

Review 181, Short: Smooth Diffusion: Crafting Smooth Latent Spaces in Diffusion Models

<https://huggingface.co/papers/2312.04410>

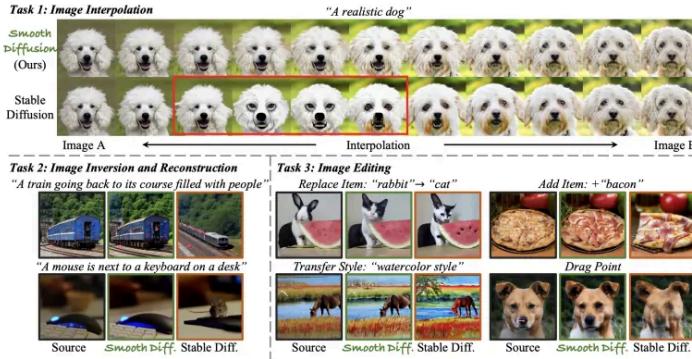


Figure 1. Smooth Diffusion for downstream image synthesis tasks. Our method formally introduces latent space smoothness in diffusion models like Stable Diffusion [59]. This smoothness dramatically aids various tasks in: 1) improving continuity of transitions in image interpolation, 2) reducing approximation errors in image inversion, & 3) better preserving unedited contents in image editing.

בסקיירטנו היום נדבר איך אנחנו יכולים "לסדר" את המרחב הלטנטי של מודלי דיפוזיה גנרטיביים. המאמר מנסה "לסדר" את המרחב הלטנטי של מודלי דיפוזיה. בשביל להבין מה זה המרחב הלטנטי של מודלי דיפוזיה הוא למעשה מרחב של וקטורים אסויים סטנדרטיים שהמייד שלהם שווה למועד שאנקודר מוקוד כל תמונה אליו. למעשה אין מדובר כי מודלי דיפוזיה לטנטים מייצרים ייצוג לטנטי של תמונה על ידי נקיי הדרגת (באייטרציות) של הרעש מוקטור גאוס סטנדרטרי (backward process).

לאחר השלמת התהילר מעבירים את הוקטור שנוצר דרך רשת הדקודר לייצרת תמונה. המאמר מנסה לגרום לכך שניינו קטן בוקטור הלטנטי שנוצר על ידי מודל דיפוזיה יוביל לשינוי קטן בתמונה הנוצרת. זה חשוב כי זה נותן לנו אפשרות לשנות בצורה יותר טובה במה אנחנו מייצרים עם המודל וגם מאפשר לנו ליצור "מעברים חלקים" בין התמונות השונות. אז מה הם עושים?

בגדי הרעיון שהזזה של הוקטור הלטנטי למפרק d תרגום להזזה שהיא לכל היוטר cd בתמונה שנוצרת ממנו כאשר c הוא קבוע (לא תלוי בתמונה). מכיוון שקשה לכפות את זה באופן ישיר במלול המודול המאמר בחר להשתמש בטכניקה נפוצה של רגולרייזציה מעולם הגאנים (GANs). ניתן להראות כי פונקציית לוס האוכפת יעקוביאן (מטריצה נגזרות) ביחס לוקטור לטנטי מוכפל בשינוי בתמונה הנוצרת(αd^*) להיות קבוע משיגה את המטרה המיוחלת.

למעשה זה קירוב טילור מסדר ראשון של התמונה הנוצרת על ידי הזזה של וקטור לטנטי. מעשיית מוסףים איבר לפונקציית לוס הרגילה של מודל דיפוזיה שكونו על אי התאמת αd^* למוצע המערבי שלו על פני האיטרציות הקודמות של gradient descent . זה נשמע קצת לא פשוט אבל הנוסחאות במאמר לא מורכבות יותר מדי...

Review 182: Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models

<https://arxiv.org/abs/2312.06585>

Algorithm 1: ReST (Expectation-Maximization). Given a initial policy (e.g., pre-trained LM), ReST^{EM} iteratively applies Generate and Improve steps to update the policy.

```

Input:  $\mathcal{D}$ : Training dataset,  $\mathcal{D}_{val}$ : Validation dataset,  $\mathcal{L}(x, y; \theta)$ : loss,  $r(x, y)$ : Non-negative reward function,  $I$ : number of iterations,  $N$ : number of samples per context
for  $i = 1$  to  $I$  do
    // Generate (E-step)
    Generate dataset  $\mathcal{D}_i$  by sampling:  $\mathcal{D}_i = \{ (x^j, y^j) \}_{j=1}^N$  s.t.  $x^j \sim \mathcal{D}$ ,  $y^j \sim p_\theta(y|x^j)$ 
    Annotate  $\mathcal{D}_i$  with the reward  $r(x, y)$ .
    // Improve (M-step)
    while reward improves on  $\mathcal{D}_{val}$  do
        | Optimise  $\theta$  to maximize objective:  $J(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}_i} [r(x, y) \log p_\theta(y|x)]$ 
    end
end
Output: Policy  $p_\theta$ 

