# Review 192: RWKV: Reinventing RNNs for the Transformer Era

**Paper: https://arxiv.org/abs/2305.13048v2**

https://arxiv.org/abs/2305.13048

אוקיי, אחרי כמה מאמרים כבדים הפעם יש לנו מאמר קליל יחסית. אתם אולי זוכרים שהמאמר השלישי שסקרנו בסדרה (״Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention״) הראה שטרנספורמר עם attention לינארי ניתן לייצג בתור RNN מצד אחד (כלומר ניתן להפעלה באופן איטרטיבי כאשר הוא דוחס את הטוקנים הקודמים בוקטור זיכרון אחד) ומצד שני ניתן להפעלה באופן כמו הטרנספורמר מן המניין. כלומר יש בו את הדואליות שרצינו: חיזוי מקבילי של טוקנים ממוסכים במהלך האימון וחיזוי טוקנים בעל סיבוכיות לינארית במהלך ההיסק (inference).

המאמר שנסקור היום מקרב את הטרנספורמר ו-RNN באופן מפורש אפילו קצת יותר. המחברים לוקחים טרנספורמר עם מנגנון ״attention״ ״פשוט יותר״ ומוסיפים קצת RNN לאופן בו מחושבים מטריצות מפתח K ומטריצת ערך V. אבל קודם אספק לכם כמה פרטים על מנגנון "attention" שלקחו המחברים בתור בסיס ולמה אני שם אותו כאן בגרשיים. אז מנגנון הזה נלקח מהמאמר AFT) An Attention Free Transformer) שלפי שמו נראה שהמאמר מציע טרנספורמר ללא attention כלל!

אוקי, אז מה הסיפור של AFT ומה זה בכלל טרנספורמר ללא attention (לי זה נשמע על ההתחלה כמו אוטו ללא מנוע). AFT מחליף את המנגנון הרגיל של חישוב attention של הטרנספורמר בכזה שדורש משמעותית פחות זיכרון מהטרנספורמר הרגיל (בגרסתו הפשוטה גם סיבוכיות חישובית מוקטנת עד כדי לינארית במונחי אורך הקלט) ועושה את זה בדרך מאוד הגיונית. AFT מחליף את המכפלות הפנימיות בין וקטור שאילתה q\_i ווקטור המפתח k\_j באקספוננט של סופטמקס (שזה הלב של המנגנון והסיבה לסיבוכיות הריבועיות) בסכום של וקטורי המפתח עם מטריצת משקלים נלמדת w\_ij (מנורמל). כלומר לא מתחשבים בוקטורי שאילתה q\_i אלא משתמשים במקדמים קבועים ומחושבים על סמך סט האימון. לאחר מכן בונים צירוף לינארי עם וקטור הערך v כמו בטרנספורמר הרגיל.

כלומר מקדמי ה-attention בין טוקן i לטוקן j לא תלויים באופן מפורש בייצוג טוקן i אלא רק ב- i ו-j. בחירות חכמות (פרמטריזציה) של w\_ij מאפשרות להקטין את דרישות זיכרון והסיבוכיות החישובים כאשר המחיר הוא כמובן expressiveness של המודל. אחת הבחירות של w\_ij היא פונקצית דועכת מעריכית כאשר הארגומנט הוא מרחק בין הטוקנים (המאמר המסוקר משתמש בה).

אוקיי, אז איך מלבישים על זה RNN? לוקחים את המנגנון ה- attention מהפסקה הקודמת עם שפצור קל ליציבות נומרית - הוספה של וקטור u (המנגנון הנקרא wkv) ומפעילים אותו עם וקטורי מפתח וערך K ו- V מחושבים כמו ב-RNN. כלומר בונים וקטורים אלו( K ו- V) תלוים באופן מפורש בייצוג הטוקן הנוכחי וגם בייצוג הטוקן הקודם(זה כל הקטע). במקום להכפיל את ייצוג הטוקן במטריצות W\_k ו- W\_v (כמו בטרנספומר הרגיל) מכפילים אותם בסכום ממושקל (עם משקלים נלמדים) של ייצוג הטוקן x\_t הנוכחי וייצוג בטוקן הקודם {x\_{t-1. בנוסף מחשבים וקטורי r (הנקרא receptaince) באותה הצורה (עם x\_t ו- {x\_{t-1 ומטריצת W\_r). וקטורי r למעשה משמשים לנו כדי ״לשערך״ עד כמה אנו צריכים להתחשב בה (מחושבת עם הסיגמואיד כמו בזמנים הטובים ב-RNN). כל הסיפור הזה נקרא באופן לא מפתיע rkwv.

בסוף משלבים את התוצאה של rwkv עם וקטורי מפתח וערך המחושבים באותה צורה כמו ב-rwkv (התחשבות ב- x\_t ו- {x\_{t-1 אבל עם מטריצות הטלה נלמדות אחרות). איך משלבים? כרגיל בצורה של ResNet.

וזהו זה. שמח לבשר שהמאמר הבא שנסקור בדרך לממבה גם יהיה קליל (Retentive Network).