استفاده از شبکه های عصبی برای تولید شبکه محاسباتی بر پایه داده های ابر نقاط: یک مطالعه مروری

نگار احمدزاده [*۱]، علیرضا جهانگیریان [۲]

۱- دانشجو ارشد، مهندسی هوافضا، دانشگاه صنعتی امیرکبیر، تهران، negar.ahmadzadeh@aut.ac.ir

۲- استاد، مهندسی هوافضا، دانشگاه صنعتی امیرکبیر، تهران، ajahan@aut.ac.ir

خلاصه

این مقاله مروری بر مطالعات انجام شده در زمینه استفاده از شبکه های عصبی برای تولید شبکه محاسباتی بر پایه داده های ابر نقاط [†]می باشد. تقسیم بندی مناسبی با هدف بررسی عملکرد هر روش صورت گرفته است. سپس هر روش مورد مطالعه قرار گرفته و ویژگی های کلیدی آن به همراه شیوه آموزش به شبکه عصبی ذکر شده است. هر یک از این روش‌ها دارای مزایا و چالش‌هایی مانند بهبود دقت در پردازش داده‌های نامنظم، پیچیدگی محاسباتی، و مدیریت نویز در داده‌های ورودی هستند. این مقاله همچنین نظراتی که باعث بهبود پیاده سازی الگوریتم می شوند را نیز ارائه داده است.

کلمات کلیدی: ابر نقاط, شبکه عصبی, شبکه محاسباتی

۱. مقدمه

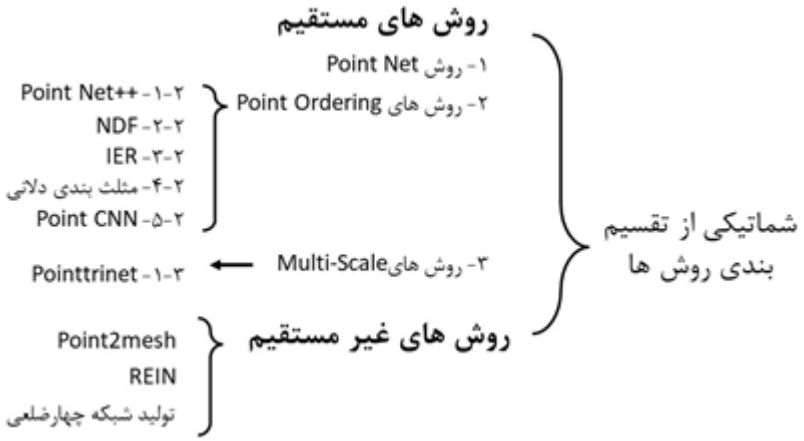
داده‌های ابر نقاط مجموعه‌ای از نقاط در فضای سه‌بعدی هستند که برای نمایش اشیا یا محیط‌های واقعی به‌کار می‌روند. هر نقطه در ابر نقاط دارای سه مولفه (x, y, z) است و ممکن است ویژگی‌های دیگری نظیر رنگ یا بردار عمود بر سطح یا ویژگی های هندسی را نیز در بربگیرد. این داده‌ها معمولاً از طریق اسکنرهای سه‌بعدی، حسگرهای لیدار[‡]، و دیگر فناوری‌های تصویربرداری تولید می شوند و به دلیل قابلیت بازسازی دقیق جزئیات هندسی، کاربرد وسیعی در زمینه‌هایی چون واقعیت مجازی[۲][۱]، مهندسی معکوس[۴][۳]، نقشه‌برداری[۶][۵]، و شبیه‌سازی دارند.

با این حال، کار با داده‌های ابر نقاط به دلیل ماهیت نامنظم و پراکنده بودن آن‌ها چالش‌های بسیاری را بوجود آورده است. این داده‌ها معمولاً فاقد ساختارهای منظم هستند و تراکم نقاط در بخش‌های مختلف فضا ممکن است متفاوت باشد، که منجر به پیچیدگی پردازش و تحلیل آن‌ها شده است. علاوه بر این، داده‌های ابر نقاط اغلب حاوی نویز و نقاط ناقصی هستند

[†] **Point Clouds**
[‡] **Lidar**

که بر دقت بازسازی و تحلیل آنها می‌گذارد. به منظور مواجهه با این چالش‌ها، روش‌های متعددی از هوش مصنوعی و به‌ویژه یادگیری عمیق طی سال‌های اخیر توسعه یافته است که قابلیت‌های قابل‌توجهی در پردازش داده‌های ابر نقاط را به جهانیان نشان داده است.

بطور کلی روش هایی که در این مقاله مورد بررسی قرار گرفته اند به دو دسته کلی روش های مستقیم *و روش های غیر مستقیم †تقسیم شده اند. در روش ها غیر مستقیم، داده های ابر نقاط به ساختار منظمی تبدیل شده و شبکه عصبی ایی که بر آن پیاده شده می تواند آن را پردازش کند. دسته روش های غیر مستقیم بدن اعمال هیچ فیلتری، داده های نامنظم ابر نقاط را برای تولید سبکه استفاده می کند. این دسته به دو نوع point ordering و multi-scale تقسیم بنده می شود. دسته بندی اول از ابر نقاط به گونه ایی استفاده می کند که به مشکل بی نظمی بر نخورد و ویژگی های کاربردی را بتواند شناسایی کند. دسته multi-scale شبکه را به زیر مجموعه هایی تقسیم کرده و پردازش اطلاعات را در هر زیر مجموعه انجام می دهد. در ادامه به معرفی الگوریتم ها و بررسی ویژگی انها که در تصویر شماره (۱) به آنها اشاره شده است می پردازیم.



شکل شماره ۱- شماتیکی از روش هایی که در این مقاله به مطالعه آنها پرداخته شده است.

۲. معرفی الگوریتم ها

هر الگوریتم شامل معرفی کلی از روند اجرا، چالش های آموزش به شبکه ی هوش مصنوعی، ویژگی های روش و چگونگی برخورد با ابر نقاط و دسته بندی نقاط می شود. سپس در نهایت به بررسی مزایا و معایب هر روش پرداخته شده و پیشنهادات قابل توجه برای بهبود عملکرد شبکه عصبی مورد ذکر واقع شده است.

۱.۲. روش های مستقیم

در روش های مستقیم، بی نظمی و پراکندگی غیر یکنواخت نقاط مورد توجه نیست و می توانیم از این موضوع به راحتی گذر کنیم. همچنین این دسته روش قادر است از این ویژگی ابر نقاط به عنوان پتانسیلی برای استفاده در بررسی ویژگی نقاط استفاده کرد.

* **Direct methods**
† **Indirect methods**

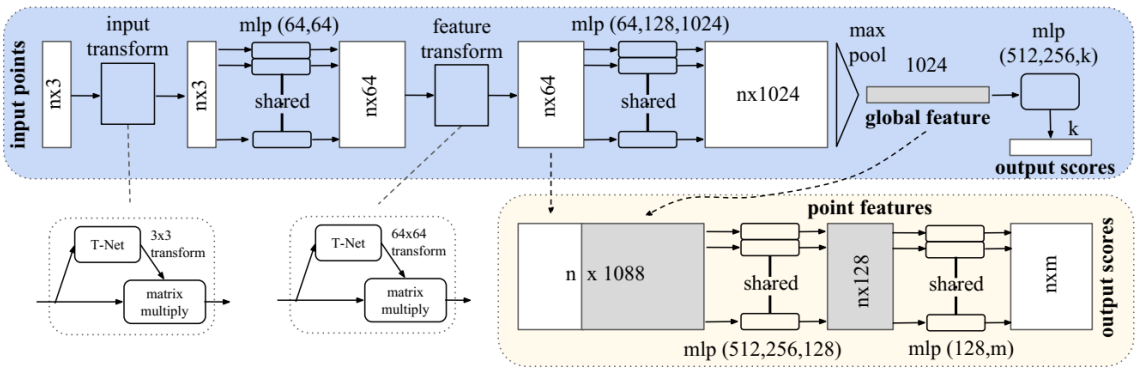
- روش Point net

اولین مقاله ارائه شده در زمینه استفاده از هوش مصنوعی و معرفی الگوریتمی که بتواند با استفاده از ابر نقاط اقدام به تولید شبکه داشته کند، در سال ۲۰۱۷ توسط چارلز و همکاران[۷] به نام Pointnet ارائه شد. این الگوریتم پایه گذار و مشوق تولید دیگر روش های مبتنی بر هوش مصنوعی برای تولید شبکه محاسباتی بوده است. در این الگوریتم با استفاده از شبکه عصبی به‌طور مستقیم از مختصات سه‌بعدی نقاط استفاده شده است. بطوریکه نیازی به تبدیل داده‌ها به شبکه‌های منظم نداریم. ابتدا هر نقطه از ابر نقاط به‌صورت جداگانه پردازش شده و ویژگی‌های آن از طریق یک شبکه عصبی چندلایه (MLP) استخراج می‌شود. سپس یک بردار حاوی ویژگی برای هر نقطه ساخته شده که منحصر بفرد است. در فرمول زیر می توان این تبدیل را با استفاده از توابع f و g متوجه شد:

