودران را بهبود دهیم تا بتوان با دقت بالاتری این داده‌ها را دسته‌بندی کرده و به تجزیه و تحلیل دقیق‌تری از محیط‌های اطراف خودروهای خودران پردازیم. این پروژه به عنوان یک فرصت مناسب برای ترکیب دو زمینه مختلف از پردازش داده‌ها و مهندسی خودروها اهمیت ویژه‌ای دارد و می‌تواند به توسعه تکنولوژی خودروهای خودران و بهبود کیفیت حمل و نقل شهری کمک کند.

## ۲- مروری بر پروژه‌ها و سامانه‌های مشابه و بیان نقاط قوتی که با انجام این پروژه حاصل می‌شود:

معماری‌های استفاده شده در این پروژه عبارتند از دو معماری پوینت‌نت و تجمیع گراف خودتوجه که هر کدام را به صورت جداگانه بررسی کرده و نقاط قوت و ضعف هر یک را مرور می‌کنیم.

مدل عصبی پوینت‌نت مبتنی بر یادگیری عمیق است که برای دسته‌بندی داده‌های ابرنقاط در سال ۲۰۱۸ معرفی شده است. معماری این مدل در شکل ۲ مشخص است. این مدل با استفاده از یک معماری جدید و شبکه چندلایه پرسپترون<sup>۱۲</sup> توانایی دسته‌بندی داده‌های نامرتب<sup>۱۳</sup> ابرنقاط، تشخیص اشیاء و تشخیص صحنه را دارد. که ما در این پروژه از قابلیت دسته‌بندی آن استفاده خواهیم کرد.

<sup>۹</sup> Lidar

<sup>۱۰</sup> Point-Net

<sup>۱۱</sup> Self-attention graph pooling

<sup>۱۲</sup> Multi Layer Perceptron

<sup>۱۳</sup> Unorder

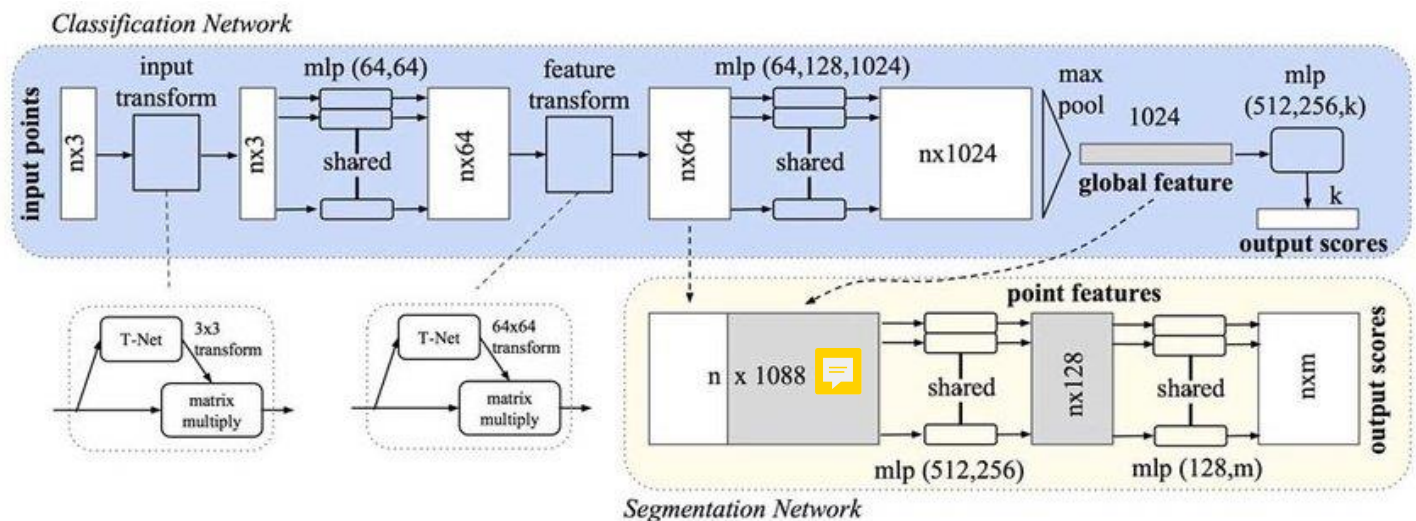
یکی از نقطه قوت‌های این معماری استفاده از یک شبکه جدا به اسم تی نت<sup>۱۴</sup> است که با استفاده از این شبکه و با تعریف یک تابع متقارن که در فرمول ۱ آمده است می‌تواند مشکل داده‌های ابرنقاط را که به صورت نامرتب هستند مرتب کرده و این مشکل را حل کند.

