לילה טוב חברים, היום אנחנו שוב בפינתנו deepnightlearners עם סקירה של מאמר בתחום הלמידה העמוקה.
הפעם בחרתי לסקירה את המאמר שנקרא:

Learning Mesh-Based Simulation with Graph Networks

שפורסם בארכיב באוקטובר 2020.
הסקירה מאת עדו בן-יאיר, אוגוסט 2021.

הוצג בכנס: ICLR 2021

תחומי מאמר:

- Partial Differential Equations
- Computational Physics
- Computational Geometry
- Graph Neural Networks

ישנו שפע של חומרים בנושא:

- https://towardsdatascience.com/a-gentle-introduction-to-graph-neural-network-basics-deepwalk-and-graphsage-db5d540d50b3
- https://www.youtube.com/watch?v=zCEYiCxrL_0
- https://www.youtube.com/watch?v=8owQBFAHw7E
- https://www.youtube.com/watch?v=fOctJB4kVlM
- https://www.youtube.com/watch?v=JAB_plj2rbA&list=PLoROMvodv4rPLKxIpqhjhPgdQy7imNkDn
- https://www.youtube.com/watch?v=2KRAOZIULzw
- https://www.youtube.com/watch?v=JSed7OBasXs

כלים מתמטיים, טכניקות, מושגים וסימונים:

- Partial Differential Equations
- Adaptive Remeshing
- Delaunay Triangulation
- Finite Element Method
- Graph Neural Networks

בהירות כתיבה: גבוהה.

רמת היכרות עם כלים מתמטיים וטכניקות של DL/ML הנדרשים להבנת המאמר: בסיסית.

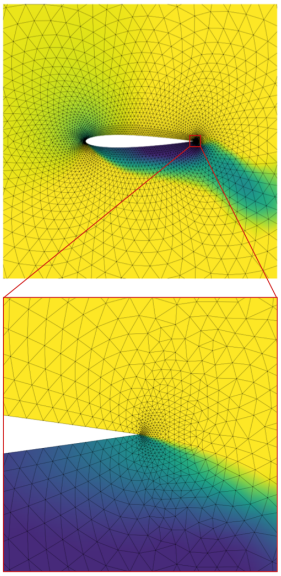
יישומים פרקטיים אפשריים: סימולציות בכל תחומי הפיזיקה וההנדסה ובעצם כל דאטה אשר ניתן לייצוג כ-mesh.

תמצית המאמר:

הלמידה החישובית מתחילה להטביע את חותמה גם על תחום הסימולציות, הידוע בשפה המקצועית בשמות פיזיקה חישובית, הנדסה חישובית או מדע חישובי. הצורך לבצע סימולציות קיים עוד מימיה המוקדמים ביותר של ההנדסה המודרנית, ויחד איתו מתעורר הקושי לבצע סימולציות בדיוק מירבי וברזולוציה גבוהה, תוך שימוש יעיל במשאבים חישוביים. תחום זה ממילא עוסק בקירובים נומריים ולא בפתרונות אנליטיים, ולכן שיטות הלמידה החישובית עשויות לספק את האיזון הרצוי בין דיוק לבין צריכת משאבים. שיטות אלה מסוגלות להתמודד בהצלחה עם מידע רב-מימדי ולמדל קשרים סבוכים מבלי להסתמך על הנדסה ידנית של האלגוריתם (כמעט). אם כך, טבעי שנרצה לנצל את גוף הידע העכשווי בלמידה חישובית גם בתחום הפיזיקה החישובית.

בשנים 2017-2019 יצאו מספר מאמרים חשובים על שימוש ברשתות קונבולוציה לפתרון מגוון בעיות על גרפים ועננים נקודות (למשל העבודות של Bronstein et al). בשנה שעברה הפתיעה DeepMind של גוגל עם יישום מעניין של רשתות קונבולוציה לתחזית של דינמיקת נוזלים באמצעות רשת message passing המקבלת כקלט ייצוג מבוסס חלקיקים (למעשה ענן נקודות ממנו נבנה גרף המייצג את דומיין הסימולציה). הרשת לומדת את הדינמיקה הפיזיקלית של הנוזל ומבצעת תחזית של השדות, הכוחות ומיקומי החלקיקים הרלוונטיים לאורך כמה צעדי-זמן Δt(Learning to Simulate Complex Physics with Graph Networks, ICML 2020). בנוסף, צוותים אחרים הציגו פתרונות מבוססי רשתות קונבולוציה לבעיות דומות, תוך שימוש בייצוגים מבוססי חלקיקים או grid רגולרי (גריד המסודר בצורה רגולרית, הדומה לתמונה המיוצגת כמטריצה של פיקסלים שלכל אחד יש מספר קבוע של שכנים, למעט האיברים הקיצוניים).