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

در این رابطه h ،x_i و g به ترتیب بیانگر مختصات هر نقطه، شبکه عصبی چند لایه برای استخراج ویژگی های هر نقطه و تابع Max Pooling برای ترکیب ویژگی های کل نقاط می باشند. در نهایت ویژگی تمام نقاط با هم ترکیب می‌شود تا یک ویژگی کلی برای کل ابر نقاط به دست آید. این ویژگی کلی نمایانگر خصوصیات ابر نقاط است و می‌تواند برای طبقه‌بندی یا تقسیم‌بندی مورد استفاده قرار گیرد.

برای جلوگیری از انجام تبدیلات هندسی مانند دوران، انتقال یا مقیاس‌دهی، الگوریتم PointNet از یک شبکه عصبی کوچک به نام T-Net استفاده می‌کند. این شبکه قادر است تا یک ماتریس تبدیل * را برای هر ابر نقاط پیش‌بینی کند. سپس این ماتریس به مختصات نقاط اعمال شده تا داده‌ها به شکل استاندارد تبدیل شوند. این عملیات باعث می‌شود که شبکه نسبت به چرخش یا جابجایی داده‌ها حساس نباشد. از این روش به دلیل سادگی و کارایی، می توان برای تحلیل داده‌های سه‌بعدی به‌طور مستقیم استفاده کرد. این عملکرد را می توان در شکل شماره ۲ مشاهده نمود.



شکل شماره ۲-دستور عملکرد الگوریتم Point Net [7]

از مزایای این روش می‌توان به پردازش مستقیم داده‌های نامنظم، عدم نیاز تبدیل به شبکه‌های منظم و مقاومت در برابر تغییرات هندسی اشاره کرد. اما از معایب آن می‌توان به حساسیت به داده‌های با تراکم غیریکسان و چالش در درک ساختارهای پیچیده محلی نام برد.

برای بهبود این روش در آینده، می‌توان از تکنیک‌های پیشرفته‌تری برای افزایش دقت در نقاط با جزئیات محلی استفاده کرد و بهینه‌سازی‌های معماری شبکه را برای کاهش پیچیدگی محاسباتی انجام داد.

*Transformation Matrix

- **روش های Multi-scale**

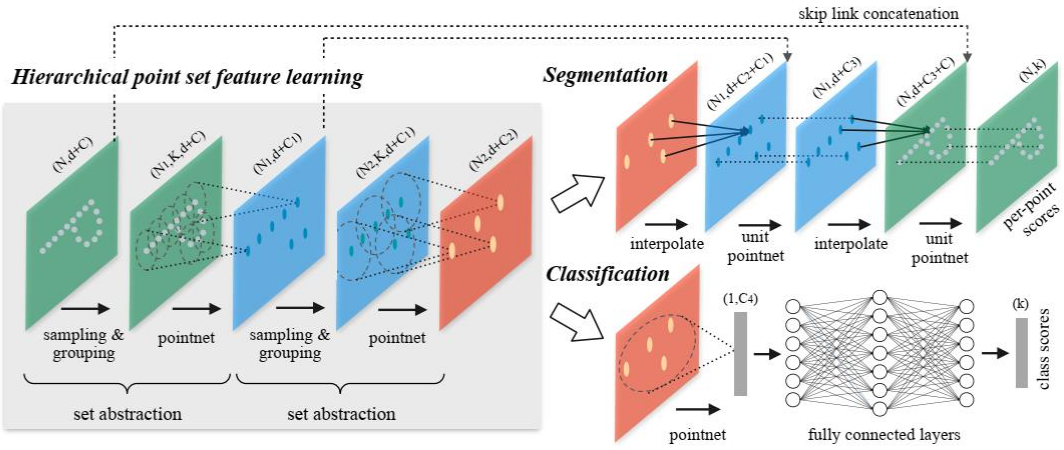
در این تقسیم بندی، میدان به مقیاس های کوچکتری تقسیم شده و شبکه محاسباتی در هر زیر میدان ایجاد می شود. سپس میدان ها با هم ترکیب شده و پردازش می شوند تا شبکه نهایی شکل بگیرد. در ادامه به بررسی روش هایی که از این تکنیک بهره برده اند، پرداخته شده است.

- **روش Point Net ++**

این روش نسخه بروز شده ایی از روش قبلی می باشد که در همان سال توسط چارلز و همکاران[۸] نیز ارائه گردیده است. این الگوریتم یک روش پیشرفته برای پردازش داده‌های سه‌بعدی برای ابر نقاط می باشد که چالش‌های مرتبط با تراکم نامنظم نقاط را برطرف می‌کند. این نقاط، به دلیل فقدان ترتیب مشخص و تراکم نامنظم، پردازش پیچیده‌ای دارند. الگوریتم Point Net ++ با بهره‌گیری از یک ساختار سلسله‌مراتبی و استفاده از تکنیک‌هایی نظیر نمونه‌برداری از دورترین نقاط و گروه‌بندی چندمقیاسی ، قادر به استخراج ویژگی‌های چندگانه از این داده‌ها است.

این الگوریتم، ابتدا نقاطی از مجموعه اصلی را به عنوان نقاط مرجع انتخاب می‌کند. بطوریکه باعث می‌شود نقاط انتخاب‌شده به‌طور یکنواخت کل فضا را پوشش دهند. نقاط مرجع انتخاب‌شده، به عنوان مرکز نواحی محلی مورد استفاده قرار می‌گیرند. برای هر نقطه مرجع، نقاط نزدیک به آن (بر اساس شعاع تعریف‌شده) به‌عنوان نقاط همسایه گروه‌بندی می‌شوند. این گروه‌بندی‌ها به الگوریتم کمک می‌کنند تا ویژگی‌های محلی ناحیه را استخراج کند. برای هر ناحیه محلی که در مرحله قبل تعریف شده است، یک نسخه کوچک از شبکه PointNet اجرا می‌شود. این شبکه عصبی ویژگی‌های محلی هر ناحیه را استخراج کرده و آن‌ها را به بردارهای ویژگی تبدیل می‌کند.

Point Net ++ از روش گروه‌بندی در مقیاس‌های مختلف استفاده می‌کند تا بتواند ویژگی‌های هندسی را در اندازه‌ها و مقیاس‌های متفاوت استخراج کند. این ویژگی به شبکه کمک می‌کند تا تغییرات تراکم نقاط را مدیریت کند. شکل شماره 3 ترتیب پیاده سازی این الگوریتم را نشان می دهد.



