

**Graph Neural Network**

**ניתוח מרחב התלת-ממד באמצעות רשתות נוירונים גרפיות**

בורודולין רומן

בהנחיית ד״ר מיה הרמן2024ג

תוכן עניינים

[תקציר 3](#_Toc174480545)

[הקדמה 4](#_Toc174480546)

[פרק 2: רקע תיאורטי 5](#_Toc174480547)

[2.1 תיאור כללי של רשתות נוירונים גרפיות (GNN) 6](#_Toc174480548)

[2.1.2 שכבות 7](#_Toc174480549)

[2.1.2 שכבות מרובות GNN 9](#_Toc174480550)

[2.2 טכנולוגיות זיהוי עצמים בתלת ממד Point-GNN 10](#_Toc174480551)

[2.2.1 בניית Point-GNN 11](#_Toc174480552)

[2.2.2 רשת נוירונים גרפית עם רישום אוטומטי (Auto-Registration) 11](#_Toc174480553)

[2.3 סגמנטציה סמנטית באמצעות תמונות RGBD 13](#_Toc174480554)

[פרק 3: שיטות 14](#_Toc174480555)

[פרק 4: תוצאות 15](#_Toc174480556)

[פרק 5: דיון 16](#_Toc174480557)

[פרק 6: מסקנות והמלצות להמשך המחקר 17](#_Toc174480558)

[ביבליוגרפיה 18](#_Toc174480559)

[נספחים 18](#_Toc174480560)

# תקציר

...סקירה קצרה של העבודה, אכתוב לאחר סיום העבודה...

# הקדמה

**מבוא ל Graph Neural Networks**

למידת עומק גרפית GDL & GNN התפתחה במידה ניכרת בשנים האחרונות. האופי של בעיות רבות מחיי היומיום הופך את GDL & GNNלכלי רב-תכליתי ויעיל. התחום הזה מציג פוטנציאל רב בתחומים כמו מדיה חברתית, גילוי תרופות, מיקום שבבים, חיזוי, ביו-אינפורמטיקה, מציות רבודה, ניתוח גופים במרכב ועומקם.

בעבודה זו, אני מתכוון להציג ניתוח של רשתות נוירונים גרפיות נפוצות ולבחון את היבטיהן המתמטיים - מעין סקר מקוצר. בנוסף אציין מספר שיטות חדשניות העוסקות בניתוח עומק וניתוח מרחב בעזרת GNN.

הרעיון המרכזי של למידת עומק גרפית הוא למידת המאפיינים המבניים והמרחביים של גרפים המורכבים מצמתים וקשרים, המייצגים ישויות שונות ואת האינטראקציות ביניהן.

**חשיבות הנושא**

היכולת של רשתות עצביות גרפיות לנתח מערכות מורכבות הפכה אותן לכלי מרכזי במגוון תחומים:

* **ברשתות חברתיות** - רשתות עצביות גרפיות ידועות ביכולת שלהן לזיוהי קבוצות חברים ולחזות קשרים עתידיים.
* **כימיה וביולוגיה** - מאפשרות חיזוי תכונות כימיות של מולקולות על בסיס המבנה שלהן.
* **פיננסי** - משמשות לניתוח רשתות סחר ולחזות את התנהגות השוק.
* **אפליקציות ניווט** - רשתות עצביות גרפיות מסייעות בזיהוי עומס בכבישים, מה שמאפשר לאפליקציות ניווט לספק מסלולים יעילים יותר בזמן אמת ולהתאים את ההמלצות בהתאם לתנאי התנועה.
* **רכב אוטונומי** - רשתות עצביות גרפיות מיושמות גם בתחום הרכב האוטונומי, שם הן מאפשרות זיהוי מדויק ככל הניתן של הכבישים והמכשולים בזמן הנהיגה והקשרם אחד לשני.
* **מציאות רבודה** - מאפשרות זיהוי והבנה של אובייקטים במרחב, ומשפרות את היכולת של מערכות AR לשלב אלמנטים וירטואליים עם העולם האמיתי

# רקע תיאורטי

## גרפים

הגרפים הם מבנים מתמטיים המשמשים לייצוג מערכות מורכבות של קשרים בין אובייקטים. הם נמצאים בשימוש נרחב בתחומים כמו מדעי המחשב, מתמטיקה, פיזיקה, סוציולוגיה ועוד. הבנת היסודות של גרפים היא קריטית לניתוח ופתרון בעיות במגוון תחומים.