היתרונות של שימוש בגריד רגולרי הם משמעותיים - הגריד מאפשר לנו להפעיל את כל הידע על רשתות קונבולוציה ולמנף הרבה קוד קיים גם עבור דאטה שאינו מורכב מתמונות, אך מסודר כגריד. עם זאת, לגריד רגולרי יש חיסרון אחד משמעותי בהקשר של סימולציות - המבנה שלו קבוע ולכן הוא לא מאפשר חישוב בדיוק גבוה יותר רק באזורים בהם הדבר נדרש, וההפך. לדוגמה, אפשר לחשוב על כנף של מטוס כאזור שסביבו נרצה לקבל דיוק מקסימלי, בעוד הרחק מהמטוס הדיוק כלל לא מעניין (ראה תרשים 1, המדגים את העיקרון על mesh לא רגולרי). הרגולריות של הגריד היא למעשה גם היתרון העיקרי שלו וגם החולשה העיקרית בהקשר הנוכחי.

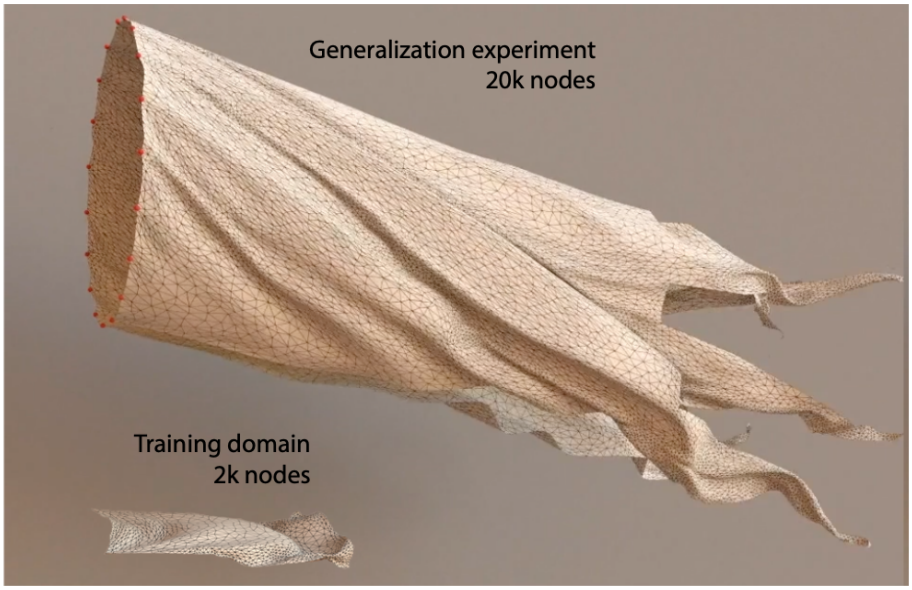


תרשים 1: דוגמה לחלוקה אדפטיבית של דומיין הסימולציה, מתוך המאמר.

לעומת זאת, ייצוג בתור mesh מאפשר לנו את האדפטיביות הזאת, בתנאי שנוכל להפעיל עליו כלים של למידה. מצד אחד, הוא ייצוג מאוד פופולרי בכלי סימולציה קלאסיים וקיים גוף עצום של ידע לגבי סימולציות על mesh, ומצד שני עולם הלמידה החישובית עוד לא התייחס אליו בצורה נרחבת עד כה. המאמר הנוכחי ממדל את דומיין הסימולציה כ-mesh תלת-מימדי או דו-מימדי ומציג תוצאות מרשימות מבחינת זמני ריצה, סקלביליות, יכולת הכללה וגמישות למילוי מגוון משימות. נראה שהבחירה במודל אדפטיבי השתלמה מכמה בחינות, כאשר הטרייד-אוף הוא כמובן מורכבות הייצוג. עם זאת, הרשתות המוצגות במאמר הן יחסית פשוטות ורצות במהירות, לפעמים כמעט בזמן אמת, וזאת תודות להימנעות מוחלטת של הכותבים משימוש בפעולות קונבולוציה יקרות.