شکل شماره ۳- دستور عملکرد الگوریتم Point Net ++ [8]

یکی از معادلات مهم در PointNet++ به صورت زیر تعریف می‌شود که نشان‌دهنده تابع مجموعه‌ای است که نقاط را به یک بردار نگاشت می‌کند:

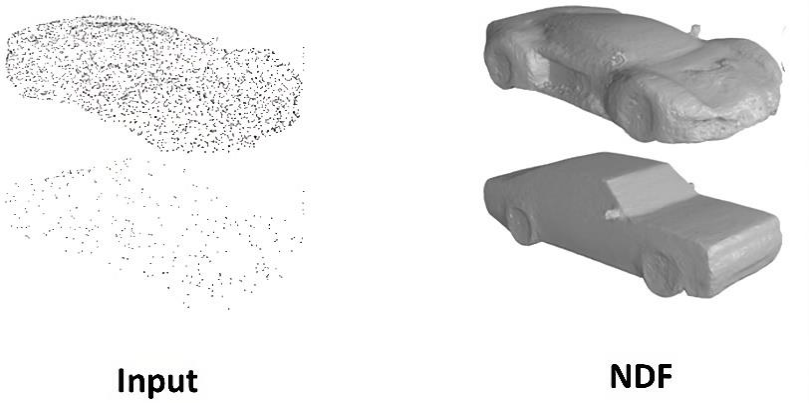
$$f(x_1, x_2, \cdots, x_n) = \gamma(\max_{i=1,\cdots,n}\{h(x_i)\}) \quad (2)$$

در این فرمول x_i ، h، max و gamma به ترتیب نشان دهنده مختصات هر نقطه، شبکه عصبی چند لایه برای استخراج ویژگی های هر نقطه، عملیاتی برای فشرده سازی ویژگی های نقاط برای از بین بردن ترتیب و تابعی نهایی باری تبدیل ویژگی های فشرده شده به خروجی مطلوب می باشند.

دقت بالا در داده های نامنظم، انعطاف پذیری در تراکم بالا نقاط از بارز ترین مقاط قوت این الگوریتم به حساب می آید. اما پیچیدگی محاسباتی زیاد و بروز حساسیت به انتخاب بردار ویژگی های باعث توجه به سایر های هایی شده است که در ادامه مورد بررسی قرار خواهند گرفت.

- روش NDF*

از دیگر روش های جدید و نوین برای بازسازی سطح توسط ابر پراکنده ایی از نقاط با استفاده از هوش مصنوعی می توان به روشNDF [9] اشاره کرد. در این روش ابر نقاط به عنوان ورودی به الگوریتم داده شده، سپس یک شبکه عصبی فاصله هر نفطه تا نزدکترین سطح را محاسبه می کند. به این عمل یادگیری فیلد فاصله گفته می شود که کلید اصلی برای تولید شبکه در این الگوریتم به حساب می آید. این شبکه عصبی برای یادگیری فیلد فاصله از داده‌های ورودی استفاده می کند. مختصات نقاط (x, y, z) را به همراه ویژگی‌های یادگیری‌شده از ابر نقاط ورودی دریافت شده و فاصله آن نقطه تا نزدیک‌ترین سطح محاسبه می شود. سپس از یک شبکه عصبی پیچشی سه بعدی برای نگاشت ابر نقاط به فیلد فاصله استفاده می شود تا در حد امکان، سطح را بطور پیوسته بازسازی کند.این عملیات در شکل شماره ۴ قابل مشاهده است. برای آموزش این شبکه عصبی، مجموعه‌ای از داده‌های ابر نقاط و سطوح واقعی به مدل داده می‌شود. در طول آموزش، شبکه سعی می‌کند فاصله‌ای که برای هر نقطه پیش‌بینی کرده است را با فاصله واقعی که از داده‌های به دست آمده مقایسه کند. سپس برای ارزیابی مدل از یک تابع هزینه استفاده می‌شود که اختلاف بین فاصله پیش‌بینی‌شده توسط مدل و فاصله واقعی را اندازه‌گیری می‌کند. این تابع معمولاً بر اساس فاصله اقلیدسی یا دیگر متریک‌های فاصله تعریف می‌شود.



شکل شماره ۴- نمونه ایی از دریافت ابر نقاط و تبدیل آن به سطوح پیوسته در الگوریتم NDF [9]

پس از محاسبه فیلد فاصله برای نقاط مختلف در فضا، سطوح با استفاده از روش‌های خاصی بازسازی می‌شوند. برای استخراج دقیق سطح از فیلد فاصله، نقاط روی سطح شیء پیش‌بینی شده و سپس از این نقاط برای تولید یک مش سه‌بعدی استفاده می‌شود. این کار معمولاً با استفاده از الگوریتم‌هایی مانند Marching Cubes یا دیگر روش‌های تبدیل ابر نقاط

* **Neural Unsigned Distance Fields**

به مش انجام می‌شود که نقاط پیش‌بینی‌شده را به هم متصل می‌کنند تا سطح شیء را تشکیل دهند. الگوریتم Marching Cubes با استفاده از مقادیر فاصله در یک شبکه مکعبی، سطح شیء را استخراج می‌کند. البته می توان از روش مثلث بندی دلانی نیز برای ایجاد مش استفاده نمود.

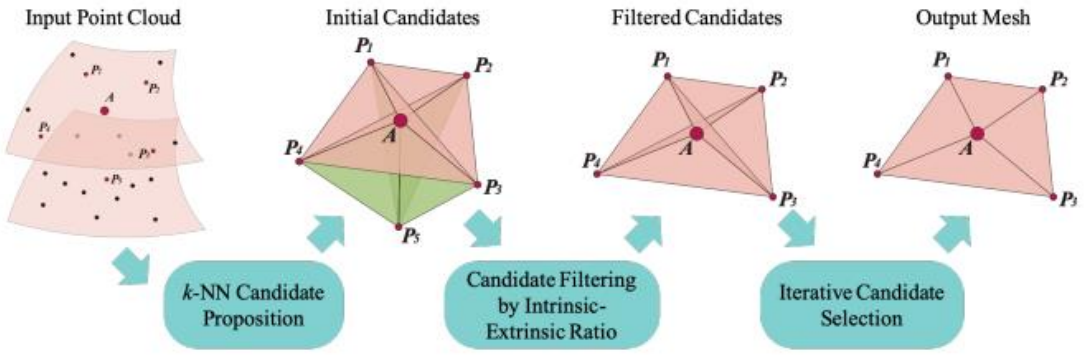
بازسازی دقیق سطوح پیچیده و باز، ایجاد و حفظ پیوستگی سطح و انعطاف پذیری بالا در مقابل پردازش داده های پراکنده و دارای نویز از فواید استفاده از این روش می باشد. در عین حال، پیچیدگی محاسباتی این الگوریتم و زمان پردازش بالا از نقاط ضعف این روش به حساب آمده که با استفاده از روش های بهینه سازی قابل بهبود می باشد.

- روش IER*

یکی از بهترین ایده های برای تولید شبکه در روش IER معرفی شده است. این روش که توسط مینقاو و همکاران[۱۰] ارائه شده است قادر است تا از یک شبکه عصبی برای فیلتر کردن مثلث های نادرست استفاده کند. این الگوریتم ابتدا اقدام به تشکیل گراف K-nn شامل نزدیکترین همسایه های هر نقطه می کند. هدف این است که مجموعه‌ای از مثلث‌های کاندیدا ایجاد شود که کل سطح را پوشش دهند. در نهایت، برای هر نقطه تعداد زیادی مثلث کاندیدا ساخته می‌شود. برخی از این مثلث‌ها به‌درستی سطح جسم را پوشش می‌دهند و برخی دیگر نادرست هستند (مانند مثلث‌هایی که نقاطی از سطوح مختلف یا نزدیک به هم اما مستقل را به هم متصل می‌کنند). سپس دسته‌ای از مثلث‌های نادرست در مناطقی مانند ساختارهای نازک، سطوح نزدیک به هم یا بخش‌های فضایی مجاور (که مستقل از هم هستند) ظاهر می شوند. این مرحله پایه‌ای و بسیار مهم است، زیرا مثلث‌های کاندیدا از روی ابر نقاط ساخته می‌شوند و این مثلث‌ها مشخص می‌کنند که کدام نقاط به هم متصل هستند و می‌توانند به‌طور دقیق سطح جسم را تشکیل دهند. اکنون با استفاده از معیار IER مثلث های نادرست که نماینده سطح واقعی نیستند را شناسایی و حذف خواند شد.

