# Review 193: Retentive Network: A Successor to Transformer for Large Language Models

**Paper: https://arxiv.org/abs/2307.08621v4**

https://arxiv.org/abs/2307.08621

זה הולכת להיות הסקירה הקלה ביותר (אך קצת ארוכה). המאמר משתמש באופן די אלגנטי ברעיונות שהוצע ב 8 המאמרים שכבר סקרנו. אזכיר שהמכנה המשותף במאמרים שסקרנו היתה מטרה למצוא ארכיטקטורה בעלת דואליות הבאה:

⬅1. ניתנת לאימון באופן מקבילי כמו הטרנספורמרים

⬅2. היסק (inference) מהיר (=לינארי במונחי אורך חלון הקשר) שלא מצריך התחשבות מפורשת בכל טוקני של חלון ההקשר.

הארכיטקטורה שהמאמר מציע היא אכן מבורכת בדואליות זאת ובאותו הזמן היא מאוד פשוטה וקלה להסבר (ככה נראה לי). אתם בטח זוכרים את הייצוג הקונבולוציוני של (SSM (state-space model עבור ייצוג הזיכרון של סדרת טוקנים?

אם לא אזכיר בקצרה. עבור סדרת טוקנים נתונה יש לנו מערכת דינמית לינארית (DMS) שבאמצעותה אנו מייצגים בצורה איטרטיבית את זיכרון s\_n הנצבר ב n הטוקנים הראשונים בסדרה. בעזרת DMS ניתן לחשב את s\_n מייצוג הזיכרון קודם {s\_{n-1 ומייצוג של טוקן ה-n, מסומן v\_n.

לאחר מכן באמצעות וקטור s\_n אנו ממדלים פלט המודל o\_n עבור טוקן n (= ייצוג תלוי הקשר או contextualized embedding של טוקן n) דרך הטלתו עם מטריצה Q. נציין כי DMS מגדירה את מעבר(הלינארי) בין ייצוג של הזיכרונות n-1 ו-n מאפשר חיזוי במקבילי עבור כמה טוקנים במהלך אימון.

אותה DMS מוגדרת באמצעות מטריצות A ו-K וכאמור הפלט o\_n מוגדר באמצעות מטריצה הטלה Q. מטריצות Q ו- K הן אלו שנקראות בטרנספורמר מטריצות שאילתה וערך ומחושבות באותה צורה: Q = X\*W\_Q, K = X\* W\_K, כאשר X הוא ייצוגי הטוקנים.

עכשיו השאלה איך אנו מגדירים חישוב מקבילי של o\_n עבור כמה n? הרי עבור n גדול מספיק העלאה של מטריצה A בחזקה עלולה להיות יקרה גם מבחינת זיכרון וגם מבחינת מאשבי חישוב. אז פותחים אחד הפרקים הראשונים של ספר של אלגברה לינארית ומגלים שניתן לתאר מטריצות ריבועיות (לא כולן!) בתור A=L\*D\*L^{-1} כאשר D היא אלכסונית עם ערכים מרוכבים lambda\_j \* exp(i\*theta\_j), j=1,....,d.

מה בעצם טוב בייצוג הנחמד הזה? זה מאפשר לנו להעלות את מטריצה A בחזקה והבעיה שלנו עם חישוב A^n נראית פתורה. המאמר גם מניח ש lamda\_j = lambda, j=1,..,d וזה מאפשר את הייצוג הבא של המודל שהם מציעים:

למעשה המחברים מחליפים את מנגנון ה-attention הממומש עם סופטמקס בטרנספורמרים עם ה-attention הדועך בצורה מעריכית כפונקציה של בין הטוקנים. חדי העין שקראו את הסקירה הקודמת שלי ישימו לב שעיקרון דומה ממומש גם ב-RWKV אבל די מיצוע מעריכי של המידע מהטוקן הקודם. וכמובן ייצוג כזה חישוב מהיר עבור כל טוקן במהלך היסק (שזה תכונה 2 שלנו).

המאמר מציע שני שכלולים נחמדים ל-RetNet. הראשון הוא כדי להאיץ את מהירות האימון עוד יותר ולנצל את משאבי החישוב הזמינים ניתן לחלק את הטוקן לצ'אנקים ולהפעיל חישוב מקבילי בתוך כל צ'אנק וחישוב איטרטיבי בין צ'אנקים.

שכלול נוסף הוא שימוש במקדמי gamma שונים ל״ראשים״ (heads) שונים של RetNet. זה למעשה מקנה למודל יכולת יותר להתמקד בטוקנים קרובים יותר (lambda גבוה) ו״לפזר״ את ה-attention גם טוקנים רחוקים(lambda נמוך). שילוב של ראשים בעלי lambdas שונים ״לחקות״ את הטרנספורמר (לפחות במידה מסוימת).

https://arxiv.org/abs/2307.08621