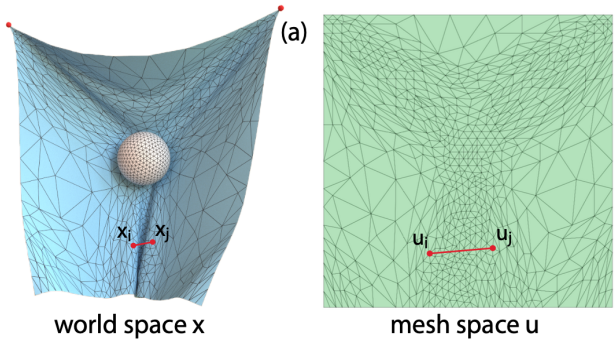
המאמר משיג את התוצאות האלה תוך שימוש בשלושה רעיונות מרכזיים:

1. **Relative Encoding**: הקלט למודל מורכב (בחלקו)מגרף אשר נבנה מה-mesh בקידוד יחסי: נגדיר וקטור $u_{ij} = u_i - u_j$ והנורמה שלו $|u_{ij}|$ כאשר u_i, u_j הם קודקודים כלשהם בגרף המיוצגים במרחב הגרף עצמו (כלומר, בבסיס הנפרש ע״י אלמנטים מהגרף, ראה תרשים 3). הדבר מאפשר למודל ללמוד מן המאפיינים הלוקאליים של גרף הקלט, להתחשב בכיוונים ואורכים יחסיים וכך לפתח את היכולת להכליל את הכללים הפיזיקליים שלמד לסימולציות מורכבות יותר. המאמר מציג את היכולת הזאת בניסוי שבו המודל לומד דינמיקה על חתיכת בד מרובעת שטוחה ובעלת מעט קודקודים, ומכליל עבור חתיכת בד מורכבת עם הבדל של סדר גודל בכמות הקודקודים (תרשים 2).



תרשים 2: דוגמה ליכולת ההכללה של המודל המאומן, מתוך המאמר.

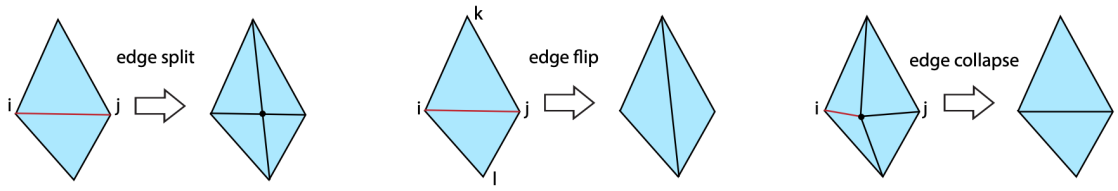
2. **World-space Edges**: העשרת הגרף בקשתות שאינן חלק מהגרף המקורי, ותפקידן לחבר בין קודקודים שרחוקים אחד מהשני ב-mesh-space אבל קרובים אחד לשני במרחב הסימולציה שבו נמצא ה-mesh עצמו. למרחב זה נקרא world-space. המוטיבציה להוספת הקשתות נובעת מכך שאנחנו מעוניינים שהמודל ילמד להתחשב בהשפעה הפיזיקלית של קודקודים אחד על השני, מבלי לדרוש מחזורי message passing רבים על מנת לפעפע את המידע בין הקודקודים. בנוסף, נחסך כך מהמפתח הצורך להגדיר ידנית קשרים בין קודקודים, דבר הדורש שימוש בידע על דומיין הבעיה הספציפית. כך מתקבלת ארכיטקטורה רובסטית יותר וגמישה יותר המסוגלת לעבוד על בעיות שונות. גם קשתות אלה מקודדות בקידוד היחסי, ונשמרות גם הנורמות בדומה לסעיף 1.