**הגדרת גרף: צמתים (Nodes) וקשתות (Edges)**

**גרף** מוגדר כזוג מסודר כאשר:

* - היא קבוצה של **צמתים Vertices או Nodes** הצמתים מייצגים את האובייקטים או הישויות במערכת.
* - היא קבוצה של **קשתות (Edges)**, המייצגות את הקשרים או היחסים בין הצמתים. כל קשת היא זוג של צמתים (בגרף לא מכוון) או זוג מסודר (בגרף מכוון).

**סוגי גרפים**

**1. גרף לא מכוון (Undirected Graph):**

בגרף לא מכוון, הקשתות אינן מכוונות, כלומר אין הבדל בין הקצה המתחיל לקצה המסתיים. הקשתות מייצגות קשר דו-כיווני בין הצמתים.

* **ייצוג קשת:** כקבוצה בלתי מסודרת המייצגת קשר בין הצמתים ו-
* **דוגמה:** ברשתות חברתיות, החברות היא לרוב דו-כיוונית.

**2. גרף מכוון Directed Graph או Digraph**

בגרף מכוון, הקשתות הן בעלות כיוון, כלומר הקשר בין הצמתים הוא חד-כיווני.

* **ייצוג קשת:** כזוג מסודר המייצג קשת מ- ל-

**3. גרף משוקלל (Weighted Graph):**

בגרף משוקלל, לכל קשת משויך **משקל** (מספר ממשי), המייצג מידע נוסף על הקשר.

* **משקלים יכולים לייצג:** מרחק, עלות, זמן, קיבולת וכו'.

### ייצוג גרפים

**1. מטריצת שכנות (Adjacency Matrix):**

מטריצה בגודל (כאשר הוא מספר הצמתים), המשמשת לייצוג הקשרים בין הצמתים.

* **בגרף לא משוקלל:**
* **בגרף משוקלל:**

**2. רשימת שכנות (Adjacency List):**

מערך או רשימה שבה כל צומת מחזיקה רשימה של הצמתים שאליהם היא מקושרת (פחות רלוונטי לGNN)

### מושגים בסיסיים

**1. דרגת צומת (Degree of a Node):**

* **בגרף לא מכוון:**
  + **דרגה :** מספר הקשתות המחוברות לצומת
    - מסומן כ
* **בגרף מכוון:**
  + **דרגת כניסה (In-degree):** מספר הקשתות המגיעות לצומת
  + **דרגת יציאה (Out-degree):** מספר הקשתות היוצאות מצומת

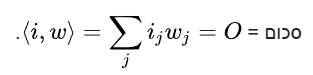
מוסגים כמו : Simple Path, Cycle, Strongly Connected, Weakly Connected, Connected Graph פחות רלוונטי לGNN ולכן לא הוסבר כאן

**סיכום**

הגרפים הם כלי עוצמתי לייצוג וניתוח מערכות מורכבות. המושגים הבסיסיים של צמתים וקשתות, יחד עם הסוגים השונים של גרפים ודרכי הייצוג שלהם, מהווים את היסוד להבנה עמוקה יותר של תורת הגרפים ויישומיה. הבנה זו חיונית במיוחד בתחומים כמו רשתות נוירונים גרפיות (GNN), שבהם הגרפים משמשים כמבנה נתונים מרכזי ללמידה וניתוח של מידע מורכב.