```

היום אנו מדברים על אימון של מודלי שפה. באחת הסקירות האחרונות הסברתי לכם איך מאומנים מודל שפה עם RLHF (למידה עם חיזוקים המשולבת עם משוב אונשי) ולמה זה נכון. כדאי לך לקרוא כדי להבין להשתמש בטכניקות של RL כדי לאמן מודל בתנואה כלומר לעדכן משקלים של המודל על הדadata שנוצר אחרי העדכן האחרון של המודל. במנוחה RL הדadata נדגם לפי-policy הci מעודכן (כלומר למידה on-policy).

אימון מודלי שפה עם RLHF לצורך policy-on הוא יקר (כל הזמן צריך ליצור דadata) ולא תמיד יציב ולכן הוציאו מספר שיטות חלופיות פחות כבדות כמו DPO ו-REST. הרעיון ב-REST הוא לא ליצור דadata חדש לצורך policy-on אלא:

- לייצור דadataסט באמצעות מודל התחלתי ולשלב אותו עם דadata המתואג על ידי בני אדם
- לבחור את הדadata בעלת ערכי פונקציית תגםול גבוהה מעל סף התחלתי
- לאמן את המודל עם הדadata זהה
- לייצור עוד דadata עם המודל המעודכן (כל איטרציות אחרת לדלג על השלב
- לייצור (לسان) דadata עם ערך(תגםול) גדול מערך סף גבוהה יותר
- לאמן מודל עם דadata חדש...

המחברים לקחו את הרעיון הזה וscalלו אותו (מבחינת הביצועים) בהתקבוס על הרעיון של Expectation-Maximization (EM) ולייתר שימוש בדadata הנוצר על ידי בני אדם. הרעיון ב-EM הוא למוקסם נראות מירבית של פונקציית הסתברות ק ביחס לפרמטרים כאשר הדadata נדגם מהתפלגות אחרת q. זה מורכב משני שלבים איטרטיביים:

- E: מקרבים את q מבחינת KL Divergence (אופטימיזציה)
- M: ממוקסים את נראות מירבית (עם דadata הנדגמת עם q) ביחס לפרמטרים.

از המחברים לקחו את הרעיון הזה והפיעלו אותו על RL בצורה הבא:

1. לייצור דadata מהמודול
 2. עד שהתגםול בסט ולידציה עולה:
 - לאמן מודל בסיס (תמיד מאומנים מודל בסיס להבדיל מ-REST) כאשר כל דגימה בדadataסט מושקלת עם ערך התגםול. מכיוון שפונקציית התגםול במאמר היא בינהית זה שקול לא'
- התוצאות בדוגמאות בעלי ערך פונקציית תגםול 0

3. חזרים לשלב 1 מספר איטרציות

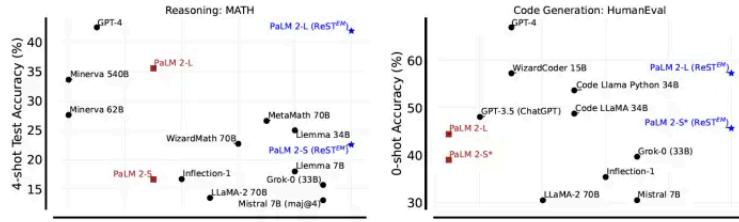


Figure 1 | Self-training with ReST^{EM} substantially improves test performance of PaLM 2 models on two challenging benchmarks: MATH and HumanEval. Results for other models are shown for general progress on these tasks and are typically not comparable due to difference in model scales. GPT-4 results are taken from Bubeck et al. (2023).