תרשים 3: הבדל בין קודקודים שקרובים ב-mesh-space וקודקודים שקרובים ב-world-space. למרות ששני הקודקודים המוצגים בחלק הימני רחוקים במרחב ה-mesh, הם קרובים במרחב העולם ולכן נוסיף קשת ביניהם. מתוך המאמר.

3. **<u>Learned Remeshing</u>**: זהו אולי הרעיון המרכזי במאמר והתרומה המרכזית שלו. בכל איטרציה של הסימולציה (הנקראת גם rollout) נבצע עידון והגסה של הגרף בהתאם למדד כלשהו שלוקח בחשבון את הכיוון היחסי של כל קשת ואת אורכה. מדד זה מחליט למעשה האם הקשת "ארוכה מדי" או "קצרה מספיק" בהתחשב במאפיינים הלוקאליים של הסביבה שלה. הרציונל כאן הוא לאפשר למודל לשלוט ברזולוציה הנדרשת בגרף בכל שלב ושלב בסימולציה. לדוגמה, נרצה שהוא ילמד להשקיע יותר משאבים באזורים עם curvature גבוה, או באזורי מגע בין עצמים שונים בסימולציה. החידוש במאמר הזה הוא ש-"מדד הרזולוציה" נלמד ביחד עם ייצוג המודל וייצוג הדינמיקה הפיזיקלית. בזמן הרצה על מידע חדש (למשל test set), המדד הזה משוערך ע"י המודל לפי הפרמטרים שהוא למד, ואז ניתן להשתמש בו בהרצת הסימולציה. בשיטה הזו, אלגוריתם ה-remeshing נשאר גנרי ולא כולל שום ידע מוקדם על הבעיה, והמודל לומד בעצמו איך להשתמש באלגוריתם הזה כדי לקבל את התוצאות הנדרשות.

נסביר בקצרה את נושא העידון וההגסה: כאשר נחליט שאזור מסוים בגרף זקוק לרזולוציה גבוהה יותר (כלומר שריג עדין יותר) נוכל להשיג זאת במגוון דרכים. אחת מהן היא פיצול קשת לארבע קשתות שונות ע"י הכנסת קודקוד למרכז האזור, וחיבורו אל ארבעת הקודקודים הקיימים (ישנן מגוון שיטות להכרעה היכן למקם את הקודקוד החדש). כך מתקבל עוד קודקוד שעליו נוכל לחשב את הגדלים המעניינים אותנו, ובמקביל, האזורים בהם ערכים מתקבלים מאינטרפולציה בלבד יהיו בעלי שטח קטן יותר. ניתן לראות דוגמה בחלק השמאלי של תרשים 4.
התהליך ההפוך גם קיים. כאשר נרצה להוריד את הרזולוציה, נוכל לבחור קודקוד ולהוציא אותו מהגרף. כעת נחבר בקשת שניים מארבעת הקודקודים שהתנתקו עקב השלב הקודם.
פעולה נוספת שאפשר לבצע היא היפוך של קשת על מנת לקבל שני אזורים יותר שווים בשטחם מאשר אם לא היינו מבצעים את ההיפוך. קריטריון זה נקרא קריטריון Delaunay (הסבר על הגירסה הפשוטה ביותר שלו ניתן למצוא כאן). קיימים כמובן מגוון של קריטריונים דומים אשר מתאימים לסיטואציות שונות והמאמר משתמש בקריטריון קצת יותר מורכב, אך דומה בבסיסו.



תרשים 4: עידון והגסה של אזור בגרף ע"י פיצול, איחוד והחלפה של קשתות בין קודקודים (מתוך המאמר)

תודה לאופיר יזרעאלב, מייק ארליכסון ואוהד עמוסי על הפידבק בהכנת הסקירה.

דאטהסטים: custom, הסבר נוסף לגבי הורדת הדאטה - בגיטהאב של המאמר.

לינק למאמר: https://arxiv.org/abs/2010.03409

לינק לקוד: פורסם חלקית
https://github.com/deepmind/deepmind-research/tree/master/meshgraphnets