## רשתות ניורוניות

1. **הקלטים** אלו הם הקלטים שמוזנים לנוירון. כל ערך מייצג משתנה שנכנס לנוירון.



1. **המשקלים** לכל קלט יש משקל שמקושר אליו. המשקלים מייצגים את החשיבות של כל קלט עבור הנוירון.
2. **סכימה** הנוירון מחשב סכום משוקלל של הקלטים באמצעות המשקלים
3. **פלט** זהו הערך שמתקבל מסכום הקלטים המשוקללים.
4. **פונקציה** הפונקציה המופעלת על הפלט היא פונקציית האקטיבציה, תפקידה להוסיף מידה של אי-לינאריות למודל, מה שמאפשר לרשת ללמוד דפוסים מורכבים יותר. הפונקציה יכולה להיות, לדוגמה או טנגנס היפרבולי.

הרעיון המרכזי הוא שהנוירון מקבל קלטים, שוקל אותם, מחשב סכום, ואז מפעיל פונקציית אקטיבציה כדי לקבוע את הפלט הסופי.

INPUT layer

OUTPUT layer

HIDDEN layers

1. **שכבת הקלט (INPUT layer)** זו השכבה הראשונה של הרשת, בה נמצאים הנוירונים שמייצגים את הקלטים לרשת. כל נוירון בשכבה זו מקבל ערך שנכנס למערכת, כמו פיקסלים של תמונה, נתוני טקסט, או נתונים מספריים אחרים.
2. **שכבות נסתרות (HIDDEN layers)** אלו השכבות האמצעיות שמתווכות בין הקלט לפלט. בשכבות אלו מתבצע עיבוד המידע. כל נוירון בשכבה נסתרת מחובר לנוירונים מהשכבה הקודמת והבאה, ומשתמש במשקלים כדי לחשב את הפלט שלו. מספר השכבות והנוירונים בכל שכבה יכולים להשתנות בהתאם למורכבות המודל.
3. **שכבת הפלט (OUTPUT layer)** זו השכבה האחרונה ברשת שמפיקה את התוצאה הסופית של המודל, בהתאם לקלטים שעברו דרך השכבות הנסתרות. מספר הנוירונים בשכבה זו תואם למספר התוצאות האפשריות או הערכים שהמודל צריך לחזות.

הרעיון הכללי הוא שהקלטים עוברים דרך השכבות הנסתרות, שם מתבצע עיבוד מורכב של הנתונים, ולבסוף מגיעים לשכבת הפלט, שם מתקבלת ההחלטה או החיזוי הסופי של המודל. כל שכבה מוסיפה לרשת יכולות למידת תבניות ומידע ברמות שונות של מורכבות.

# תיאור כללי של רשתות נוירונים גרפיות

## תהליך תכנון של

התהליך מורכב ממספר שלבים, כולל בחירת המודולים החישוביים, קביעת סוג הגרף והיקפו (בשלב זה נעשה על ידי אדם), ועיצוב פונקציית ההפסד, השלבים מפורטים בהמשך.

תהליך התכנון הכללי של מודל *GNN* למשימה מסוימת על גרף מסוים כולל ארבעה שלבים עיקריים:

* **זיהוי מבנה הגרף**: תחילה, יש לקבוע את מבנה הגרף ביישום ,קיימים שני תרחישים מרכזיים: תרחישים מבניים, שבהם מבנה הגרף מוגדר במפורש (כמו ביישומים המתמקדים במולקולות, מערכות פיזיות וכדומה), ותרחישים לא-מבניים, שבהם מבנה הגרף מרומז ויש לבנות אותו מתוך המשימה עצמה, כגון יצירת גרף מילים לטקסט או גרף סצנה לתמונה. לבסיום בניית הגרף, ניתן להתקדם למציאת מודל *GNN* אופטימלי עבור הגרף.
* A diagram of text and words

  Description automatically generatedA person walking a dog on a leash

  Description automatically generatedA diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of a diagram of