Review 183, Short: WEAK-TO-STRONG GENERALIZATION: ELICITING STRONG CAPABILITIES WITH WEAK SUPERVISION

<https://openai.com/research/weak-to-strong-generalization>,
<https://cdn.openai.com/papers/weak-to-strong-generalization.pdf>

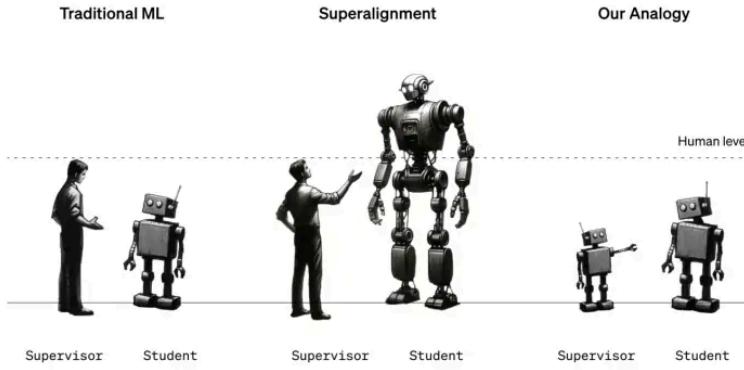


Figure 1: **An illustration of our methodology.** Traditional ML focuses on the setting where humans supervise models that are weaker than humans. For the ultimate superalignment problem, humans will have to supervise models much smarter than them. We study an analogous problem today: using weak models to supervise strong models.

אוקי', הסקירה של היום על מחקר מואוד מעניין מבית לא אחר אלא AIOpen. המאמר מנסה לפתח שיטות אימון (כיול) למודלים חזקים עם מודל פיקוח חלש. מה זה בעצם אומר? נניח שיש לנו מודל S בעל יכולות חזקות יותר (נגיד במספר פרמטרים) ממודל חלש יותר W ובנוסף יש לנו נתונים שאנו רוצים לכיל על מודל S.

פיקוח חלש אומר לנו קודם כל מתייגים את הדadata עם W ואז מאמנים מודל חזק S עם הדatasets המתויג זהה. למה זה בעצם חשוב? המאמר מדבר בעתיד הקרוב יחסית אם נגיע לאימון של מודלים בעלי יכולות superhuman למשימות שאנו בני אדם לא מסוגלים לבצע באיכות טוביה ואז אנחנו בעצם מהווים את המודל החלש W.

השאלה עד כמה אימון מודל בפיקוח חלש עובד גרוע יחסית לכיוול של מודל S עם דатаה מותוג נכון (על ידי בני אדם)? מתרבר שההפרש בביטויים הוא די גדול למרות שאימון בפיקוח חלש כן מצליח לשפר את הביצועים של המודל החזק.

השאלה בעצם האם יש שיטות שימושיות בביטויים טובים יותר מאשר אימון בפיקוח חלש? המאמר מציע שתי שיטות של שיטות. השיטה הראשונה היא אימון הדרגתית של המודל החזק. מתחילה ממודל חלש וכל פעם "מחזקים" אותו בקצב (בכמות הפרמטרים למשל) כאשר המודל מהאיטרציה הקודמת משרת בתור מודל פיקוח חלש. הגישה השנייה (consistency loss) היא לתת פחות לKENOS את המודל החזק על אי התאמת עם המודל החלש כאשר המודל החזק מאוד בטוח בתוצאה שלו. יש תוצאות מעניינות...

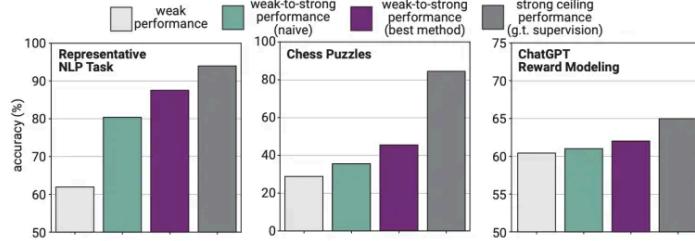
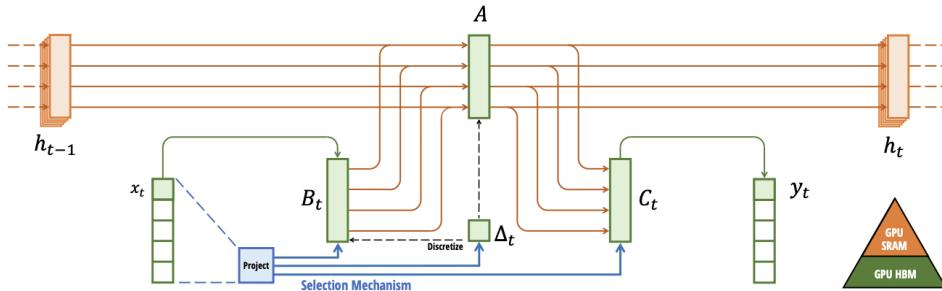