برای یک مثلث با سه رأس u، v و w، IER به شکل زیر گسترش می‌شود:

$$\mathrm{IER}(u,v,w) = \frac{d_G(u,v) + d_G(u,w) + d_G(v,w)}{d_E(u,v) + d_E(u,w) + d_E(v,w)} \qquad (۳)$$



شکل شماره ۵- تصویری از نحوه پیاده سازی الگوریتم IER [10]

این فرمول مجموع فاصله‌های ژئودزیک و اقلیدسی بین هر جفت رأس‌های مثلث را در نظر می‌گیرد. پس از محاسبه معیار IERبرای هر جفت نقطه در هر مثلث، مثلث‌هایی که IER آن‌ها بزرگ‌تر از یک مقدار آستانه (τ) است، به‌عنوان مثلث‌های نادرست فیلتر می‌شوند. این آستانه معمولاً مقدار مشخصی بزرگ‌تر از ۱ است که تعیین می‌کند که چه مثلث‌هایی باید حذف

* **Intrinsic-Extrinsic Ratio**

شوند. مثلث‌هایی کهIER آن‌ها بزرگ‌تر از این آستانه هستند، به دلیل اینکه نقاطشان به سطوح متفاوت یا بخش‌های غیرمرتبط تعلق دارند، حذف می‌شوند. این مثلث‌ها معمولاً در مناطقی با ساختارهای نازک، سطوح نزدیک به هم، یا بخش‌های مجاور که نباید به هم متصل شوند، ظاهر می‌شوند.

پس از فیلتر کردن مثلث‌های نادرست با استفاده از نسبت IER در مرحله قبلی، مثلث‌های باقی‌مانده (که به احتمال زیاد سطح واقعی جسم را پوشش می دهند) باید بر اساس برخی معیارها مرتب شوند تا در نهایت بتوان مش نهایی تکمیل کرد. دو معیار اصلی برای مرتب‌سازی مثلث‌ها وجود دارد:

- فاصله تا سطح واقعی

مثلث‌هایی که به سطح مرجع نزدیک‌تر هستند، اولویت بالاتری دارند. این معیار کمک می‌کند مش نهایی زودتر تشکیل شود.

- طول لبه‌های مثلث

مثلث‌هایی که لبه‌های کوتاه‌تری دارند، در اولویت قرار میگیرند. مثلث‌های با لبه‌های کوتاه معمولاً نماینده‌ی خوبی برای بخش‌های یکنواخت سطح هستند و به جلوگیری از ایجاد مش‌های نامنظم کمک می‌کنند.

پس از مرتب‌سازی، مثلث‌ها به‌صورت تدریجی به مش نهایی اضافه می‌شوند. اما قبل از اضافه کردن هر مثلث، دو شرط مهم دیگر باید بررسی شوند:

- عدم تداخل مثلث ها

مثلث جدید نباید با مثلث‌هایی که قبلاً به مش اضافه شده‌اند، تداخل داشته باشد. یعنی هیچ‌یک از لبه‌های مثلث نباید با لبه‌های مثلث‌های قبلی تقاطع داشته باشد.

- لبه‌های مانفولد

هر لبه در مش نهایی باید به حداکثر دو مثلث متصل باشد. اگر اضافه کردن مثلث جدید باعث شود که یک لبه بیش از دو مثلث داشته باشد (یعنی لبه غیرمانفولد شود)، آن مثلث پذیرفته نمی‌شود. این فرایند مرتب‌سازی و اضافه‌کردن مثلث‌ها به‌طور مداوم تکرار می‌شود تا تمام مثلث‌های صحیح به مش نهایی اضافه گردند. در نهایت، تمام مثلث‌های صحیح انتخاب شده و مش نهایی ساخته می‌شود که نماینده‌ای دقیق از سطح واقعی جسم است.

نقاط قوت این روش شامل دقت بالا، حفظ جزئیات و قابلیت تطبیق با داده های غیر یکنواخت است. در حالیکه حساسیت به تنظیمات از دلایل اصلی ایجاد ضعف در این روش می باشد.

- روش مثلث بندی دلانی* بر پایه شبکه عصبی

روش های ساده و اکادمیک تولید مش همیشه مورد توجه پژوهش گران جوان واقع شده است. به همین دلیل ماری و همکاران [11] مطالعات خود را بر روی پیاده سازی مثلث بندی دلانی با استفاده از هوش مصنوعی را در سال 2021 شروع کردند و موفق شدند الگوریتم مناسبی را ارائه دهند. روشی که آنها معرفی کردند ابر نقاط را به عنوان ورودی دریافت کرده و همسایگی ژئودزیک هر نقطه را محاسبه می کند. این کار با انتخاب نزدیک ترین همسایه های هر نقطه آغاز می شود. سپس فاصله اقلیدسی محاسبه گردیده و با اعمال یک شبکه عصبی می توانیم همسایه هایی که روی سطح جسم واقع شده است را شناسایی کنیم. به این کار شناسایی همسایگی ژئودزیک می گویند. خروجی این شبکه عصبی به شکل عددی بین ۰ و ۱ برای هر همسایه است. این عدد بیان می کند هر همسایه چقدر احتمال دارد که یکی از نزدیک ترین همسایه های ژئودزیک باشد. این مرحله پایه‌ای برای مراحل بعدی است، زیرا شناسایی دقیق همسایگی‌های ژئودزیک محلی باعث می‌شود که در

* **Delaunay Triangulation**

مراحل بعدی، مثلث‌بندی دلاونی و نگاشت‌های لگاریتمی دقیق‌تر انجام شود و مش نهایی کیفیت بالاتری داشته باشد. در مرحله بعد، نگاشتی لگاریتمی *از تکه های ۳ بعدی همسایگی های ژئودزیک به یک فضای ۲ بعدی صورت میپذیرد. بدین گونه که روشی ریاضی برای تبدیل یک سطح منحنی به یک فضای ۲ بعدی پیاده سازی می شود. برای بهبود دقت نگاشت لگاریتمی، دو نوع خطا محاسبه می گردد:

- خطای مختصات دوبعدی

تفاوت بین مختصات نگاشت پیش‌بینی‌شده توسط شبکه و نگاشت صحیح برای هر نقطه محاسبه و مینیمم می‌شود.

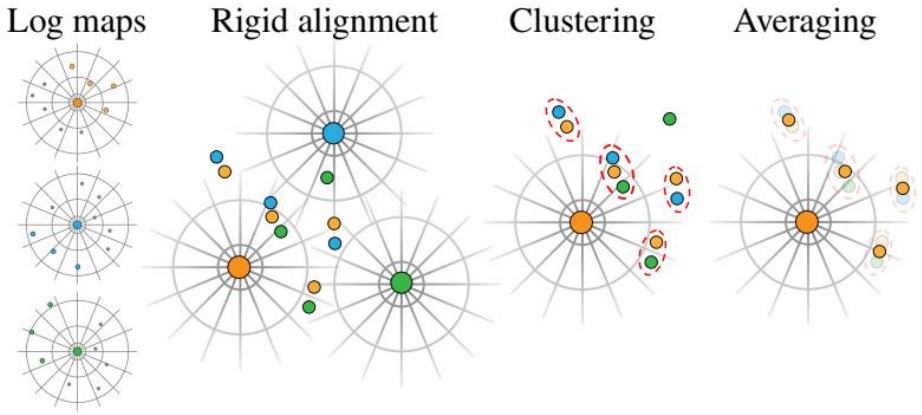
- خطای فاصله ژئودزیک

تفاوت فاصله ژئودزیک واقعی هر نقطه تا مرکز با فاصله پیش‌بینی‌شده در نگاشت دوبعدی محاسبه و به حداقل می‌رسد.

