סקירה זו היא חלק מפינה קבועה בה אני סוקר מאמרים חשובים בתחום ה-ML/DL, וכותב גרסה פשוטה וברורה יותר שלהם בעברית. במידה ותרצו לקרוא את המאמרים הנוספים שסיכמתי, אתם מוזמנים לבדוק את העמוד שמרכז אותם תחת השם deepnightlearners.

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

Regularizing Towards Permutation Invariance in Recurrent Models

---

פינת הסוקר:

המלצת קריאה ממייק: כמעט חובה (לא חייבים אך ממש מומלץ).

בהירות כתיבה: גבוהה.

רמת היכרות עם כלים מתמטיים וטכניקות של ML/DL הנדרשים להבנת מאמר: בינונית מינוס - צריך להבין מה זה RNN ותכונותיו הבסיסיות. בנוסף מומלץ לרענן את הידע הבסיסי בקומבינטוריקה (תמורות) ובתורת הקבוצות (מושגי יסוד).

יישומים פרקטיים אפשריים: ניתן להשתמש בטכניקה זו בשביל משימות עיבוד סדרות אינווריאנטיות (באופן מלא או חלקי) לסדר איבריהן כמו משימות זיהוי של עני נקודות,מציאת דמיון בין סטים של אובייקטים, זיהוי אותות ECC וכדומה.

פרטי מאמר:

לינק למאמר: זמין להורדה.

לינק לקוד: לא הצלחתי לאתר.

פורסם בתאריך: 25.12.20, בארקיב.

הוצג בכנס: NeurIPSi 2020.

תחומי מאמר:

- רשתות מסוג RNN.
- משימות אינווריאנטיות לסדר של קלט.

כלים מתמטיים, מושגים וסימונים:

- תמורה (פרמוטציה) של סדרת קלט (יסומן כ- p).

תמצית מאמר:

קיימות לא מעט בעיות בתחום למידת מכונה שהן אינווריאנטיות לסדר של הקלט, כלומר לא תלויות באיזה סדר אנו מכניסים את הקלט לרשת. בין משימות כאלו נמנות משימות כמו זיהוי סגמנטציה בענני נקודות במידול 3D, מציאת דמיון בין סטים של אובייקטים (למשל מציאה אוספי תמונות הכי דומים) ועוד. קיימות מספר גישות לבעיות מהסוג הזה בלמידה עמוקה. אחת הגישות היא בניית רשת נוירונים שהיא אינווריאנטית לסדר של קלט באופן אינהרנטי. אחת הדוגמאות לארכיטקטורות כאלו הינה "סט טרנספורמר" שבליבו נמצא מנגנון של "self-attention" המוכר לנו מהטרנספורמר הקלאסי המשמש כמעט בתור "ארכיטקטורת ברירת המחדל למשימות NLP". שימו לב הטרנספורמר המקורי אינו (!!) אינווריאנטי לתמורות של הקלט כי הוא מכיל רכיב קידוד המיקום (positional encoding). בנוסף הארכיטקטורה של הטרנספורמר מכילה כמה שכבות שלא מקיימות את תנאי האינווריאנטיות לתמורות כמו שכבת FC. רשתות נוירונים בסגנון RNN (והשכלולים שלו כמו LSTM, GRU וכדומה) המיועדות לעיבוד דאטה סדרתי כמו שפות טבעיות, אותות דיבור וסרטי וידאו (החלק הטמפורלי) כמובן אינן אינווריאנטיות לסדר של הקלט מהסיבה הפשוטה שיש להם "סדר טבעי אינהרנטי".

אז נשאלת השאלה איך נוכל לבנות ארכיטקטורת רשת שהיא אינווריאנטית לתמורות ולא בעלת סיבוכיות חישובית גבוהה (ריבועית ביחס לאורך הקלט) כמו הטרנספורמרים. זה תחום מחקר פעיל שהספיק להניב כמה תוצאות מעניינות. למשל מחברי DeepSets חקרו תכונות של פונקציות אינווריאנטיות לתמורות והוכיחו שכל פונקציה f כזו ניתן לתאר (למדל) עי" שתי רשתות R1 ו- R2 באופן הבא:

- עבור כל איבר בסדרת קלט **x** = (x_1,..., x_n) מחשבים את הפלט של R1(x_i).
- סוכמים את כל הערכים של R1(x_i) מהשלב הקודם
- מעבירים את הסכום הזה דרך רשת R2