  Description automatically generatedA diagram of a molecule

  Description automatically generated**הגדרת סוג הגרף והיקפו**: לאחר קבלת הגרף, יש לקבוע את סוגו והיקפו. גרפים בעלי סוגים מורכבים עשויים לספק מידע נוסף על הצמתים והקשרים ביניהם. בדרך כלל, גרפים מסווגים כגרפים מכוונים או לא-מכוונים (כאשר קשתות בגרפים מכוונים מספקות מידע נוסף בהשוואה לגרפים לא-מכוונים), וכן כגרפים הומוגניים או הטורוגניים (כאשר בגרפים הטרוגניים ישנם סוגים שונים של צמתים וקשתות שיש לקחת בחשבון בתהליך התכנון).

מספר דוגמאות להגדרת GNN

* **עיצוב פונקציית הפסד**: שלב זה כולל את עיצוב הפונקציה המשמשת לחישוב ההפסדים במהלך הלמידה. פונקציית הפסד היא חלק מרכזי בתהליך האימון של המודל.
* **בניית המודל באמצעות מודולים חישוביים**: לאחר קביעת מבנה הגרף, סוגו ופונקציית ההפסד, מגיע שלב בניית המודל עצמו תוך שימוש בכלים חישוביים מתאימים

### שכבות

שכבה אחת של רשת נוירונים גרפית *GNN* כוללת מספר שלבים שמתבצעים על כל צומת בגרף:

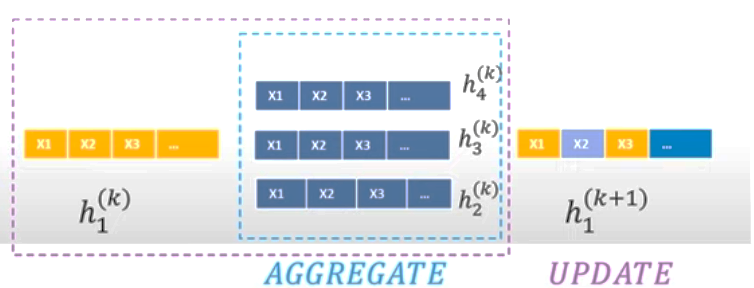
**קבלת מידע מהשכנים (Message Passing)**: רשתות נוירונים גרפיות *GNN* ידועות ביכולתן ללמוד מידע מבני. בדרך כלל, צמתים בעלי מאפיינים או תכונות דומות מקושרים זה לזה (דבר זה נכון גם בהקשר של רשתות חברתיות). ה-*GNN* מנצל עובדה זו ולומד כיצד ומדוע צמתים מסוימים מתחברים זה לזה בעוד שאחרים לא. כדי לעשות זאת, ה-*GNN* מתבונן בשכונות של הצמתים.

**הגדרת שכונה:** השכונה של צומת מוגדרת כקבוצת הצמתים שמחוברים לצומת באמצעות קשת, פורמלית האדם מושפע מהמעגל החברתי שבו הוא נמצא. בדומה לכך, יכול ללמוד הרבה על צומת על ידי התבוננות בצמתים שבשכנות שלו כדי לאפשר את שיתוף המידע הזה בין צומת מקור לבין השכנים שלו , רשתות GNN משתמשות בתהליך של **העברת מסרים**.

ב-GNN, העברת מסרים מוגדרת כתהליך שבו לוקחים את הפיצרים של הצמתים השכנים, מעבדים אותם, ו"מעבירים" אותם לצומת המקור. תהליך זה מתבצע במקביל עבור כל הצמתים בגרף, כך שבסופו של שלב זה נבדקות כל השכונות בגרף.

נבנה גרף ועבור צומת 1 נבחן את השכונה שלו נקח את כל הפיצרים של הצמתים ונעביר אותם דרך פונקציה , שיכולה להיות רשת נוירונים פשוטה *MLP* או *RNN* או טרנספורמציה אפינית בפשטות, "מסר" הוא המאפיין המעובד של הצומת שנכנס לצומת המקור.

