סקירת המאמר:

LAMBDANETWORKS: MODELING LONG-RANGE INTERACTIONS WITHOUT ATTENTION

שיצא לפני כמה חודשים

הוצג בכנס: עוד לא פורסם (כרגע under review ב-ICLR2021).

תחומי מאמר:

זיהוי אובייקטים (object detection) בתמונה וסיווג תמונות (image classification) בעזרת רשת בעלת סיבוכיות נמוכה.

כלים מתמטיים, מושגים וסימונים:

טרנספורמרים בעלי סיבוכיות חישובית נמוכה למשימות סיווג בדומיין התמונות.

תמצית מאמר:

הסיבוכיות של מנגנון attention עבור כל מכפלה הינה O(MxN), כאשר M,N הם הסדרה שעליה עושים את ה-attention, וה-query. במקרה של self-attention הסיבוכיות הינה O(N^2) כאשר N הוא אורך הסדרה. כאשר מדובר במשימות NLP, אז הסיבוכיות אינה גדולה – גם עבור טקטסים ארוכים עדיין מדובר במשימות שלא דורשות יותר מדי כח חישוב וזיכרון. אך כשמדובר על תמונה אז הסיבוכיות נהיית ענקית – אם התמונה היא בגודל WxH, אז הסיבוכיות הינה O((WxH)^2), ובתמונות ברזולוציה גבוהה זה יכול להגיע לסיבוכיות של O(256^4), כיוון שאורך הסדרה המתקבלת מתמונה בגודל 256x256 הינה 256^2.

במאמר הזה מזהים את הנקודה העיקרית המשפיעה על הסיבוכיות, ומנסים לעקוף אותה בלי לפגוע בביצועים. במילים אחרות – מנסים למצוא דרך מהירה ויעילה למצוא קונטקסט בין איברים בסדרה (למשל: למצוא קשר בין פיקסלים בתמונה וכך לזהות אובייקטים), ולמצוא להם משמעות (למשל: לסווג אובייקטים ל-labels). כיוון שמנגנון ה-attention הקלאסי הוא בעל סיבוכיות גדולה ולא פרקטית במקרים מסוימים, צריך למצוא דרך לשפר את הסיבוכיות.

דרך אחת אפשרית היא local attention – במקום להסתכל על תמונה שלמה, ניתן להסתכל על אזור מסוים בתמונה, ועליו לבצע את ה-attention. הסיבוכיות היא עדיין O(N^4), אך כעת N הוא לא גדול אלא יחסית קטן, באופן כזה שמאפשר בצורה די מהירה לבצע את החישובים הדרושים.

במאמר הזה מציעים דרך אחרת שאינה מוגבלת לאזור מסוימת. בכדי להבין את הפתרון, יש לזהות את הנקודה הבעייתית – בה נמצא עיקר העומס. כאשר מחפשים ייצוג חדש הקושר בין סדרה מסוימת (נקרא לה מעכשיו context, ונסמן ב-C) לבין query מסוים (נסמן ב-Q), יש לבצע מכפלה בין כל איבר ב-C לבין כל איבר ב-Q. זה מה שקורה ב-attention הרגיל. כדי להימנע מזה, המחברים מציעים לבצע על C מניפולציה מסוימת, בכדי שהכפל CxQ יהיה הרבה יותר יעיל. בפועל מה שעושים זה מעבירים את C דרך שכבת lambda. שכבת lambda הינה פונקציה לינארית כלשהיא (נלמדת כמובן) המהווה ייצוג ממימד נמוך יותר של איבר ב-C. כלומר, לוקחים חלון מסוים מתוך תמונה (נניח d^2 פיקסלים), ומוצאים לו ייצוג פשוט יותר, ואת אותו ייצוג מכפילים ב-Q. באופן הזה הסיבוכיות ירדה, כיוון שחילקנו את C לחלקים, וכל אחד מהם ייצגנו בצורה חדשה פשוטה יותר, ורק אז ביצענו את המכפלה ב-Q.