דרך אגב, המאמר הנסקר מוכיח משפט מאוד מעניין הטוען שלכל אורך סדרה K גדול מ- 4, קיימות פונקציה אינווריאנטית לתמורות כאשר ניתן לממש אותה עם 3 נוירונים מוסתרים בלבד כאשר DeepSets צריך לפחות K נוירונים בשביל לממש אותה. כלומר, המחברים רומזים שהגישה של DeepSets עלולה להיות לא מאוד יעילה עבור משימות מסוימות (נכון שהמשפט בונה רק משימה אחת כזו אך לדעתי יש משפחה רחבה של משימות כאלו).

בעבודות אחרות הציעו למצע פלטים של הרשת עבור כל הפרמוטציות של הפלט. כמובן רשת כזו הינה אינווריאנטית לתמורות אך אינה ישימה בסקייל (עבור סדרות ארוכות) עקב סיבוכיותה המעריכית. כמו שכבר הזכרתי ארכיטקטורת self-attention הינה אינווריאנטית לתמורות אולם גם היא בעלת סיבוכיות ריבועית ביחס לאורך הקלט שגם מקשה על יישומה למשימות עם קלט ארוך.

המאמר מציע גישה אחרת שאומרת כך: [scroll_highlight]בואו ניקח רשת שהיא לא אינווריאנטית לפרמוטציות ונאלץ אותה להיות כזו בעזרת רגולריזציה[/scroll_highlight]. הרי אם ניקח רשת RNN ו"נאלץ" אותה כך להוציא אותו פלט זהה עבור כל פרמוטציה אפשרית של כל סדרות הקלט מהדאטהסט, אז האינטואיציה אומרת שאנו אמורים לקבל רשת "קרובה אינווריאנטיות לתמורות של הקלט". אבל צריך לזכור שדרך אימון כזו לא מבטיחה אינווריאנטיות לתמורות(!!) לא משנה עד כמה גדול סט האימון (פרט למקרה הלא מעניין שסט האימון מכיל את כל

הסדרות האפשריות עבור משימה זו). הסיבה היא, שלא ברור עד כמה אופן אימון כזה יודע "להכליל". כלומר עד כמה אינווריאנטיות לפרמוטציות על הדוגמאות מסט האימון מועברת גם לטסט סט. מדובר באמת בשאלה מאוד לא טריוויאלית.

המאמר מציע לקחת את הגישה הזו אבל [scroll_highlight]במקום "לאלץ" RNN להיות אינווריאנטית על כל הפרמוטציות של סדרת קלטים, המאמר מציע "לאלץ" אותו להיות אינווריאנטי על תת-קבוצה P_pr של הפרמוטציות[/scroll_highlight]. תת קבוצה זו מורכבת מכל התמורות ששוות לפרמוטציית זהות פרט לשני מיקומים. כלומר כל תמורה מ- P_pr היא למעשה שחלוף של שני איברים בסדרת קלט המקורית. קל לראות שאם הרשת אינווריאנטית על כל פרמוטציה p מ- P_pr לכל הקלטים אז היא אינווריאנטית לכל התמורות האפשריות של הקלט. השיטה המוצעת קיבלה SIRE - Subset Invariant REgulalizer.

הערה: הסיבה שהמחברים בחר בארכיטקטורה של RNN נובעת מהמבנה הייחודי שלו, כאשר המצב המוסתר (s_t+1 (hidden מוגדר כפונקציה של המצב s_t בזמן t והקלט x_t+1. זה מבנה מאוד נוח לבעיות אינווריאנטיות לתמורות (כמו מקסימום של סדרה) כי המצב s_t "יכול לצבור" את הסטטיסטיקה על הקלט עד זמן t בצורה אינווריאנטית.

הסבר של רעיונות בסיסיים:

בתכלס המאמר מציע להוסיף ללוס הרגיל של המשימה איבר רגולריזציה השווה להפרש הריבועי בין פלט הרשת, עבור הסדרה המקורית, לפלט הרשת עבור הקלט אחרי פרמוטציה מ P_pr. פרמוטציה זו נבחרת באקראי מכל הפרמוטציות האפשריות מ- P_pr. המאמר לא מפרט האם מוסיפים הפרש כזה עבור תמורה אחת בלבד או בוחרים כמה כאלו באקראי (אני חושב שמספר התמורות באיבר רגולריזציה צריך להיות אדפטיבי להיקבע ע"י היחס בין הלוס על המשימה וגודל הפרש ממוצע על כמה תמורות מ- P_pr).