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n)) \quad (1)$$

در فرمول ۱ توابع به صورت  $f: 2^{\mathbb{R}^N} \rightarrow \mathbb{R}, h: \mathbb{R}^N \rightarrow \mathbb{R}^K, g: \mathbb{R}^K \times \dots \times \mathbb{R}^K \rightarrow \mathbb{R}^k$  تعریف می‌شوند.

ورودی این شبکه یک زیر مجموعه از فضای اقلیدسی است که ۳ ویژگی اصلی دارد. این ویژگی‌ها عبارتند از:

- ۱- نامرتب بودن: داده‌های ما نقطه‌های یک داده ابرنقاط هستند که بر خلاف تصاویر ترتیب مشخصی ندارند و می‌توان آن‌ها را با جایگشت‌های مختلف مرتب کرد.
  - ۲- تعامل بین نقاط: نقاط ما در فضایی با معیار مشخصی از فاصله هستند. این به این معناست که می‌توان برای تعیین همسایگی و تعامل بین نقاط از آن‌ها استفاده کرد.
  - ۳- تغییر ناپذیری تحت تبدیل: نقاط ما تحت تبدیل‌های هندسی همانند چرخش، دوران و یا ترکیب این دو نباید تغییر پذیر باشند و به هیچ وجه شکل و یا دسته‌بندی کلی آن‌ها نباید تغییر کند.
- داده‌های ما در ابتدا به عنوان ورودی به شبکه تی نت وارد می‌شوند و سپس ویژگی‌های آن‌ها و ساختار کلی نقاط با هم ترکیب شده و استخراج می‌شود.



شکل ۲: تصویری از معماری پوینت نت [۴]

در مرحله بعدی داده‌ها وارد یک شبکه چند لایه پرسپکترونی شده و دسته‌بندی می‌شوند. با استفاده از بخش تقسیم‌بندی معماری که در شکل ۲ وجود دارد نیز می‌توان هم ویژگی‌های محلی و هم ویژگی‌های سراسری داده‌های ابرنقاط را استخراج کرد و با استفاده از این ویژگی‌ها داده‌های ابرنقاط را دسته‌بندی کرد. از نقاط قوت این معماری می‌توان به عملکرد خوب آن در دسته‌بندی اشیاء پیچیده، پیچیدگی زمانی کم، نقاط گمشده و داده‌های نویز دار اشاره کرد [۴]. اما از نقاط ضعف این معماری نیز می‌توان به عملکرد ضعیف آن در تشخیص ساختارهای محلی، عدم توانایی در تشخیص الگوهای کوچک و نداشتن عملکرد مناسب در صحنه‌های پیچیده اشاره کرد [۵].