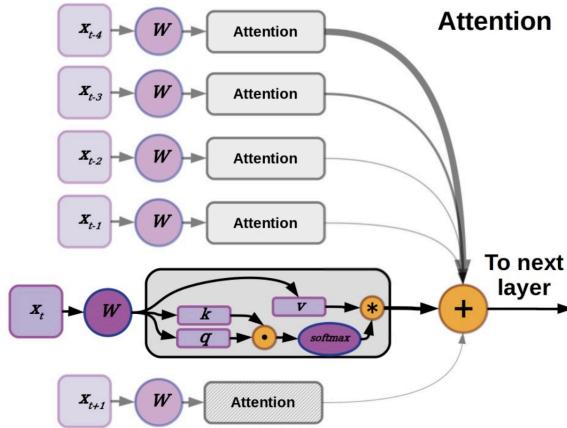
Figure 2: **Strong models trained with weak supervision generalize beyond their supervisor, and improving weak-to-generalization is tractable.** We show test accuracy on a representative NLP task (left), chess puzzles (middle) and the ChatGPT reward modeling task (right). We show the weak supervisor trained on ground truth labels (light grey), with the best method in each setting (purple), or with ground truth supervision (dark grey). For NLP and chess we supervise GPT-4 using GPT-2-level supervision, while for reward modeling we supervise a 3.5-level model using GPT-2-level supervision. The best method is the auxiliary confidence loss for the NLP task (Section 4.3.2), bootstrapping for Chess puzzles (Section 4.3.1), and unsupervised generative fine-tuning for reward modeling (Section 5.2.2; generative-finetuning is also used for the strong ceiling performance).

Review 184: Mamba Series, An Intro

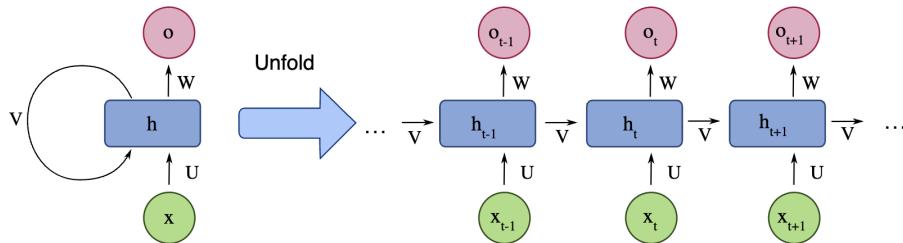


כמו שהבטחתי היום אנחנו מתחננו את סדרת הסקירות שתוביל אותנו בסופו של דבר לדובדבן שבקצת שזה תיאור של ארכיטקטורה בשם Mamba (Mamba) שעשתה הרבה רעש לאחרונה. היום אספק לכם מבוא כללי על מה שנחננו הולכים לדבר בשבוע הקרוב.

אז מה זה ממא? כאמור ממא היא ארכיטקטורת רשת נירונית שבאה לתת מענה לחישון הבולט של הטרנספורמרים והוא הסיבוכיות הריבועית במונחי אורקל הקלט (עבור קלט טקסטואלי אורכו הינו מספר הטוקנים בחולון ההקשר). נכון שלאחרונה הוציאו מספר שיפורים למנגנון תשומת הלב של הטרנספורמר (שהוא לב הבעה) כמו 2-chunk FlashAttention אבל עדין הטרנספורמרים מתकשים לעבד בצורה המיטבית עם דטהה בעל אורכי הקשר בסדר גודל של מיליון טוקנים.



از איר אנחנו נתמודד עם הסיבוכיות הריבועית של מנגנון תשומת הלב של הטרנספורמרים? הצעירים שבינו זוכרים שפעם הייתה לנו ארכיטקטורה הנקראת (Recurrent Neural Networks) (RNN) שבהם לא הייתה לנו בעיה של הסיבוכיות הריבועית (הו שם חסרונות אחרים שנדרן בהם בהמשך). ה-RNN והשכלולים שלו כגון GRU ו-LSTM לא היו צריכים לנקח בחשבון את ייצוג של כל פיסות DATA (נגיד טוקנים בטקסט) בצורה מפוזרת (כפי שמנגנון תשומת הלב של הטרנספורמרים עשה). במקום זאת הוא היה דוחש את המידע על הטוקנים (היציגים והקשרים ביניהם) באמצעות וקטור המצב (ב-LSTM יש בנוסף עוד כמה וקטורים האחרים על דחישה של זכרו).