הישגי מאמר: .

המאמר מראה ש- SIRE מפגין ביצועים יותר טובים מ- DeepSets במספר משימות (הם לא השוו את הביצועים על מול Set Transformer מבוסס על מנגנון attention כנראה בגלל הסיבוכיות הריבועית שלו).

1. חישוב של parity של סדרה (סכום מודולו 2).
2. חישוב של סכום, טווח (הפרש בין האיבר המקסימלי למינימלי) שונות של סדרה.
3. זיהוי אובייקטים וסיווג בענני נקודות.
4. חצי טווח של סדרה (הפרש של המקסימום של החצי הראשון של סדרה והמינימום של החצי השני שלה). משימה זו סמי-אינווריאנטית לתמורות (עבור סט מאוד גדול של תמורות אך לא עובר כולן, התוצאה לא משתנה.
5. משימת סיווג על Locally Perturbed MNIST, הנוצר מ- MNIST רגיל עי" שחלוף של פיקסלים שכנים רנדומליים. משימה זו כמובן סמי-אינווריאנטית לתמורות.

בכל המשימות הם הראו יתרון של SIRE על DeepSets פרט למשימה אחת עבור ענני נקודות. מעניין שעבור Locally Disturbed MNIST הם ניסו RNN רגיל שקיבל דיוק של 87% בזמן של CNN סטנדרטי הפגין דיוק של 96%. אולם, אחרי ההוספה של איבר הרגולריזציה שלהם (כנראה כן הם לא לקחו כל פרמוטציה אלא רק כאלו שמתאימות לאופן יצירת הדאטה סט) הדיוק הגיע ל- 97.7%.

נ.ב. מאמר מציע שיטה להתאים את RNN למשימות אינווריאנטיות לתמורות של קלט. השיטה מאוד אינטואיטיבית, קלה להבנה ומוסברת היטב - כיף לסקור כזה. אבל לי עלו כמה שאלות שלא מצאתי תשובה עליהן

במאמר. למשל לא ברור לי כמה תמורות מ- P_pr אני צריך בשביל לאמן סדרות עד גודל מסוים (נגיד אני רוצה לאמן רשת לחישוב טווח של סדרה עד גודל 100 - כמה אפוקים אני צריך בשביל להתכנס לתוצאה טובה). בנוסף מאוד מעניין לראות אנליזה של השפעת אורכי הסדרות שעליהם מאמנים RNN עם SIRE על הביצועים). המאמר גם הציג השוואה של השיטה שלהם מול גישה אחת בלבד DeepSets ובעיקר על בעיות צעצוע - הייתי רוצה לראות איך הביצועים של SIRE מול שיטות כמו transformer set ודומיה.

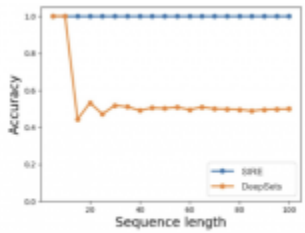

Figure 1: Test accuracy as a function of sequence length for learning parity, using DeepSets and RNNs.

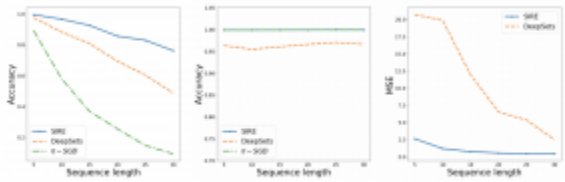

Figure 2: Test prediction accuracy (zero-one error) of *saw* (left) and *range* (center). For the *variance* experiment we report mean square error (as in Murphy et al. [2018]).

Method	100 pts	1000 pts	5000 pts
DeepSets	0.825	0.872	**0.90**
SIRE	**0.835**	**0.878**	0.899

Table 1: Point cloud classification results.

#deepnightlearners

הפוסט נכתב על ידי מיכאל (מייק) ארליכסון, PhD.

מיכאל עובד בחברת סייבר Salt Security בתור Principal Data Scientist. מיכאל חוקר ופועל בתחום הלמידה העמוקה, ולצד זאת מרצה ומנגיש את החומרים המדעיים לקהל הרחב.