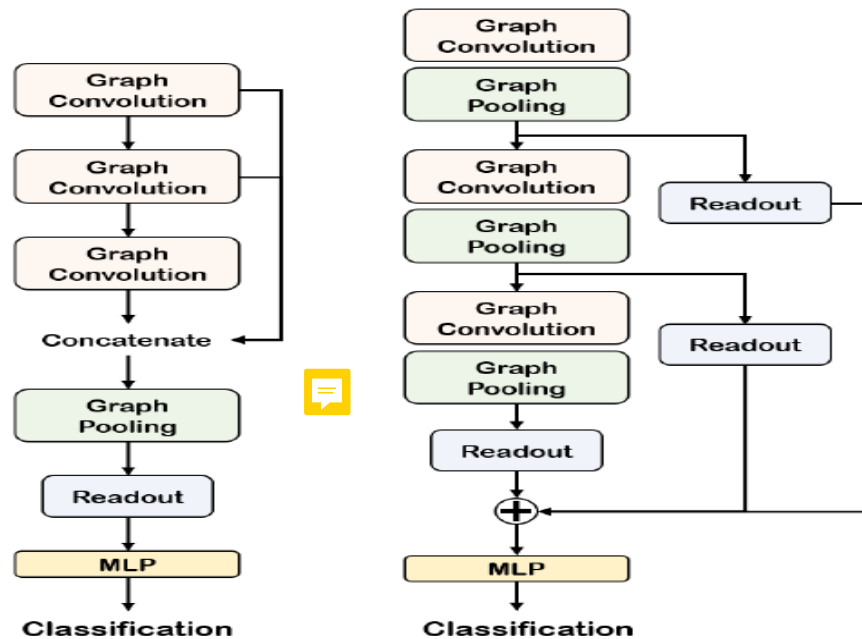
مدل دیگر مورد استفاده ما مدل گرافایی تجمیع گراف خودتوجه است که برای دسته‌بندی داده‌های گرافایی استفاده می‌شود. معماری این مدل در شکل ۳ آمده است. این معماری با استفاده از تکنیک‌های پیشرفته یادگیری عمیق و ترکیب شبکه‌های عصبی گرافایی و شبکه‌های عصبی پیچشی<sup>۱۵</sup> به دسته‌بندی داده‌ها می‌پردازد. این معماری با استفاده از شبکه‌های پیچشی گرافایی و محاسبات ماتریسی تمرکز بر روی ویژگی‌های مهم‌تر گراف دارد. دلیل استفاده از ساز و کار خودتوجهی توجه به توپولوژی گراف علاوه بر ویژگی‌های آن است. مجموع داده‌های استفاده‌شده در این معماری مجموع داده‌های یکسان با معماری‌های مشابه است تا بتوان نتایج این پژوهش را مقایسه کرد. هرکدام از این داده‌ها مناسب شبکه‌های عصبی گرافایی می‌باشد و هرکدام حداقل بالای ۱۰۰ گراف را در خود گنجانده‌اند. همانطور که در شکل ۳ نیز مشخص است دو معماری تجمیعی وجود دارد. معماری سمت چپ معماری تجمیعی سراسری<sup>۱۶</sup> و معماری سمت راست معماری تجمیعی سلسله‌مراتبی است. معماری سراسری عملکردی بهتری در داده‌های کوچکتر دارد و از دست دادن داده را به کمترین مقدار خود می‌رساند. درمقابل معماری سلسله‌مراتبی به علت اینکه ویژگی‌های اصلی داده‌ها را استخراج می‌کند عملکرد بهتری در داده‌های بزرگتر دارد. درکل هر دوی این معماری‌ها نتایج بهتری نسبت به بقیه مدل‌های

<sup>14</sup> T-net

<sup>15</sup> Convolutional neural networks

<sup>16</sup> Global pooling

تجمیعی دارند. همچنین این معماری نسبت به مدل‌های تجمیعی دیگر پیچیدگی زمانی کمتری نیز دارد [۶]. از نقاط ضعف این معماری می‌توان به این اشاره کرد که نرخ کاهشی<sup>۱۷</sup> گراف عدد متغیری نیست و هیچ بررسی بر روی آن نشده است. علاوه بر آن در قسمت‌هایی از آن مجموعه داده‌های مورد استفاده شفاف نیست و نحوه استفاده از داده‌ها نیز دارای مشکل است [۷].



