SiMBA: Simplified Mamba-based Architecture for Vision and Multivariate Time series

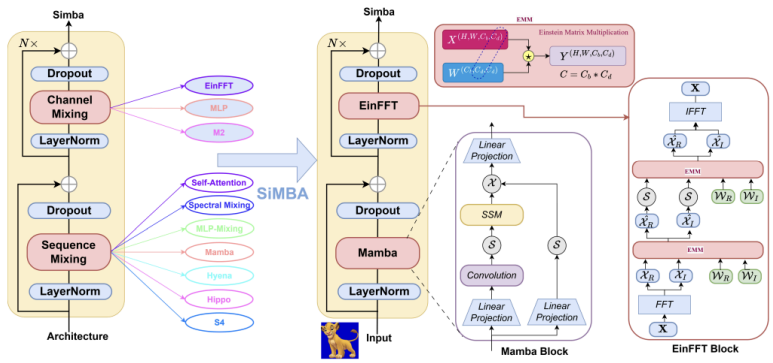

Fig. 1: Simplified Mamba Based Architecture.

המאמר הזה משך את תשומת ליבי כי מצד אחד יש בו שימוש נרחב בהתמרת פורייה ובייצוגים של דאטה בתחום התדר. החולשה שלי לתחום התדר נובעת מכך שביליתי כמה מהשנים הראשונות של הקריירה בתחום עיבוד של אותות אלחוטיות. מצד שני המאמר גם משתמש בארכיטקטורת ממבה שסקרתי בהרחבה בחודשים האחרונים (וכנראה אמשיך עם זה כי מאמרים מעניינים בנושא מרתק זה לא מפסיקים להגיע).

אוקיי, אז מה יש לנו במאמר הזה? המאמר מציע שדרוג יפה לארכיטקטורה של ממבה המערב כאמור התמרות פורייה וקצת משחקים בתחום התדר. הארכיטקטורה המוצעת מתאימה גם לדאטה ויזואלי וגם לסדרות זמן multivariate. המאמר כתוב בצורה די מסורבלת והיה לי לא טריוויאלי לגלות מה הם באמת עשו עקב הסברים וסימונים לא ברורים. אבל כאמור הרעיון מאחורי המאמר הוא די חמוד.

המחברים מנסים לשפר את ממבה על ידי הוספת שכבה שבגדול לוקחת את הייצוגים המופקים על ידי ממבה ו"מחזקת" אותם על ידי פלטור תדרים מסוימים מהם (הייצוגים). קודם כל נציין שמפעילים את המנגנון המוצע, שקיבל שם EinFFT, על כל איבר סדרה בנפרד (פאץ' של תמונה) בצורה ממוקבלת. כאמור הסיפור מתחיל מהפעלת התמרת פורייה על הפלט(=ייצוג פיסת דאטה) של שכבת ממבה. ואז המאמר הופך להיות די לא ברור והדבר הזה גזל ממני בערך שעותיים כדי להבין שלא אני מפספס משהו אלא המאמר עצמו קצת לא מדייק (בתקווה עמדתי במשימה זו).

כאמור הרעיון הוא מפלטר תדרים(תלויות) הלא נחוצים (לביצוע המשימה) בייצוגי איברי הסדרה. הפלטור מתבצע במחרב הייצוג של הדאטה (כלומר אמבדינג) ונקרא channel-mixing. כלומר שכבה זו היא משמשת בתור תוספת/החלפה ל-MLP שלפעמים משמשת לאותה המטרה.

אבל איך הוא עושה את זה משתנה בין נוסחה לנוסחה במאמר. במאמר עצמו (נוסחה 4) קודם כל מפעילים שכבה לינארית במישור המרוכב ולאחריה סיגמויד (גם במישור המרוכב). ב-appendix (בתחילת עמוד 22) זה כבר מופיעה שכבה אחת של ReLU, לאחר מכן עוד שכבה לינארית, לאחר מכן מפעילים פונקצית softshrink שמטרתה היא לאפס תדרים סביב אפס באינטרבל באורך lambda*2 ולהזיז כאלו מעבר לזה ב-lambda. כלומר איזה stop-band filter מוזז.

