# Review 64b: TRAIN SHORT, TEST LONG: ATTENTION WITH LINEAR BIASES ENABLES INPUT LENGTH EXTRAPOLATION

**Paper: https://arxiv.org/abs/2108.12409v2**

כמו שאתם יודעים אחת החולשות העיקריות של הטרנספורמר היא הסיבוכיות הריבועית שלו ביחס לאורך סדרת הקלט. יצאו מאות מאמרים המציעים וריאנטים של הטרנספורמר עם סיבוכיות נמוכה יותר.

המאמר הזה לוקח כיוון די מפתיע בהתמודדות עם סוגיית הסיבוכיות הריבועית. הוא שואל את השאלה הבא - אולי נאמן טרנספורמר לאורכי קלט לא גדולים ואז נריץ אותו עם קלטים ארוכים יותר באינפרנס? זה נשמע מגניב אבל

האם זה בכלל יעבוד? באופן לא מפתיע התשובה טמונה בקידודים מיקומיים (positional encodings -PS) שמשתמשים בהם בשביל להעביר לטרנספורמר מידע עם מיקום של טוקנים בסדרה.

מתברר שלאחרונה יצאו מספר מחקרים המציעים PS שונים לשיפור ביצועי טרנספורמר (ולאו דווקא למטרה המתוארת לעיל). אז המאמר הזה מציע שני שינויים ל- PS המאפשרים לטענות ״לאמן קצר ולהריץ ארוך״. במקום לחבר את PS לייצוג הטוקנים כמו שעשו בטרנספורמר האמיתי, המחברים כאן מציעים לעשות את הדבר הבא:

מחשבים מכפלה פנימית של וקטורי query ו- key.

מחברים למכפלה הזו את המרחק השלילי בין הטוקנים, מוכפל בקבוע m (שהוא שונה בכל head של הטרנספורמר). למשל למכפלה פנימית של טוקן 3 ו- 8 מחברים (5m-).

ובונים מכפלה בין וקטורי value עם המקדמים שחישבנו בסעיפי 1-2 (למעשה השינוי הזה הוצע במאמר אחר). ההגיון כאן שכל המידע על מיקום כבר מקודד ב- key ו-query ואין צורך להוסיף אותו ל- value.

אינטואיציה: למעשה הקידוד הזה אומר את הדבר הבא: בוא נקטין את הקשר (מקדם רלוונטיות) בין טוקנים ככל שמרחק בין הטוקנים גדל. המקדם m השונה בין ראשי הטרנספורמר מאפשר לשלוט במידת ״ההקטנה״ של הרלוונטיות.

וזה כל הסיפור. שימו לב שהמאמר מציע את ה- PS שלהם רק עבור טרנספורמר אוטורגרסיבי (שנקרא decoder) אבל אני לא רואה שום סיבה לא להשתמש בזה בצורה דו-כיוונית.

ה- PS המוצע מאפשר להאט את הירידה בביצועים בתרחיש שבו מאמנים טרנספורמר לאורך קלט נתון (512) ואז עושים אינפרנס לאורכי קלט כמו 1024, 2048 וארוכים יותר (יחסית ל- PS האחרים כולל המקוריים של הטרנספורמר).

מאמר: https://arxiv.org/pdf/2108.12409.pdf

יניק: https://www.youtube.com/watch?v=-Kgxv64aG3o