**איגוד המידע (Aggregation)**: כעת, לאחר שיש בידינו את המסרים המעובדים שהועברו לצומת , עלינו לאגד ("לשלב") אותם בצורה כלשהי. ישנן מספר דרכים לשלב את המסרים הללו. פונקציות איגוד פופולריות כוללות:



נניח שאנו משתמשים בפונקציה כדי לאגד את המסרים מהשכנים (באמצעות סכימה, ממוצע, מקסימום או מינימום). המסרים המאוגדים הסופיים יכולים להיות מסומנים כך:

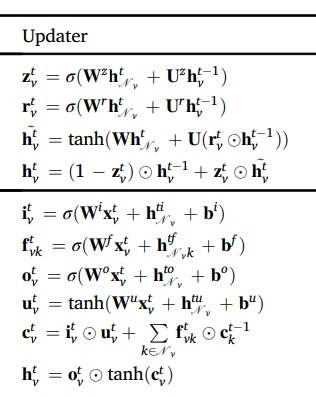
* **עדכון הייצוג (Update)**:השלב האחרון בקונבולוציה ב-*GNN* הוא עדכון הייצוג של הצומת בהתבסס על המידע המאוגד. הייצוג המעודכן הזה הופך להיות הייצוג החדש של הצומת, שמשמש לאחר מכן לחיזוי או ללמידה נוספת.

– צומת לעדכון

– צומתים המתחברים לצומת

– שכבת עדכון

**דוגמאות ל UPDATE**

****פונקציות אלו יכולות להיות רשתות נוירונים כמו רשתות רב שכבתיות (יחידת זיכרון גרפי), או שימוש בפונקציות ליניאריות כמו טרנספורמציה אפינית.

**LSTM (Long Short-Term Memory)**

**GRU (Gated Recurrent Unit)**

-מטריצות משקל.

- הייצוג הנוכחי של הצומת.

- הייצוג הביניים של הצומת.

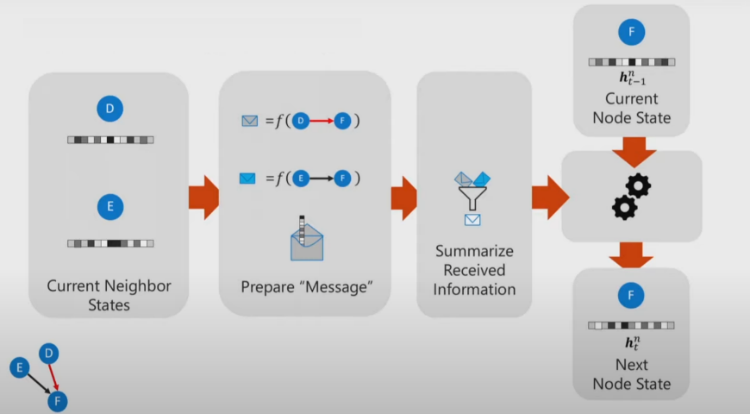
- פונקציית הסיגמואיד.

-כפל רכיב-ברכיב.

-שער האיפוס.

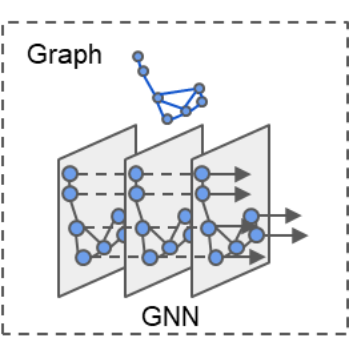
חשיבות פונקציות העדכון: פונקציות העדכון קובעות כיצד המידע עובר מצמת לשכנים שלו ומה הוא לומד מתוך המידע. הן מאפשרות לצמתים להתאים את עצמם למבנה הסביבה שלהם ולשפר את היכולת של ה- לבצע חיזויים מדויקים יותר על בסיס הקשרים הגרפיים.