شکل ۳: تصویر معماری تجمیع گراف خودتوجه [۶]

ایده اولیه این پروژه توسط یکی از دانشجویان دانشگاه صنعتی امیرکبیر در طول ترم بهار ۱۴۰۱-۱۴۰۲ به عنوان پروژه پایانی درس بینایی سب‌بعدی مطرح شده است [۸]. در ادامه سعی داریم که این ایده نوین را که برای دسته‌بندی داده‌های ابرنقاط و گرافی است را بهبود داده و تاثیر پارامترهای مختلف را در ترکیب دو مدل ارزیابی کنیم. مطالعه‌های زیادی در مورد ترکیب این دو موضوع انجام نشده است. نقطه قوتی که این پروژه نسبت به معماری‌های پوینت‌نت و گراف تجمیعی خودتوجه به صورت جداگانه دارد دسته‌بندی بهتر آن بر اساس نتایج اولیه مشاهده شده است. به طوری که هم با داده‌های ابرنقاط و هم با داده‌های گرافی می‌توان به این نتیجه رسید. در این پروژه که برای دسته‌بندی داده‌های ابرنقاط است ما علاوه بر مختصات سب‌بعدی هر نقطه ویژگی‌های دیگری مانند ویژگی‌های مرکزیت که مرتبط با خواص گره در گراف است نیز به داده‌های خود اضافه کرده‌ایم تا بتوانیم تاثیرگذاری ویژگی‌های گرافی بر روی داده‌های ورودی خود را بررسی کنیم.

### ۳- روش انجام پروژه (روش، نمودار بلوکی اجزای سامانه‌ی مورد نظر پروژه، ورودی‌ها و خروجی‌ها):

برای انجام این پروژه مجموعه داده مدل‌نت ده<sup>۱۸</sup> توسط دانشجو قبلی استفاده شده است. این مجموعه داده که بخشی از داده‌های مدل‌نت چهل است حاوی تقریباً پنج هزار شکل از اشیاء مختلف می‌باشد که در ده دسته مختلف قرار دارند. این مجموعه داده از نتایج جستجو در موتورهای جستجو به دست آمده و توسط افرادی در شرکت آمازون برچسب‌گذاری شده. پس از آن اشیایی که برچسب درستی نداشته‌اند حذف شده و یک مجموعه داده دقیق و کامل به دست آمده است [۹]. معماری پیشنهادی برای این پروژه را در شکل ۴ مشاهده می‌کنید. برای انجام این پروژه ما از قسمت دسته‌بندی معماری پوینت‌نت و از قسمت سلسله‌مراتبی معماری گرافی تجمیعی خودتوجه استفاده کرده‌ایم. ورودی ما داده‌های ابرنقاط خام هستند که پس از انجام پیش پردازش و مهندسی ویژگی به صورت ورودی به هر دو معماری داده می‌شود.

مراحل انجام پروژه به شرح زیر است که در ادامه هر یک را به صورت جداگانه شرح می‌دهیم:

#### ۱- بررسی معماری پوینت‌نت و گراف تجمیعی خودتوجه مورد استفاده:

در ابتدا ما نیاز به مطالعه هر دو معماری به صورت جداگانه و درک محاسبات آن‌ها و نحوه تاثیرگذاری پارامترهای مختلف در شبکه‌ها را داریم. تا بتوانیم در صورت نیاز هر کدام از این معیارها را متناسب با نیازهای خود بهینه کنیم. برای مثال هر داده ابرنقاط شامل مختصات نقاط در فضای اقلیدسی است. دانشجو قبلی علاوه بر این مختصات از هفت ویژگی دیگر که مرتبط با مرکزیت برای هر گره در گراف است استفاده کرده که این ویژگی‌ها شامل معیار کتز<sup>۱۹</sup>، بردارویژه، نزدیکی، میانگی، هم‌آوایی، رتبه بندی و بارمرکزیت است. این ویژگی‌ها باعث بالا رفتن تاثیر یک نقطه و اهمیت پیدا کردن روابط بین نقاط می‌شود.