הגרסה השלישית מגיע מהדף האחרון של ה-appendix ששם יש רק ReLUs. לא הסתכלתי בקוד אז לא ברור מה באמת קורה שם. כל הפעולות האלו מתבצעות בצורה נפרדת במישור הממשי ובמישור המדומה שלאחר מכן משלבים אותם. בשלב האחרון מבצעים התמרת פורייה הפוכה (IFFT).

אוקיי, אז בואו נחזור לעיקר. המנגנון שבא אחרי שכבת ממבה נקרא EinFFT וכבר הבנו ש- FFT מתאים להתמרת פוריה. אבל מה זה Ein? באופן לא מפתיע אלו 3 האותיות הראשונות נלקחו מאינשטיין. אז מה בעצם איינשטיין עושה כאן?

למעשה המאמר משתמש בסכימת איינשטיין שהיא דרך לרשום מכפלות הטנזורים או המטריצות במקרה פרטי. למעשה במקום לרשום כל איבר ij של המכפלת מטריצות A ו- B בתור מכפלה פנימית של שורה i ועמודה j סכימת איינשטיין כותבת אותו ללא סימן של סכום(=סיגמה) אלא על ידי ציון של מספר שורה i, מספר עמודה j ואינדקס סכימה k.

אז איך המאמר משתמש בסכימה הזו? הרי אמרתי שהסכימה הזו מוגדרת גם לטנזורים ומתברר שלחבילות תוכנה כמו pytorch יש חבילות שיודעות לבצע מכפלת טנזורים רב מימדיים המבוטאים דרך סכימת איינשטיין בצורה די יעילה. וזה בדיוק מה שעושים במאמר. המאמר מפרק את המטריצות מהשכבות הלינאריות של EinFFT לכמה מטריצות במימד נמוך יותר ובונה מזה טנזור רב מימדי הבנוי ממטריצות בלוקיות (אפסים מחוץ לבלוקים). הטענה במאמר שזה מאפשר לבצע את המכפלות (באימון אבל כנראה גם באינפרנס) בצורה מהירה יותר על ידי ניצול טוב יותר של משאבי החומרה.

הביצועים כרגיל מפתיעים לטובה....

$$
\begin{aligned}
X &= \mathrm{FFT}(x) \\
X_{\mathrm{real}_1} &= \max(\mathrm{Re}(X) \cdot W_{1,\mathrm{real}} - \mathrm{Im}(X) \cdot W_{1,\mathrm{imag}} + B_{1,\mathrm{real}}, 0) \\
X_{\mathrm{imag}_1} &= \max(\mathrm{Re}(X) \cdot W_{1,\mathrm{imag}} + \mathrm{Im}(X) \cdot W_{1,\mathrm{real}} + B_{1,\mathrm{imag}}, 0) \\
X_{\mathrm{real}_2} &= \mathrm{Re}(X_{\mathrm{real}_1} \cdot W_{2,\mathrm{real}} - X_{\mathrm{imag}_1} \cdot W_{2,\mathrm{imag}} + B_{2,\mathrm{real}}) \\
X_{\mathrm{imag}_2} &= \mathrm{Im}(X_{\mathrm{real}_1} \cdot W_{2,\mathrm{imag}} + X_{\mathrm{imag}_1} \cdot W_{2,\mathrm{real}} + B_{2,\mathrm{imag}}) \\
X_{\mathrm{shrink}} &= \mathrm{softshrink}(X_{\mathrm{real}_2} \,\&\, X_{\mathrm{imag}_2}, \lambda = \mathrm{sparsity\_threshold}) \\
x_{\mathrm{ifft}} &= \mathrm{IFFT}(X_{\mathrm{shrink}}, \mathrm{dim} = 1, \mathrm{norm} =' ortho') \\
x_{\mathrm{reshaped}} &= \mathrm{reshape}(x_{\mathrm{ifft}}, (B, N, C))
\end{aligned}
$$