پس از اعمال تطبیق، مجموعه مختصات دوبعدی نقاط مشترک بین تکه‌های همسایه ایجاد شده که باید آنها را در یک گروه قرار دهیم تا از ایجاد نویز جلوگیری کنیم. به این مرحله خوشه بندی می گویند. در نهایت از نقاط هر خوشه یک مقدار میانگین گرفته شده تا بهترین تحمین برای معرفی نقطه مشترک معرفی گردد. در شکل شماره (۵) زیر می توان خلاصه ایی از روند الگوریتم تا به اینجا را مشاهده کرد.

مثلث بندی دلانی برای نقاط هر گروه اعمال می گردد. ویژگی مثبت مثلث بندی دلانی، تضمین با کیفیت بودن مثلث ها در فضای دو بعدی است. نتیجه این کار یک مجموعه مثلث‌های دلانی برای هر تکه می باشد که به آنها عناصر سطح دلاونی (DSE) گفته می‌شود. هر سطح دلانی، مجموعه ایی از مثلث های دلانی بوده که حول نقطه مرکزی برای هر خوشه ساخته شده است. اما وجود بعضی مثلث ها در بیش از یک سطح دلانی کار را مشکل ساز می کند. یعنی، مثلث‌های مربوط به نقاط مشترک بین تکه‌های همسایه ممکن است چندین بار ظاهر شوند. به همین دلیل مثلث ها را به سه دسته عمده زیر تقسیم بندی می کنند:

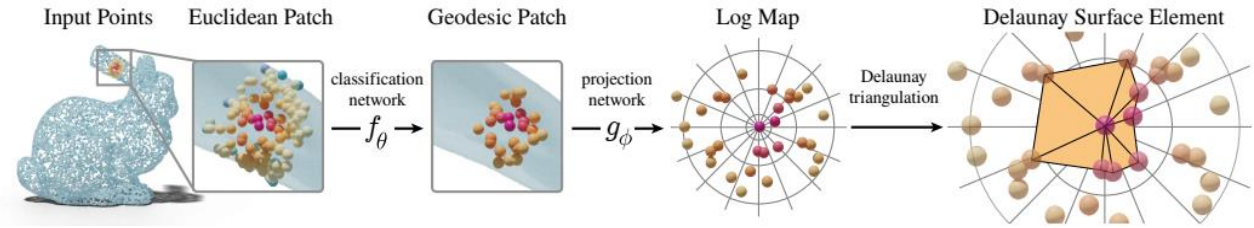
- مثلث‌هایی که در سه DSE ظاهر شده‌اند.
- مثلث‌هایی که در دو DSE ظاهر شده‌اند.
- مثلث‌هایی که تنها در یک DSE ظاهر شده‌اند.



شکل شماره ۶- تصویری از نحوه پیاده سازی الگوریتم مثلث بندی دلانی برپایه شبکه عصبی[11]

* **log map**

برای تشکیل یک مش که در آن هر لبه دقیقاً به دو مثلث متصل باشد، نیاز است که مثلث‌های تولیدشده در هر سطح دلانی، بیش از یکبار ظاهر شده باشند.



شکل شماره ۷- تصویری از نحوه پیاده سازی الگوریتم مثلث بندی دلانی برپایه شبکه عصبی[11]

مرحله آخر مربوط به ارزیابی مش و نمایش نتیجه نهایی است. برای ارزیابی مش تولیدشده، چند معیار کلیدی مورد استفاده قرار می‌گیرد:

- درصد لبه‌های غیرمانفولد یاNW *:

این معیار درصد لبه‌هایی از مش را نشان می‌دهد که بیش از دو مثلث به یک لبه متصل شده‌اند یا لبه‌ای بدون اتصال مناسب باقی مانده است. این معیار به طور مستقیم کیفیت مش را ارزیابی می‌کند. درصد پایین‌تر نشان‌دهنده مش‌های باکیفیت‌تر است.

- فاصله چمفر یاCD †:

این معیار برای ارزیابی دقت بازسازی سطح استفاده می‌شود. فاصله چمفر تفاوت بین نقاط مش بازسازی‌شده و ابر نقاط را اندازه‌گیری می‌کند. فاصله کمتر نشان‌دهنده دقت بالاتر در بازسازی سطح است.

- خطای زاویه‌ای نرمال یا NR ‡:

این معیار زاویه بین نرمال‌های سطح مش بازسازی‌شده و نرمال‌های سطح اصلی را اندازه می‌گیرد. زاویه کمتر به معنای همخوانی بیشتر سطح بازسازی‌شده با سطح اصلی می باشد.

از مزایا این روش می توان به ایجاد مش دقیق، حفظ دقت هندسی و استفاده از نگاشت های لگاریتمی اشاره کرد. همچنین این روش در مواجهه با داده‌های غیر یکنواخت عملکرد مناسبی دارد. معایب شامل حساسیت به هندسه‌های پیچیده، چالش در بخش‌های نازک، و وابستگی به داده‌های آموزشی مناسب است.

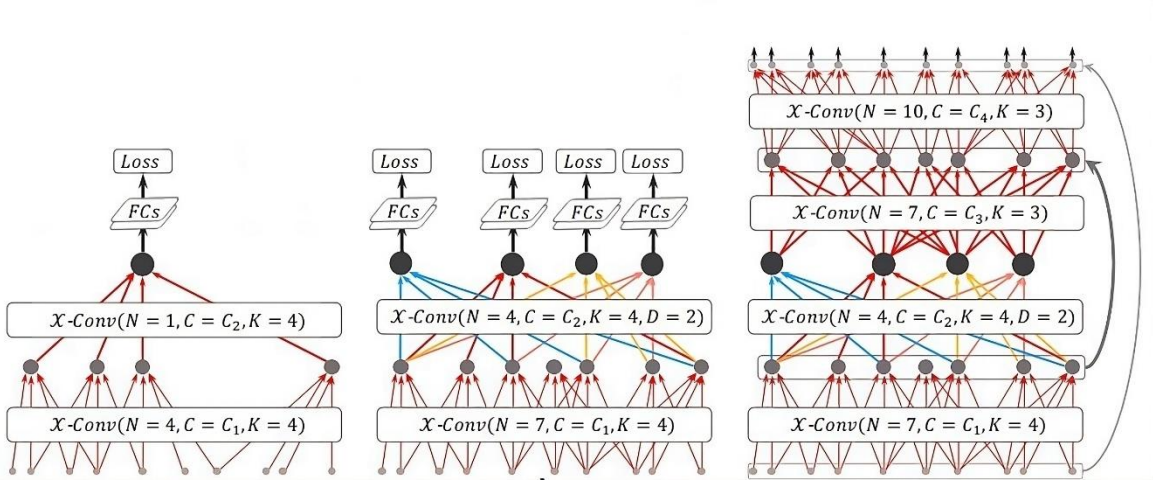
- **روش Point CNN**

روش دیگر برای پردازش ابر نقاط و ایجاد مش، روش PointCNN می باشد که توسط یانگیان و همکاران [12] ارائه شده است. این الگوریتم از یک شبکه عصبی پیچشی دارای سلسله مراتب برای یادگیری ویژگی های ابر نقاط استفاده می نماید. بطوریکه یادگیری این ویژگی ها بصورت لایه ایی انجام می پذیرد و در هر لایه ویژگی های پیچیده تری محاسبه می شود. این عملیات یادگیری بر روی ویژگی های محلی، انتخاب همسایه های محلی، کاهش تعداد نقاط و تکرار عملیات در چندین لایه انجام می پذیرد. فرض کنید داده‌های ورودی شامل ۱۰۲۴ نقطه باشد. در لایه اول الگوریتم، ممکن است تعداد نقاط به ۵۱۲ کاهش یابد اما ویژگی‌های این ۵۱۲ نقطه عمق بیشتری پیدا می‌کند. این عملیات تا لایه‌های بعدی ادامه پیدا می‌کند تا جایی که تعداد کمی از نقاط باقی می‌ماند. این ساختار پیچشی سلسله مراتبی به PointCNN اجازه می‌دهد تا هم

* **Non-watertight Edges**
† **Chamfer Distance**
‡ **Normal Reconstruction Error**