אם הצלחנו לדוחס את כל המידע הטמון בטוקנים הקודמים **בצורה טובה** אז לא צריך להתחשב שום מידע באופן מפוזר בזמן אימון ובזמן היסק (inference) של טוקן הבא. אמנם אם כל הזיכרון שלנו נמצא בוקטור הדוחס זהה אז נשתמש בו במקום להתחשב בכל הטוקנים הקודמים. אולם יש בגישה זו שתי בעיות עיקריות:

1. **ארQUITקטורות RNN לא הצלחו לדוחס בצורה טובה את הטוקנים הקודמים** כאשר אורך חלון הקשר ארוך וזו הייתה הסיבה העיקרית שארכיטקטורות אלו נהפכו ממשימות הכרוכות בעיבוד קטעי טקסט ארוכים. הרי אם המודול לא מסוגל לזכור את המידע מהטוקנים הקודמים, לא ניתן לצפות ממנו ביציעים טובים בחיזוי טוקן הבא.

2. בטע כבר שמעתם שארכיטקטורות RNN הן **לא scalable**. מה זה בעצם אומר? כאשר אנו מבצעים אימון של מודל שפה המשימה היא לחזות חלקו קלט שהוא מסתירם (מסמכים) מהמודול. עם טרנספורמרים יש לנו יכולת לחזות את כל הטוקנים הנסתירים בצורה מקבילית עלי ידי שימוש בו בזמן מסווגות שונות (כל פעם מסמכים רק את מה שצירף). ב-SRNN זה בלתי אפשרי כי עבור חיזוי של כל טוקן אנו צריכים לחשב את ייצוג הזיכרון שלוקח בחשבון את כל הטוקנים שהיו לפניו. כמובן עבור טוקן מספר 1000 אנו צריכים לבצע חישוב סקוונציאלי (אחד אחרי השני) עבור 999 טוקנים שהיו לפניו. פעולה זו לא ניתנת למקובל עקב נוכחות של פונקציות לא לינאריות ביחסות ייצוג הזיכרון (מ מבה עוקף את

המכשול הזה באlgנטיות). כזכור זה לא יעיל ומהוות מכשול ממשוני בניתוח ייעיל של משאבי חישוב (GPUs).

עכשו נשאלת השאלה האם אנחנו יכולים להתגבר על הסיבוכיות הריבועית של הטרנספורמרים ובאותו זמן להיפטר משלבי חסרוןות שתיארנו בפסקה הקודמת? זו בדיקת השאלה המחרקית העיקרית שנדרן בה בסירות הבאות שיבילו אותנו לארQUITטורת ממבה הנחשכת.

לבסוף אתן לכם כמה טירירים קטנים בנוגע למה שאתם יכולים לראות ביום הקרובים:

1. ארכיטקטורות אותן הולכים לדבר עליו מאפשרים שני מטרים הפעלה:

a. אימון מוקבלי כמו עם טרנספורמרים במהלך אימון המודל

b. היסק מהיר כמו עם ה-RNN-ים

2. באופן די מפתיע ארכיטקטורה זו היא בעלת מבנה דומה לרשף קונבולוציה רק שהקרנלים של קונבולוציות אלו הן מאוד ארוכות

3. ארכיטקטורות אלו שואבות השראה מערכות דינמיות לינאריות וקשרות לשערור של פונקציות על ידי פולינומים אורטוגונליים.

Review 185: Legendre Memory Units: Continuous-Time Representation in Recurrent Neural Networks

https://proceedings.neurips.cc/paper_files/paper/2019/file/952285b9b7e7a1be5aa7849f32ffff05-Paper.pdf

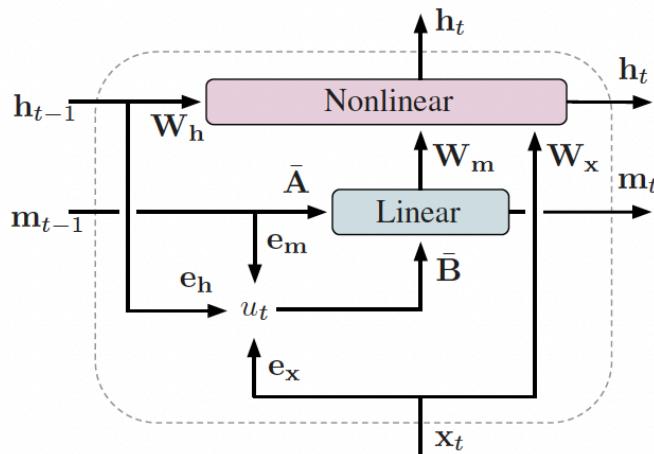


Figure 2: Time-unrolled LMU layer. An n -dimensional state-vector (\mathbf{h}_t) is dynamically coupled with a d -dimensional memory vector (\mathbf{m}_t). The memory represents a sliding window of u_t , projected onto the first d Legendre polynomials.