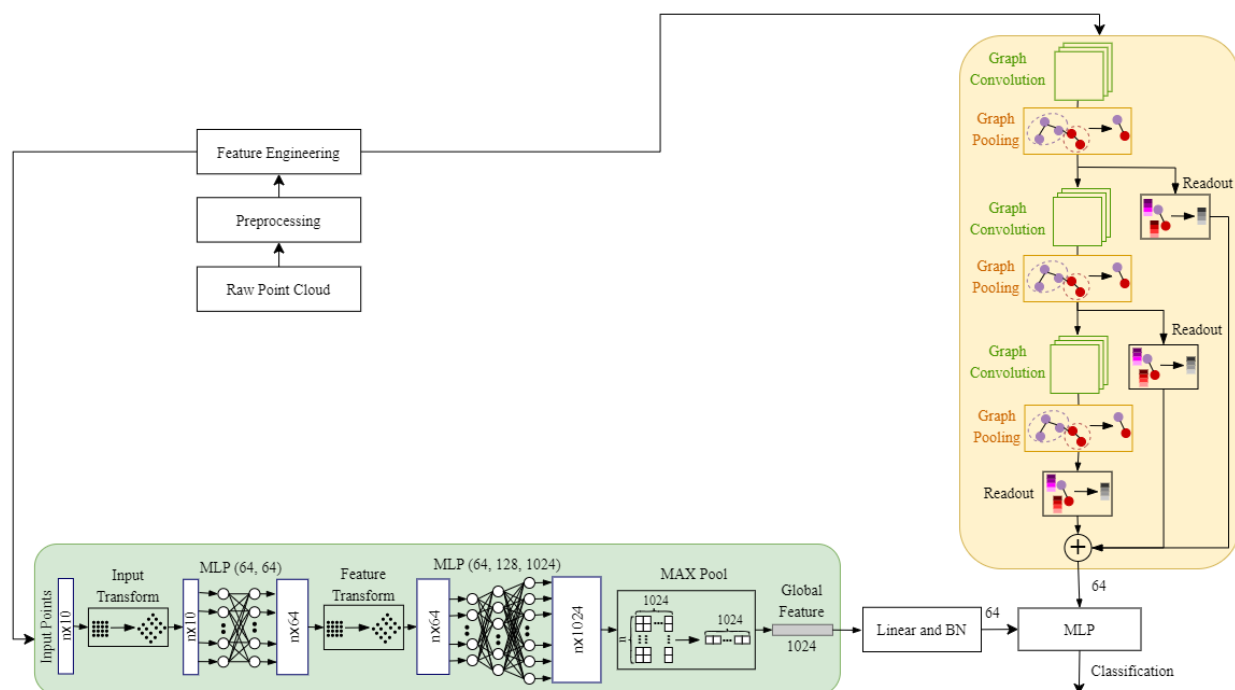
<sup>17</sup> Pooling ratio

<sup>18</sup> Model-Net10

<sup>19</sup> Katz



به نوعی نقاط غیر مرتبط را به هم مرتبط می‌کند. برای برقراری این نقاط نیز از روش نزدیک ترین همسایه استفاده شده است.



شکل ۴: معماری پیشنهادی پروژه

۲- تحلیل معماری معرفی شده توسط دانشجو قبلی:

پس از مطالعه و بررسی شبکه‌های قبلی در این مرحله ما باید معماری جدید معرفی شده را بررسی کرده و نحوه ورودی گرفتن و مهندسی ویژگی متفاوت آن را بررسی کنیم. علاوه بر این بررسی نحوه ترکیب خروجی دو معماری با هم و تغییراتی که در داده‌ها به وجود می‌آید برای اینکه یک نوع داده برای ورودی هر دو معماری مناسب باشد نیز حائز اهمیت است. پس از مطالعه و درک کدهای این قسمت به مرحله عملی می‌رسیم.