ویژگی‌های محلی را از داده‌های نقطه‌ای استخراج کند و هم به مرور ویژگی‌های کلی‌تر و پیچیده‌تر در مورد شکل دست پیدا کند. سپس باید الگوریتم را آموزش دهیم. آموزش شبکه شامل تنظیم وزن‌های مدل بر اساس داده‌های آموزشی است تا بتواند به درستی وظایفی مانند طبقه‌بندی را انجام دهد. برای این کار، از یک الگوریتم یادگیری و بهینه‌سازی استفاده می‌شود که وزن‌های شبکه را با توجه به داده‌های ورودی و برچسب‌های هدف تنظیم کند. اولین گام در آموزش شبکه تعریف تابع هدف می باشد. سپس از یک الگوریتم بهینه ساز استفاده شده تا با توجه به گرادیان تابع هدف نسبت به وزن‌ها، تغییرات لازم را در وزن‌ها اعمال کند و مقدار تابع هدف به حداقل برساند. آموزش شبکه به صورت تکراری انجام می‌شود، یعنی چندین بار داده‌های آموزشی از طریق شبکه عبور داده شده و وزن‌های شبکه به‌روزرسانی می‌شوند. پس از اتمام آموزش، مدل باید با استفاده از داده‌های تست که در طول آموزش و اعتبارسنجی استفاده نشده‌اند، آزمایش شود. در نهایت شبکه از طریق داده های لایه های عمیق تر پردازش شده تا به خروجی نهایی منجر شود. شماتیکی از روند الگوریتم را می توان در شکل شماره (۷) مشاهده کرد.

از نقاط قوت این الگوریتم می توان به عدم حساسیت نسبت به ترتیب نقاط، قابلیت یادگیری ویژگی های پیچیده و سازگاری با داده های نامنظم اشاره کرد. اما پیچیدگی محاسباتی بالا منجر شده تا دیگر روش ها مورد توجه قرار بگیرند.



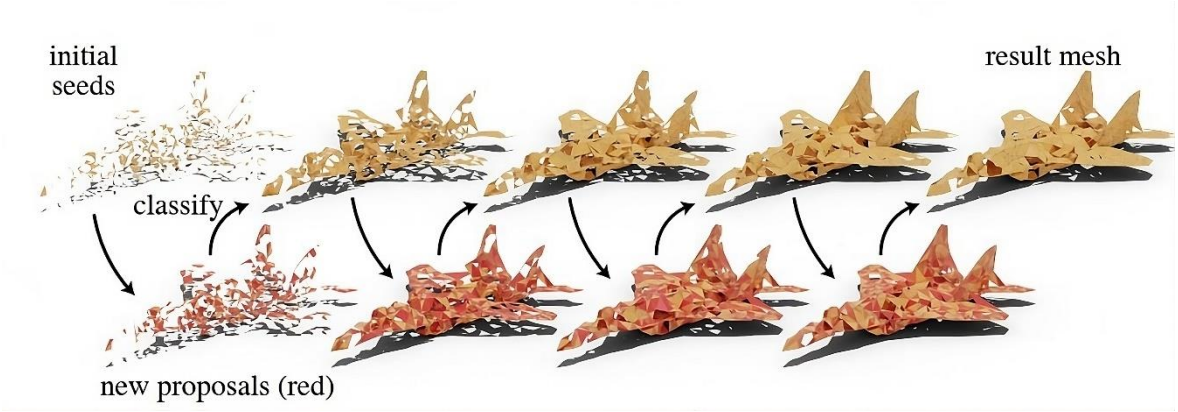
شکل شماره ۸- تصویری از نحوه پیاده سازی الگوریتم Point CNN [12]

- **روش های Point ordering**

اولویت قرار دادن برخی نقاط در دامنه میدان باعث می شود توجه بیشتری به نقاط بحرانی و نواحی پیچیده شود. به همین دلیل با بررسی ویژگی های نقاط، برخی از آنها را به عنوان نقاط با پتانسیل بالا برای تشکیل سلول های مناسب شناسایی می کنیم و در شبکه نهایی قابل روئیت هستند. در ادامه به بررسی روشی که زیر مجموعه Point ordering می باشد، پرداخته شده است

- روش **Pointtrinet**

نیکولاس و همکاران [13] روش pointtrinet را در سال ۲۰۲۰ ارائه دادند که قادر است شبکه ایی ۳ بعدی متشکل از سلول های مثلثی را با استفاده از دو شبکه عصبی به نام های ۱) شبکه پیشنهاد دهنده *و ۲) شبکه طبقه بندی†، ایجاد کند. شبکه پیشنهاد دهنده وظیفه دارد که از میان این نقاط، نقاطی را که می‌توانند با هم یک مثلث تشکیل دهند انتخاب کند. این شبکه با استفاده از اطلاعات موجود در داده‌های ابر نقاط (مانند موقعیت نسبی نقاط و ویژگی‌های مکانی آنها) تعیین کند که چه نقاطی به هم نزدیک‌تر هستند و به احتمال زیاد می‌توانند یک مثلث معتبر تشکیل دهند. فرض کنید مجموعه‌ای از ابر نقاط دارید که باید به صورت مثلثی در شبکه مرتب شوند. شبکه پیشنهاد دهنده ابتدا نقاطی را که در همسایگی یکدیگر قرار دارند شناسایی کرده و به عنوان کاندیدای تشکیل یک مثلث جدید انتخاب می‌کند. سپس شبکه طبقه‌بندی وظیفه دارد تصمیم بگیرد که آیا این مثلث‌ها باید در شبکه نهایی لحاظ شوند یا خیر. شبکه طبقه‌بندی مانند یک فیلتر عمل می‌کند که فقط مثلث‌های معتبر را تأیید می‌کند. این شبکه بررسی می‌کند که آیا این سه نقطه واقعاً می‌توانند یک مثلث معتبر و صحیح در شبکه نهایی تشکیل دهند یا خیر، و در صورت تأیید، آن را به شبکه اضافه کند. این دو شبکه به صورت تکراری و ترکیبی از هم کار می‌کنند. به جای بررسی تمامی نقاط، شبکه پیشنهاد دهنده تعداد محدودی نقاط کاندیدا را پیشنهاد داده، که این امر پیچیدگی محاسباتی را به شدت کاهش می‌دهد. شبکه طبقه‌بند باعث می‌شود که فقط مثلث‌هایی که از نظر توپولوژیکی و هندسی معتبر هستند در شبکه نهایی لحاظ شوند، که بهبود کیفیت شبکه تولید شده را تضمین می‌کند. انعطاف پذیری مناسب در تولید مثلث ها بدون نیاز به تعریف دستی مثلث های کاندید و کاهش پیچیدگی محاسباتی از مزایا بارز این الگوریتم می باشد. در حین حال، ایجاد حفره در شبکه نهایی به دلیل نادیده گرفتن برخی نقاط از بزرگترین نقاط ضعف این روش به شمار می آید.



شکل شماره ۹- تصویری از شبکه تولید شده توسط Point CNN [13]

برای بهبود این روش در آینده، می‌توان از الگوریتم‌های یادگیری قوی‌تری برای هماهنگی دقیق‌تر نقاط کاندید استفاده کرد. همچنین بهبود روش مثلث بندی می تواند باعث کاهش ایجاد حفره در شبکه نهایی شود.