### שכבות מרובות GNN

כאשר מבינים כיצד עובדת שכבה אחת של *GNN*, ניתן לבנות רשת של שכבות כאלה. הנה תהליך הזרימה של המידע בין השכבות ואופן השיפור של הייצוגים של הצמתים הקשתות על ידי ה, *GNN*

**הקלט לשכבת GNN הראשונה**:

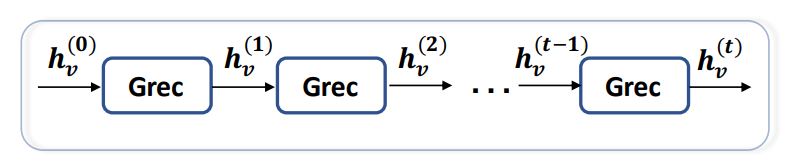
הקלט הוא מאפייני הצמתים המסומנים כ הפלט הוא הייצוגים הביניים של הצמתים המסומנים ב כאשר הוא הממד הראשון של השכבת הגרפים, מורכב מהוקטורים .



**הקלט לשכבת GNN השנייה**:

הפלט של השכבה הראשונה משמש כקלט לשכבה השנייה. הפלט מהשכבה השנייה הוא כאשר הוא הממד השני של השכבת הגרפים, מורכב מהוקטורים .

**הפלט לאחר מספר שכבות**:

**לאחר מספר שכבות, הפלט מהשכבה מסומן כ. בסופו של דבר, מורכב מהוקטורים .

ההחלטה לגבי ערכי , היא היפר-פרמטרים שנקבעים על ידי המתכנן. ניתן לחשוב עליהם כבחירת מספר היחידות (או "נוירונים") עבור כל אחת מהשכבות ברשת.

המבנה נשאר אותו הדבר, אך הייצוגים של הצמתים משתנים כל הזמן דרך השכבות. באופן אופציונלי, ייצוגי הקשתות שלך עשויים גם להשתנות, אך ללא שינוי בחיבורים או בכיווניות.

# Diagram of a diagram of a diagram Description automatically generated טכנולוגיות זיהוי עצמים בתלת ממד Point-GNN

**A diagram of a cloud computing process

Description automatically generated**Point-GNN הוא מודל מתקדם לזיהוי עצמים בתלת-ממד בתוך ענן נקודות, שמתבסס על ייצוג גרפי שמאפשר לשמור על המבנה הלא סדור של הנקודות. המודל בונה גרף שבו הקודקודים מייצגים נקודות במרחב תלת-ממדי, והקשתות מחברות בין נקודות שכנות, כך שניתן לבצע עיבוד מבני מורכב על בסיס המידע הגרפי הזה. בעזרת רשת נוירונים גרפית (*GNN*), המודל משפר את הייצוגים של הקודקודים על ידי איגוד מידע מהשכנים שלהם במספר איטרציות. תהליך זה כולל גם מנגנון אוטומטי להתאמת קואורדינטות הקודקודים, מה שמסייע למודל להיות עמיד לשינויים במיקום הנקודות. המודל נועד לפעול בתהליך זיהוי בשלב אחד, ללא צורך בעידכונים.

התמונה מציגה את הארכיטקטורה הכוללת של גישת *Point-GNN* לזיהוי אובייקטים תלת-ממדיים בענן. הארכיטקטורה מורכבת משלושה מרכיבים עיקריים:

### בניית Point-GNN

**בניית הגרף:** בהינתן ענן נקודות , עבור כל כאשר:

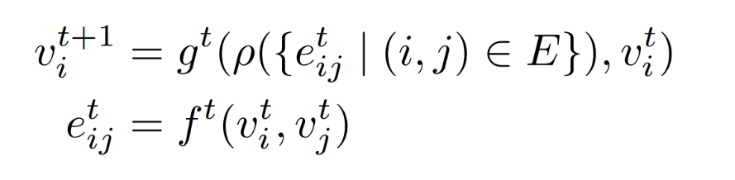
* הקואורדינטות התלת-ממדיות
* ווקטור המייצג תכונות הניקודה

הגרף על ידי שימוש בנן נקודות של - כקודקודיםנבצע חיבור של כל נקודה לשכנות שלה בתוך רדיוס קבוע כלומר, הקשתות מוגדרות כ-

שיתה זאת מתאר בניית גרף יעיל מענן נקודות צפוף על ידי שימוש בדגימת (*down sampling*) להקטנת העומס החישובי בזמן כאשר זהו מספר השכנים ו- מספר נקודות בענן .