۳- آموزش مجدد مدل و تعیین پارامترهای مناسب:

در این مرحله با استفاده از مجموعه داده‌های قبلی استفاده شده مدل خود را مجدد آموزش می‌دهیم و با تغییر متغیرهای گوناگون مانند نرخ یادگیری، تعداد دوره<sup>۲۰</sup> و تعداد لایه‌ها سعی داریم که نتایج بهتری را بدست آوریم. همچنین با بازنویسی کدهای قبلی آن‌ها را به طوری مرتب می‌کنیم که بتوان به صورت عمومی از آن استفاده کرد. سپس با پیدا کردن یک مجموعه داده جدید از نوع گراف عملکرد مدل خود را در دسته‌بندی داده‌هایی از جنس غیر از ابرنقاط نیز بررسی می‌کنیم. انتظاری که ما پس از آموزش مجدد داریم این است که نتایج بهتری نسبت به دو مقاله پایه داشته باشیم و همچنین نسبت به نتایج اولی به دست آمده از پژوهش قبلی نیز نتایج بهتری داشته باشیم. این امکان وجود دارد که ما پس از انجام آموزش مجدد به این نتیجه برسیم که برخی معیارهای استفاده شده و یا ویژگی‌های اضافه شده به داده‌های اولیه مناسب نبوده و مجبور به انجام مجدد آن بشویم اما به طور کلی انتظار ما این است که نتایج بهتری نسبت به حالت‌های قبل در مجموع داده‌های اولیه مشابه بدست آوریم.

۴- ارزیابی تاثیر ویژگی‌های اضافه شده بر داده‌های ابرنقاط:

پس از آموزش مجدد مدل خود هفت ویژگی مرتبط با مرکزیت را که به داده‌های خود اضافه کرده‌ایم تغییر داده و تاثیر هر کدام را به صورت جداگانه در نتیجه دسته‌بندی و دقت مدل بررسی می‌کنیم. با استفاده از نتایج این بررسی می‌توان نتیجه گرفت که با استفاده از کدام یک از این ویژگی‌ها می‌توان ارتباط بهتری بین نقاط پیدا کرد. انتظار داریم که ویژگی‌های اضافه شده همگی تاثیر یکسانی بر روی پیشرفت مدل ما نداشته باشند و باید بررسی کرد که آیا این پیشرفت‌ها در صورت با هم بودن معیارهای مرکزیت به دست می‌آید و یا خیر؟ تخمین اولیه بنده در این جهت است که احتمالاً هر یک به صورت تنها لزوماً در پیشرفت مدل ما تاثیر ندارند.

۵- نمایش نقاط تاثیرگذار همراه با ویژگی‌های آن‌ها:

در نهایت با نتایج به دست آمده از مرحله قبلی و به دست آوردن مقدار گرادینت هر نقطه می‌توان نقاطی را که در دسته‌بندی ما تاثیر گذارتر بوده‌اند را پیدا کرده و یک ارزیابی جامع نسبت به اینکه کدام نقاط در اشیا تاثیر بیشتری در دسته‌بندی دارند انجام داد.

پس از رسم نمودار گرادینت و به دست آوردن نقاط تاثیرگذار می‌توانیم یک نقشه گرمایی برای تمامی داده‌های خود رسم کرده و نقاط پر اهمیت را به همراه

ویژگی‌های آن‌ها نمایش دهیم. چیزی که از خروجی کار ما تخمین زده می‌شود و انتظار داریم که نمایش داده شود این است که در دو معماری استفاده شده نقاط گوشه از اهمیت بالایی برخوردارند پس قطعاً نقاط حاشیه ما نقاطی هستند که تاثیرگذار خواهند بود همچنین با ویژگی‌های مرکزیت که به داده‌های خود اضافه کرده‌ایم می‌توان تخمین زد در نواحی که تجمع نقاط بیشتر است نیز تاثیرگذاری قابل توجهی داشته باشیم.

#### ۴- روش ارزیابی:

برای ارزیابی این پروژه ما با استفاده از دقتی که از مدل به دست آمده است ابتدا معماری جدید را نسبت به معماری‌های مشابه ارزیابی می‌کنیم. پس از آن مدل خود را نسبت به مدلی که دانشجو قبلی آموزش داده بود ارزیابی کرده و سپس هر یک از ویژگی‌های مرتبط با مرکزیت را که به داده‌های اولیه اضافه شده بود بررسی کرده و تاثیر گذارترین ویژگی را انتخاب می‌کنیم. در نهایت با نقشه گرمایی به دست آمده و رسم نمودار گرادینان داده‌های نقاط پراهمیت را پیدا کرده و با حذف کرده نقاط کم اهمیت تر درستی نتایج خود را تصدیق می‌کنیم. انتظاری کلی که با نسبت به روش خود داریم بهبود قابل توجهی در دقت آن نسبت به پژوهش‌های قبلی است به دلیل اینکه این موضوع در پژوهشی که دانشجو قبلی انجام داده بود بدون هیچ گونه تغییری در متغیرهای دو مدل پیشرفت داشته است حال با عوض کردن آن‌ها قطعاً دقت بهتری می‌توان به دست آورد.