המאמר הראשון בסדרה שלנו מנסה לטפל בעיה הראשונה של RNNs יכולabilities של רשותות אלו לדוחות את היזירון (קלט בחילון ההקשר) בצורה מספקת טוביה. המאמר מציע גישה מקורית ומעניינת שמקורה במערכות

динמיות (Dynamic Systems) לבניית ייצוג האיךון. נניח שיש לנו פונקציית קלט רציפה (f) וANO רוצים לבנות מערכת ש"זוכרת את הפונקציה זו". ככלומר בונה ייצוג כך שהיא אפשר לשחזרה באופן מדויק. תזכירו שכדי לתאר קלט דיסקרטי כמו ציריכים רק לעשותות דיסקרטיזציה או לדגום את הפונקציה זו.

המאמר בונה מערכת דינמית המתוארת על ידי משוואות דיפרנציאליות לינאריות (מערכת דינמית, משווהה 1 במאמר) כאשר (t) זה הוא וקטור הזמן - (t) u כאמור הקלט (כרגע חד ממדי). מתרבר שעבור בחירה מסוימת של מטריצת A במשוואת המערכת הדינמית ניתן לתאר את הקלט (פרק זמן מסוים) על ידי שילוב של פונקציית הזמן (t) u ופונקציות מתמטיות הנקבעות פולינומיים של Legendre (משווהה 3 במאמר). כלומר ניתן לתאר את כל מה שקרה מבחינת הקלט עד זמן מסוים על ידי פונקציה (t) u - זהה בבדיקה מה שרצינו, נכון?

אולם הדאטה שלנו דיסקרטי (טוקנים נגיד) אז צריך לעשות דיסקרטיזציה (dagima) לגישה הזו. תלומר במקום פונקציות רציפות תהיה לנו סדרת הקלט t_n וקטור הזיכרון \vec{z}_m . גם מטריצות במערכת הדינמית שלנו צרכות לעבור דיסקרטיזציה (השערור הרגיל של הנזרת/גרדי'אנט) ועוד נקבל נוסחה רקורסיבית עבור \vec{z}_m כפונקציה של t_n ו- t_{n-1} . ניתן לתאר את את הדגימות עד $T=t$ על ידי נוסחת נוסיגה הזו.

זהו זה - יש לנו רשות בסגנון RNN כאשר הזיכרון ממודל על ידי דיסקרטיזציה של מערכת דינמית, המחשבת מקדמים של פולינומי Legendre ובאופן זה עבד לא רע אי שם ב-2020.

Review 186: HiPPO: Recurrent Memory with Optimal Polynomial Projections

<https://arxiv.org/abs/2008.07669>

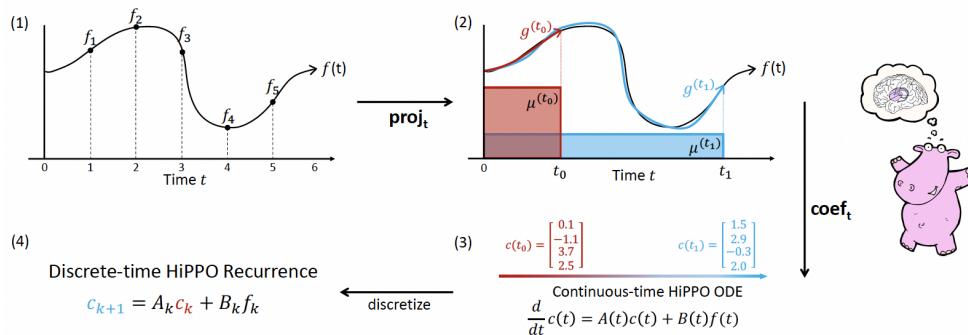


Figure 1: Illustration of the HiPPO framework. (1) For any function f , (2) at every time t there is an optimal projection $g^{(t)}$ of f onto the space of polynomials, with respect to a measure $\mu^{(t)}$ weighing the past. (3) For an appropriately chosen basis, the corresponding coefficients $c(t) \in \mathbb{R}^N$ representing a compression of the history of f satisfy linear dynamics. (4) Discretizing the dynamics yields an efficient closed-form recurrence for online compression of time series $(f_k)_{k \in \mathbb{N}}$.

הגענו למאמר השני - המאמר הזה חשוב מאוד כי הוא מפתח בסיס מתמטי מוצק המשמש כל המודלים מבוססים על מערכות דינמיות לינאריות כולל כטבון ממבה. המאמר הזה קצר (די הרבה) כבד מתמטי אך אנסה לעשות כמעט יכלתי כדי להבהיר לכם את המשך העיקרי שהוא מביא אותנו.