התהליך מבוסס על חיפוש שכנים קרובים ברדיוס קבוע תוך שמירה על המידע המקורי של הנקודות באמצעות קידוד תכונות בצמתים. השימוש ברשת עצבית מרובת שכבות () לאיגוד התכונות ו-*RNN* גרפי (*GNN* ) לעיבוד הגרף מבטיח יעילות על ידי שמירה על רמת דיוק גבוהה במודל תוך הפחתת הדרישות החישוביות.

### רשת נוירונים גרפית עם רישום אוטומטי (Auto-Registration)

רשת נוירונים גרפית () משפרת את תכונות הצמתים על ידי אגירת מידע מקשתות המחברות ביניהם. בכל איטרציה, פונקציות מתמטיות כמו ו - משמשות לעדכון תכונות הצמתים על בסיס תכונות הקשתות והמצב הנוכחי של הצמתים. כדי להקטין את רגישות המודל לתזוזות קטנות בין הצמתים, משתמשים בקואורדינטות יחסיות של השכנים ומוסיפים מנגנון רישום אוטומטי () שמחשב קיזוזים לתיקון התזוזות. פעולה זו מאפשרת למודל לשמור על דיוק גבוה, גם במקרים של שינויים קטנים במבנה הגרף.

כאשר :

תכונת הצומת באיטרציה

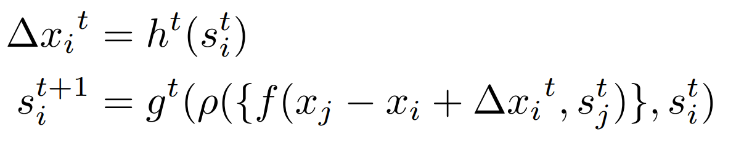
פונקציה שמעדכנת את תכונת הצומת בהתבסס על תכונות הקשתות.

פונקציית אגירה המאחדת תכונות קשתות.

תכונת צומת לצומת באיטרציה .

תכונת הקשת המחברת בין הצומת לצומת באיטרציה .

פונקציה שמחשבת את תכונת הקשת בין שני צמתים.

כדי לשפר את הדיוק ולהקטין את רגישות המודל לתזוזות קטנות, המערכת משתמשת בקואורדינטות יחסיות של השכנים ובמנגנון רישום אוטומטי

שימו לב שאנו משתמשים בקואורדינטות היחסיות של השכנים כקלט ל- לחילוץ תכונות הקשת. הקואורדינטות היחסיות כוללות אי-תלות בהסטות תרגום גלובליות של ענן הנקודות. עם זאת, הן עדיין רגישות לתרגומים בתוך אזור השכונה. כדי לצמצם את הרגישות הזו, אנו מציעים להוסיף קיזוז () בהתבסס על תכונות מבניות של השכנים, ולהציע מנגנון רישום אוטומטי ():

# סגמנטציה סמנטית באמצעות תמונות RGBD

המודל מתמקד בסגמנטציה סמנטית של תמונות שמשלבות נתוני עומק (RGBD), ומשלב בין מידע חזותי דו-ממדי ותלת-ממדי.

הרעיון המרכזי הוא להשתמש ברשת נוירונים גרפית תלת-ממדית (3DGNN) כדי להבין טוב יותר את המידע המורכב בתמונות. כל נקודה בענן הנקודות התלת-ממדי מיוצגת על ידי וקטור סמוי שמכיל מידע חזותי מתמונות דו-ממדיות, בעזרת רשת נוירונים קונבולוציונית .