#### ۵- مراحل انجام و زمان‌بندی پروژه:

هدف	مرداد	شهریور	مهر	آبان	آذر	دی	بهمن	اسفند	فروردین
بررسی معماری پوینتنت و گرافایی تجمیعی خودتوجه									
بررسی معماری‌های استفاده شده									
بررسی تاثیر پارامترهای استفاده شده									
پیدا کردن پارامترهای مناسب برای مدل									
آموزش مدل با استفاده از پارامترهای جدید									
پیدا کردن پارامترهای تاثیر گذار									
نمایش نقاط مهم و تاثیر گذاری آن‌ها در خروجی									
نگارش پایان نامه									

#### ۶- امکانات لازم (ابزارها، محیط‌ها، و نرم‌افزارهای مورد استفاده):

- یک عدد GPU حداقل ۳۰۹۰ برای آموزش مجدد مدل

#### ۷- مراجع و منابع:



- [1]J. Adrian Bondy and U. S. R. Murty, *Graph theory with applications*. New York ; Chichester: Wiley, 2002.
- [2]J. Zhou *et al.*, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020, doi: <https://doi.org/10.1016/j.aiopen.2021.01.001>.
- [3]D. Fernandes *et al.*, “Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy,” *Information Fusion*, vol. 68, pp. 161–191, Apr. 2021, doi: <https://doi.org/10.1016/j.inffus.2020.11.002>.
- [4]R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, Jul. 2017, doi: <https://doi.org/10.1109/cvpr.2017.16>.
- [5]C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” *Neural Information Processing Systems*, vol. 30, pp. 5099–5108, Jun. 2017.
- [6]J. Lee, I. Lee, and J. Kang, “Self-attention Graph Pooling,” *International Conference on Machine Learning*, pp. 3734–3743, Apr. 2019.
- [7]C. Holtz, “Issues · inyeople77/SAGPool,” *GitHub*, Sep. 11, 2019. <https://github.com/inyeople77/SAGPool/issues> (accessed Aug. 26, 2023).
- [8]M. Ebadpour, “PointNet Meets Self-Attention Graph Pooling: a Synergistic Approach to Point Cloud Classification,” *GitHub*, Aug. 21, 2023. <https://github.com/MohsenEbadpour/PointNet-meets-Self-Attention-Graph-Pooling-A-Synergistic-Approach-to-Point-Cloud-Classification> (accessed Aug. 25, 2023).
- [9]Z. Wu *et al.*, “3D ShapeNets: a Deep Representation for Volumetric Shapes,” *IEEE Xplore*, pp. 1912–1920, Jun. 2015, doi: <https://doi.org/10.1109/CVPR.2015.7298801>.
- [10]M. G. Seenappa, “Graph Classification Using Machine Learning Algorithms,” *SJSU ScholarWorks*, May 2019, doi: <https://doi.org/10.31979/etd.b9pm-wpng>.
- [11]P. Wang, T. Gu, B. Sun, D. Huang, and K. Sun, “Research on 3D Point Cloud Data Preprocessing and Clustering Algorithm of Obstacles for Intelligent Vehicle,” *World Electric Vehicle Journal*, vol. 13, no. 7, p. 130, Jul. 2022, doi: <https://doi.org/10.3390/wevj13070130>.
- [12]I. Lang, A. Manor, and Shai Avidan, “SampleNet: Differentiable Point Cloud Sampling,” *ArXiv (Cornell University)*, Dec. 2019.