בסקירה הקודמת דיברנו על איך ניתן לבנות וקטור זיכרון (\hat{x}) בעל יכולת לשחזר פונקציית קלט (x) על-

ידי מינימום (f , $[0, 1]$; כאן \hat{x} מסמן גודל חלון הקשר (כלומר אורך הזיכרון). פונקציה (\hat{x}) ממודלת על ידי מערכת דינמית לינארית ושלובה עם פולינומי Legendre משוחזר לנו את הקלט x . נעיר שאנו עובדים עם הגרסאות הדיסקרטיות של המודלים האלו שהן בעצם נוסחת נסיגה עבור סדרת וקטורי הזיכרון \hat{x} .

המאמר המסורק מנסה מתמטית כללית עבור בעיית ייצוג הזיכרון של פונקציית קלט (x) בטוחום $[t, 0]$. והנה מתחילה הסיבור: קודם כל פולינומי Legendre הם מקהה פרטיאי של פונקציות אורתוגונליות במרחב הילברט (יותר נכון מרחב פונקציוני L של לבג - המקהה הפרטיאי של הילברט) המצויד בנוסף בפונקציית מידת ס. אוקי, מה הדבר הזה אומר בעצם? ממש בגודל זה מרחב של פונקציות שהמכפלה הפנימית ביןיה מוגדרת בתור אינטגרל של מכפלתן תחת מידת ס. (במקהה הפשוט ביותר מידה ס' שווה ל 1 זהותית ואנו מקבלים אינטגרל Riemann-Stieltjes). פונקציות אורתוגונלית במרחב החמוד הזה מוגדרות בתור אלו שהמכפלה הפנימית שלן שווה ל 0 (תחת מידת ס). פולינומי Legenge הן אורתוגונליות תחת מידת ס' השווה ל- \int_0^t ואפס בכל מקום אחר.

از נניח שיש לנו N פונקציות אורתוגונליות $N, \dots, 1 = i, (x)_i$ במרחבינו החמוד. ועכשו המטרה היא לתאר את הקלט $(x)_i$ על ידי $N, \dots, 1 = i, (x)_i$. כלומר אנו רוצים לבנות סכום ממושקל $(x)_i^*$ של $(x)_i$ עם מקדמים מסוימים (שימו לב שעבור i -ים שונים מקבלים וקטורי מקדמים שונים וכך יש לנו כאן פונקציה וקטורית של המקדמים התלויה ב- t).

כלומר $(x)_i^*$ צריך לקרב בצורה טובה את הקלט $(x)_i$ (כלומר לפחות שגיאה ביןיה ב- \int_0^t). והדיק מחושב בתור אינטגרל של ההפרש הריבועי בין $(x)_i^*$ ו- $(x)_i$ תחת מידת ס' (\int_0^t). כאמור היא שווה ל- \int_0^t עבור כל x ואפס בכל מקום אחר עבור פולינומי Legendre אבל כמובן קיימות עוד אפשרויות. איך נחשב מקדמים המגדירים את ההפרש הזה? לא זה מסובך: מקדם i שווה למכפלה פנימית (=אינטגרל) בין פונקציה מסוימת לפונקציית קלט x תחת אותה מידת ס'.

עכשו איך כל זה קשור למערכות דינמיות לינאריות החמודות שלנו? מתרבר כי מערכת דינמית לינארית שתיארנו בסקרירה הקודמת עבור וקטור (t) מתרבת את המקדמים של ייצוג הקלט באמצעות N פולינומי Legendre אורתוגונליים תחת מידת ס' שהגדרכו לפני. ו- N זה המימד של וקטור הזיכרון (t) תחת מידת ס' הדורשת קרבה אחידה (=זיכרון אחד) בין x^* ו- x ב- $[t, 0]$.

אם נגידר מידת ס' להיות פונקציה $(-x)\exp$ עבור t נתון, מערכת דינמית לינארית אחרת תתאר לנו מקדמים של פולינומי Laguerre (אורתוגונליים תחת ס' זה). שימו לבshima זר זיכרון הדועץ מערכית לומר הרבה ככל שעבור הזמן הנוכחי t , הזיכרון הולך ונהייה מעומעם יותר.

בנוסף המאמר מדבר גם על שיטות דיסקרטיזציה של מערכת דינמית זו וגם דן בקשר בין לבין RNNs.

אוקי, העכשו סיכום המשפט אחד של המאמר הדי כבזזה. המחברים בנו מסגרת מתמטית למידול בעיתת הזיכרון של פונקציית קלט שישתמשו אוטומטית באלגוריתם לבניית מודל attention כל הדרך למבה.