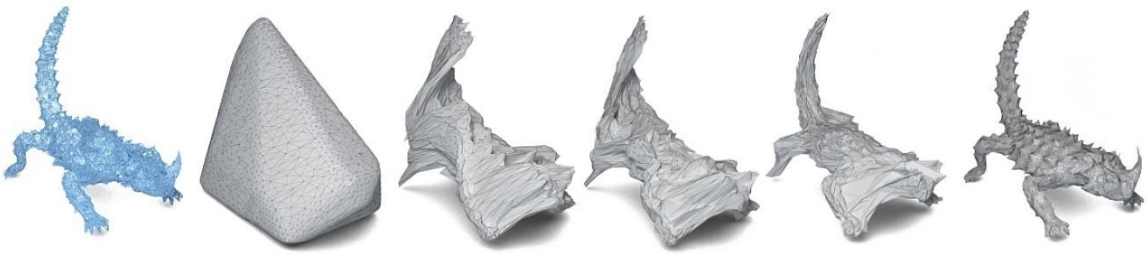
* **Proposal Network**
† **Classification Network**

۲.۲. روش های غیر مستقیم

همانطور که قبل تر بیان شد، روش های غیر مستقیم نمی توانند عملیات مناسبی بر روی داده های ابر نقاط انجام دهند. زیرا ابر نقاط دارای ماهیت غیر منظم هستند که استفاده از آنها را به چالش بزرگی تبدیل کرده است. این روش با تبدیل و دسته بندی نقاط می تواند عملیات مناسبی را با استفاده از شبکه عصبی برای تولید شبکه بر روی آنها پیاده سازی کند.

- **روش Point2mesh**

روش دیگری که بسیار مورد توجه قرار گرفته، روش mesh2point نام دارد که در سال ۲۰۲۰ توسط رانا و همکاران [14] معرفی شد. این روش حاوی تکنیک مبتنی بر شبکه عصبی عمیق برای بازسازی سطح مش با استفاده از ابر نقاط می باشد. این فرآیند با استفاده از یک مش اولیه که تخمین ساده ایی از شکل ابرنقاط است شروع می شود. سپس به کمک شبکه عصبی، شبکه ایجاد شده دچارتغییر شده تا به شکل نهایی ابرنقاط در آید. مش اولیه دارای نقشه ی ساده ایی است که ابر نقاط را توصیف می کند، اما دقیق یا پیچیده نیست. سپس شبکه عصبی، مش اولیه را به‌تدریج به سمت ابر نقطه منطبق می‌کند تا مش نهایی به دست آید. این کار توسط عملیاتی به نام shrink wrap صورت می پذیرد. بطوریکه هر بار یک سری تغییرات کوچک در موقعیت رأس‌های مش اعمال می شود. این شبکه نه تنها مش را به ابر نقطه نزدیک می‌کند، بلکه سعی می‌کند ویژگی‌های ساختاری مانند رشد نرم، پیوستگی، و تکرارهای هندسی را نیز حفظ کند. البته فرآیند بهینه‌سازی و تغییر مش در چندین مرحله و به‌صورت تکراری انجام می‌شود. ابتدا شبکه عصبی فاصله بین مش فعلی و ابر نقطه را محاسبه می‌کند تا ببیند چقدر مش به شکل نهایی نزدیک شده است.



شکل شماره ۱۰- تصویری از شبکه تولید شده توسط Point2mesh[۱۴]

یکی از ویژگی‌های مهم Point2Mesh این است که در طی این فرآیند، شبکه می‌تواند نویزها و بی‌نظمی‌های هندسی را که ممکن است در ابر نقطه وجود داشته باشند، حذف کند. در نهایت، پس از طی کردن تمام مراحل بهینه‌سازی، یک مش نهایی که سطح آن با ابر نقطه ورودی همخوانی دارد، تولید می‌شود.

از مزایا روش Point2Mesh می توان به حذف نویز و تکمیل نواحی گم شده در حین تکمیل تصویر برداری ها برای ایجاد ابر نقاط اشاره کرد. اما در عین حال این روش نیاز به داده های آموزشی خارجی ندارد و می تواند همخوانی مناسبی را برای داده های ناقص فراهم سازد. اما از معایب آشکار این روش می تواند به زمانبر بودن پیاده سازی الگوریتم و عدم توانایی برای ایجاد مش اولیه برای هندسه های پیچیده اشاره نمود .

- **روش REIN**

برای تولید یک شبکه سه بعدی یکپارچه، استفاده از روش REIN که توسط رانگل و همکاران [15] در سال ۲۰۲۰ ارائه شده، توصیه می گردد. این روش قادر است با دریافت ابر نقاط و پیاده سازی یک شبکه عصبی تکراری، مش سه بعدی تولید کند. این الگوریتم به صورت پایین به بالا و به صورت مرحله به مرحله، مش را با استفاده از اطلاعات موجود در ابر نقطه‌ها تولید می‌کند. ابتدا بردار پنهان* از ابر نقاط با استفاده از الگوریتم point Net ایجاد می شود. یک بردار به عنوان نماینده کل ساختار ابرنقاط عمل می کند بطوریکه اطلاعاتی نظیر شکل کلی، موقعیت نقاط و ویژگی‌های محلی را در بر می‌گیرد. فشرده‌سازی اطلاعات باعث می‌شود که الگوریتم بتواند با کارایی بهتری کار کند. سپس الگوریتم شروع به کار می کند و در هر بار تکرار نقاط را از ابرنقطه خوانده و با توجه به بردار پنهان، مش را ایجاد می کند. ترتیب ورود نقاط به شبکه اهمیت دارد و معمولاً الگوریتم به صورت ترتیبی نقاط را وارد می‌کند تا بتواند بهترین ساختار را از طریق ارتباط با نقاط قبلی بسازد. با معرفی هر نقطه جدید، وضعیت مش به‌روزرسانی می‌شود. این یعنی پس از ورود نقطه جدید، الگوریتم بررسی می‌کند که این نقطه چگونه می‌تواند به نقاط قبلی متصل شود تا بخشی از سطح مش را تشکیل دهد. اتصالاتی که بین نقاط جدید و نقاط قبلی شکل می‌گیرد، به ایجاد اضلاع مش کمک می‌کند. این اضلاع نمایانگر بخش‌هایی از سطح جسم هستند که به مرور با ورود نقاط بیشتر، مش کامل‌تری ایجاد می کنند. سپس عملیات نهایی برای تکمیل و ارزیابی مش تولید شده انجام می‌شود. این مراحل شامل ارزیابی کیفیت مش نهایی، تولید خروجی، و در نهایت بهبود و تصحیح مش با استفاده از روش‌های بازبینی و اصلاح می باشد. برای ارزیابی کیفیت مش تولید شده از معیار های مختلفی از جمله معیار فاصله، مقایسه نرمال های سطح †و مقایسه نقاط و سطح مش استفاده شده است. پس از عبور از این مرحله، مش تولید شده در یک فایل سه بعدی ذخیره شده تا برای تحلیل و شبیه سازی مورد استفاده قرار گیرد.

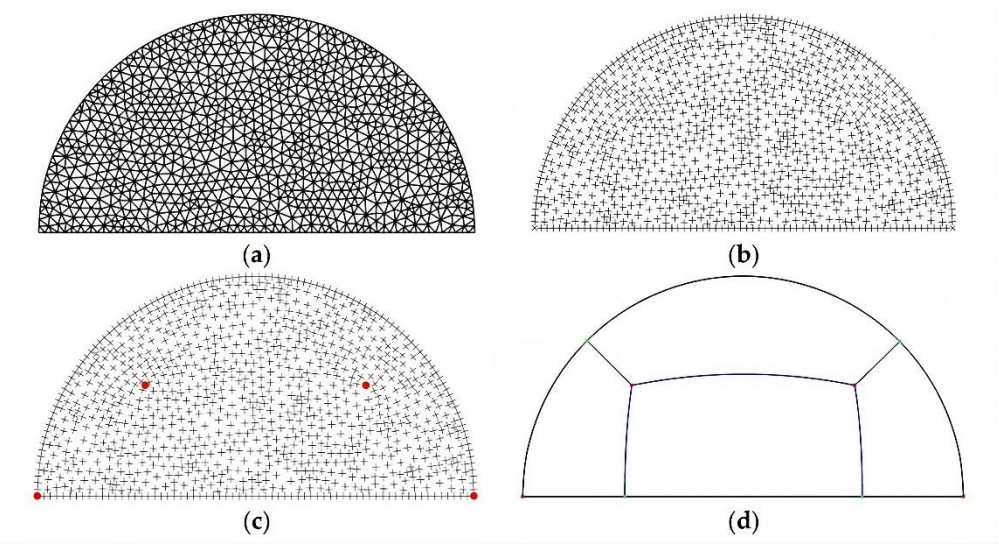
از مزایا استفاده از این روش می توان به انعطاف پذیری بالا نسبت به پذیرش تعداد بالایی از ابر نقاط، بالا بودن دقت در بازسازی سطوح پیچیده و نوآوری استفاده از بردار پنهان به عنوان نماینده از ویژگی های هندسه اشاره کرد. با وجود این مزایا، پیچیدگی بالا محاسبات، حساس بودن الگوریتم به پذیرش داده های دارای نویز و نیازمندی به استفاده از حافظه زیاد ما را بر آن داشت تا به استفاده از روش های دیگری روی آوریم.