בניגוד לסגמנטציה רגילה שמבוססת על תמונות דו-ממדיות, סגמנטציה של RGBD מנצלת מידע עומק כדי להבין טוב יותר את המידע הגאומטרי בתמונה. לדוגמה, בתמונה דו-ממדית, פיקסלים סמוכים על שולחן עשויים לכלול פריטים שונים, כמו מיקרוגל ושיש, בעוד שב-D3 אין בלבול כזה כי הנקודות מרוחקות זו מזו.



ישנן שיטות שמתייחסות לסגמנטציה של RGBD כמו לסגמנטציה דו-ממדית, שבה העומק משמש כתמונה נוספת. רשתות CNN מעבדות בנפרד את התמונה ואת מידע העומק. שיטות אלו מצריכות שני CNN, מה שמכפיל את כמות החישוב והזיכרון הנדרשים, ועשויות לגרום לטעויות אם לא מתחשבים במידע הגאומטרי כראוי.

A close-up of a table

Description automatically generated

איור מציג את הרשת הנוירונים הגרפית התלת-ממדית. החלק העליון של האיור מציג את ענן הנקודות התלת-ממדי ותקריב של הגרף שנבנה על בסיס ענן הנקודות. A diagram of a graphing system

Description automatically generated

הנקודות הכחולות והקווים המקווקווים השחורים מייצגים צמתים וקצוות שנמצאים בגרף שנבנה מתמונה דו-ממדית. ניתן לראות שגרף שנבנה מענן נקודות תלת-ממדי מקודד מידע גאומטרי שקשה להסיק אותו מתמונה דו-ממדית בלבד. בחלק התחתון, מוצג תת-גרף המחובר לנקודה האדומה כדוגמה להמחשת תהליך ההפצה.

כאן בהמשך אסביר את האלגוריתם שך RGBG-3DGNN

# שיטות

**3.1 בניית גרף מנתוני ענן נקודות**

* הסבר על איך יוצרים גרף מנתוני ענן נקודות והמרת תמונות RGBD לגרף.

**3.2 עיבוד הנתונים באמצעות GNN**

* תיאור האלגוריתמים והשיטות שנלקחו ללמידה על הגרף.

# תוצאות

* הצגת נתוני הבדיקה והתוצאות שהתקבלו.
* תרשימים וגרפים שממחישים את הממצאים.

# דיון

* ניתוח התוצאות והשוואה לעבודות קודמות.
* דיון בחשיבות הממצאים והשפעתם.

# מסקנות והמלצות להמשך המחקר

* סיכום העבודה והמלצות למחקרים עתידיים.

# ביבליוגרפיה

[1] A Comprehensive Survey on Graph Neural Networks, Zonghan Wu, Shirui Pan, Member, IEEE, Fengwen Chen, Guodong Long, Chengqi Zhang, Senior Member, IEEE, Philip S. Yu, Fellow, IEEE

[2] Graph neural networks: A review of methods and applications, Jie Zhou a,1 , Ganqu Cui a,1 , Shengding Hu a , Zhengyan Zhang a , Cheng Yang b , Zhiyuan Liu a,\* , Lifeng Wang c , Changcheng Li c , Maosong Sun a

[3] Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud, Weijing Shi and Ragunathan (Raj) Rajkumar Carnegie Mellon University Pittsburgh, PA 15213

[4] 3D Graph Neural Networks for RGBD Semantic Segmentation Xiaojuan Qi† Renjie Liao‡,§ Jiaya Jia†,♭ Sanja Fidler‡ Raquel Urtasun§,‡ † The Chinese University of Hong Kong ‡ University of Toronto § Uber Advanced Technologies Group ♭ Youtu Lab, Tencent

# נספחים

* קוד מקור, נתונים נוספים, תמונות או תוצאות שלא הוצגו במלואן בגוף העבודה.