הסבר של רעיונות בסיסיים:

נזכיר תחילה שב-attention רגיל בכל פעם אנו לוקחים איבר אחד מסדרת ה-input ומבצעים מכפלה פנימית עם שאר האיברים בסדרה אחרת (query) על מנת לקבל סדרה חדשה, בה כל איבר מייצג איבר בסדרה המקורית, כאשר בייצוג החדש כלול גם מידע על הקשרים שבין האיברים בסדרה המקורית. אנו רוצים לקחת את המכפלה הזו ולשפר את הסיבוכיות שלה בלי לפגוע באיכות התוצאה.

יותר בפירוט: ב-attention, ה-C מחולק ל-key, value. ה-key מוכפל ב-Q, עובר SoftMax על מנת לקבל ייצוג חדש, ואז מוכפל ב-value.

בשיטה המוצעת, ה-key עובר SoftMax, מוכפל ב-value, ואז עובר סכימה וכך מיוצג באופן חדש ופשוט יותר, הנקרא lambda_c. במקביל, מייצרים מטריצה הנקראת E, ומכפילים אותה ב-value, ומקבלים את lambda_p. שני איברי ה-lambda נסכמים לאיבר אחד. בשלב זה מתבצעת המכפלה בין ה-Q לבין lambda, על מנת לקבל ייצוג מחודש של הסדרה המקורית, המכיל קשרים בין האיברים. בניגוד ל-attention הרגיל, כאן ה-lambda הוא בעל מימדים קטנים

הרבה יותר מה-key, ולכן הסיבוכיות יורדת באופן משמעותי. החיסכון המשמעותי נובע מכך שנמנענו מהמכפלה בין כל ה-C לבין כל ה-Q, על ידי מכפלות קטנות יותר בשלבים אחרים של התהליך.

תוצאות המאמר:

בשלב הניסויים באו לבחון מספר דברים – האם הצליחו לקבל את אותן תוצאות כמו של SOTA algorithms ולקבל שיפור ב: מהירות החישוב, מספר הפרמטרים הדרוש, סיבוכיות החישוב וגודל הזיכרון הדרוש.

ראשית השוו בין attention רגיל לבין שימוש בlambda layers ביחס ל-ImageNet, כאשר לוקחים ResNet50 בתור הארכיטקטורה. הראו שיפור ניכר בכמות הפרמטרים הדרוש (כחצי מה-attention הרגיל), אבל עיקר השיפור נמצא בזיכרון הדרוש לביצוע החישובים. עבור תמונות בגודל 224x224, global self-attention צרך 120GB של זיכרון, בעוד השימוש בלמדה הוריד את הזיכרון הדרוש לפחות מ-1G.

לאחר מכן השוו עבור דאטה סט COCO במגוון משימות, והראו כיצד השיגו אותן תוצאות כמו EfficientNet במהירות משמעותית יותר טובה – פי 4.5. גם כמות הפרמטרים הדרושה הייתה קטנה יותר. עוד צוין כי בניגוד ל-EfficientNet בו השתמשו באוגמנטציות וב-dropout probabilities, הרשת של lambda השתמשה באופן עקבי באותם פרמטרים, מה שמפשט את הרשת.

הישגי מאמר: המאמר מציג תוצאות של SOTA על COCO במשימת של vision, כאשר הצליחו לשפר את הזיכרון הדרוש, כמות הפרמטרים, הסיבוכיות וזמן החישוב ביחס לאלגוריתמים קודמים (לא הכל בבת אחת, אלא וריאציות שונות של ה-lambda layers השיגו שיפורים בפרמטרים שונים). הרעיון לא טריוויאלי ולא קל למימוש והבנה, אבל נראה שעובד טוב.

לינק למאמר: https://openreview.net/pdf?id=xTJEN-ggl1b

ניתן לקרוא גם ביקורות של reviewers כאן:

https://openreview.net/forum?id=xTJEN-ggl1b

לינק לקוד: https://github.com/lucidrains/lambda-networks

נ.ב. מאמר מעניין, שהכותבים שלו עדיין אנונימיים. המאמר קשה לקריאה, עמוס במשוואות וכתוב בצורה קצת מסורבלת, אבל עדיין נותן תוצאות יפות ומעוררות עניין. ניסיתי להימנע מלהיכנס לעומק הפרטים והמשוואות המתמטיות, ולתת את הרעיון המרכזי בצורה פשוטה ולא מסובכת. אשמח לתת הרחבה למי שמעוניין.

#deepnightlearners