- **روش تولید شبکه‌های چهارضلعی بلوکی**

از دیگر روش های مورد توجه در دنیا محاسبات می توان به استفاده از یک شبکه عصبی پیچشی برای تولید شبکه‌های چهارضلعی بلوکی در هندسه های ۲ بعدی اشاره کرد. این روش اولین بار توسط یوژیانگ و همکاران [16] معرفی شد. در این روش ابتدا با تعریف مدل هندسی، تولید شبکه پس‌زمینه مثلثی و تخصیص مختصات و بردارهای چارچوب به هر نقطه از شبکه شروع می‌شود. سپس این نقاط و بردارها برچسب‌گذاری شده و به‌عنوان داده‌های ورودی برای آموزش شبکه عصبی پیچشی مورد استفاده قرار می‌گیرند. شبکه عصبی پس از یادگیری از داده‌های آموزشی، قادر است خطوط جداسازی و ساختارهای تکین را در مدل‌های جدید تشخیص داده و شبکه‌های چهارضلعی تولید کند. در نهایت، خطوط جداسازی ایجاد شده و با تقسیم مدل به زیرناحیه‌های بلوکی، شبکه نهایی ایجاد می‌شود. پس از چندین دور آموزش، شبکه به تدریج یاد می‌گیرد که بردارهای چارچوبی را که در نزدیکی خطوط جداسازی قرار دارند شناسایی کند. پس از اتمام مرحله آموزش،

* **latent vector**
† **Point normal similarity**

شبکه عصبی پیچشی آموزش دیده و آماده می‌شود تا با داده‌های جدید کار کند و ساختارهای تکین را به طور خودکار شناسایی کند.



شکل شماره ۱۱- نمونه ایی از روند پیاده سازی الگوریتم تولید شبکه چهارضلعی بر پایه شبکه عصبی[۱۶]

دقت بالا و سرعت بیشتر، خودکارسازی فرآیند و وجود پایداری مطلوب در تحلیل های عددی ما را بر آن داشت تا از این الگوریتم استفاده بهینه ایی داشته باشیم. با این وجود، نقاط ضعفی از جمله نیاز به آموزش های تکراری و پیچیدگی های تنظیمات غیر قابل چشم پوشی هستند.

با این حال استفاده از تکنیک‌های یادگیری تطبیقی می‌تواند به شبکه کمک کند تا بدون نیاز به آموزش مجدد، برای هندسه های جدید بهتر عمل کند. همچنین توسعه الگوریتم‌هایی که نیاز به داده‌های آموزشی کمتری داشته باشند، می‌تواند کارایی و دقت روش را افزایش دهد.

۳. نتیجه گیری

بهره گیری از هوش مصنوعی در پردازش داده های ابر نقاط بی شک باعث افزایش کارآیی و بهبود عملکرد در بازسازی سه‌بعدی و تولید شبکه‌های محاسباتی منجر شده است. الگوریتم‌هایی مانند PointNet و PointNet++ با پردازش مستقیم ابر نقاط، نیاز تبدیل این داده‌ها به شبکه‌های منظم را از بین برده و با انعطاف‌پذیری بیشتر، توانسته داده‌های نامنظم را با دقت بالا مدیریت کنند. با این حال، این روش‌ها همچنان با چالش‌های مهمی مواجه هستند. یکی از چالش‌های اساسی، پیچیدگی محاسباتی این الگوریتم ها می باشد. شبکه‌های عصبی، به‌ویژه در پردازش داده‌های بزرگ و پیچیده مانند ابر نقاط، نیازمند منابع پردازشی و محاسباتی قدرتمند و حافظه بالا هستند. این امر باعث می شود اجرای این الگوریتم‌ها برای داده‌های عظیم و جزئیات دقیق، بسیار زمان‌بر و پرهزینه باشد. همچنین، روش‌های فعلی در مواجهه با داده‌های ناقص و یا حاوی نویز، بهبود بیشتری نیاز دارند.

از دیگر چالش‌ها می‌توان به وجود تراکم نامنظم نقاط اشاره نمود. داده‌های ابر نقاط معمولاً در برخی نواحی تراکم و چگالی بالایی دارند. این عدم یکنواختی چگالی می‌تواند درک صحیح ساختارهای پیچیده را برای الگوریتم‌ها دشوار سازد و منجر به کاهش دقت در محاسبات و اعمال تغییرات شود. در نهایت، یکی از مسائل مهم دیگر حساسیت به هندسه‌های

بسیار پیچیده و نازک است. امید بود با به کار گیری شبکه های عصبی برای تولید شبکه بتوانیم بر این مشکل فائق آییم اما چنین نیست! این شبکه های عصبی هنوز قادر به بازسازی و تحلیل ساختار های نازک نیست. علاوه بر این، زمان آموزش و تنظیم پارامترهای شبکه به عنوان یکی دیگر از مشکلات بزرگ، نیازمند بهینه‌سازی‌های بیشتری است. در نتیجه، اگرچه هوش مصنوعی به‌طور قابل توجهی قابلیت‌های پردازش، تحلیل ابر نقاط را افزایش داده، اما چالش‌های مرتبط با پیچیدگی محاسباتی، مدیریت داده‌های ناقص و نامنظم، و بهبود دقت در ساختارهای پیچیده نشان می‌دهند که پیشرفت‌های بیشتر در این حوزه ضروری می باشد.

۴. مراجع

1. Diab, A., et al. (2022). "Deep learning for LiDAR point cloud classification in remote sensing." Sensors **22**(20): 7868.
2. Yue, X., et al. (2018). A lidar point cloud generator: from a virtual world to autonomous driving. Proceedings of the 2018 ACM on international conference on multimedia retrieval.
3. mGarrido, D., et al. (2021). Point cloud interaction and manipulation in virtual reality. 2021 5th International Conference on Artificial Intelligence and Virtual Reality (AIVR).
4. Zhang, W., et al. (2022). "Point cloud computing algorithm on object surface based on virtual reality technology." Computational Intelligence 38(1): 106-120.
5. Kubátová, D., et al. (2023). "Data Preparing for Reverse Engineering." Advances in Science and Technology 132: 320-329.
6. Wang, R., et al. (2023). "Improving point cloud classification and segmentation via parametric veronese mapping." Pattern Recognition **144**: 109784.
7. Qi, C. R., et al. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition.
8. Qi, C. R., et al. (2017). "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in Neural Information Processing Systems 30.
9. Chibane, J. and G. Pons-Moll (2020). "Neural unsigned distance fields for implicit function learning." Advances in Neural Information Processing Systems 33: 21638-21652.
10. Liu, M., et al. (2020). Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16, Springer.
11. Rakotosaona, M.-J., et al. (2021). Learning delaunay surface elements for mesh reconstruction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
12. Li, Y., et al. (2018). "Pointcnn: Convolution on x-transformed points." Advances in Neural Information Processing Systems 31.

13. Sharp, N. and M. Ovsjanikov (2020). Pointtrinet: Learned triangulation of 3d point sets. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16, Springer.
14. Hanocka, R., et al. (2020). "Point2mesh: A self-prior for deformable meshes." arXiv preprint arXiv:2005.11084.
15. Daroya, R., et al. (2020). REIN: Flexible mesh generation from point clouds. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.
16. Zhou, Y., et al. (2023). "Quadrilateral Mesh Generation Method Based on Convolutional Neural Network." Information 14(5): 273.