Review 187: Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention

<https://arxiv.org/pdf/2006.16236>

$$\begin{aligned}
Q &= xW_Q, \\
K &= xW_K, \\
V &= xW_V, \\
A_l(x) &= V' = \text{softmax} \left(\frac{QK^T}{\sqrt{D}} \right) V.
\end{aligned}$$

אחרי הסקירה הקודמת הובדה מאוד מכך לנו היום סקירה קילילה (הסקירה הבאה הולכת להיות די כבודה). כמו שכבר אמרנו אחד החסרונות הבולטים של הטרנספורמר היא הסיבוכיות הריבועית שלו במנוחי אורך הקלט (= מספר איברים בסדרת הקלט). הסיבוכיות זו בא על ידי ביטוי גם במהלך האימון וגם במהלך ההיסק (inference). סיבוכיות ריבועית זאת כואבת במיוחד בזמן ההיסק כאשר אין לנו יכולת לחזות מספר טוקנים בו זמן ניטי כדי ליזמי טוקן חם צריכים לדעת את ה-(1-ח) הטוקנים הראשונים. האם ניתן להפוך את הטרנספורמר לסוג של RNN במהלך ההיסק כאשר כל הזיכרון על הטוקנים הקודמים נדחס לכמה וקטורים בודדים (וקטור זכרן ווקטור של המצב)?

הטרנספורמר המקורי אינו מאפשר אופן חישוב זה כי הוא מכיל פעולה לא לינארית (softmax) בתוך המנגנון תשומת הלב שלו. ניתן לראות די בקלות שלא ניתן לעקוף את מגבלת הסיבוכיות הריבועית שלו ללא שינוי של אופן חישוב של תשומת הלב. המאמר המסביר כיצד להחליף את חישוב הסופטמקס במנגן זה בחישוב לינארי (מכפלת מטריצות) המחשבות על ידי הפעלת פונקציה לא לינארית ϕ_k על וקטור השאלות Q ושל וקטור המפתחות K . מי שעוד זוכר מה KT Kernel Trick (Kernel Trick) מבין מה שנעשה כאן הוא KT בכיוון הפוך.

כמובן שגם מאבדים כאן מהעוצמה של המנגנון תשומת הלב הרגיל אבל זה יעזר לנו לפתור את סוגיית הסיבוכיות הריבועית בזמן ההיסק. למעשה המחברים מוכיחים (ראו את התמונה לעלה) כי ניתן למשת את המנגנון הזה בסדרתי בעל סיבוכיות לינארית במנוחי אורך הקלט. כמובן בזמן האימון ניתן לחשב חזוי של כמה טוקנים בו זמן ניטי (לפי היכולת החישובית שעומדת לרשותנו) ולהנות מהיתרונו של מנגנון תשומת הלב הרגיל.

כלומר יש לנו טרנספורמר (מוחלש כמובן) באימון ו- RNN בהיסק. בהמשך נראה כיצד לשפר את הגישה זו עם SSMs (state-space models).

Review 188: Efficiently Modeling Long Sequences with Structured State Spaces

<https://arxiv.org/abs/2111.00396>

לאט לאט הגיענו למאמר הריבעי בסדרת סקירות בדרך למבהה. הפעם נסקור מאמר מ-2022 שיצא שנתיים אחרי 3 המאמרים הראשונים שסקרנו בנושא המעניין הזה. כמובן במהלך תקופה זו יצאו כמה מאמרי מעניינים שפיתחו ארכיטקטורות מבוססות מערכות דינמיות לינאריות (ובשם כלל יותר Space-State Models- SSMs).

המאמר שנסקרו לקח את הגישה זו לגביים חדשים והגיע לתוצאות די מרשימות עם נתונים בעלי אורך הקשר ארוך (למשל עברו אות אודיו המכיל אלפי או אפילו עשרות אלפי דוגמאות בשנייה. אם יש לנו מטלה שדורשת התחשבות בכמה עשרות שניות של אודיו אז אנו צריכים אורך הקשר של מאות רבעות של דוגמאות זהה די כבד עבור הטרנספורמר עם הסיבוכיות הריבועית שלו - במנוחי אורך הקשר).

אוקי', אז בואו נזכיר מהו היתרון הבולט של ארכיטקטורות מבוססות SSMs (inference). מצד אחד בעת ההיסק של טוֹקָן (המונעים מאיינו צריך להתחשב באופן מפורש בכל הדגימות הקודמות על ידי דחיסה של המידע בטוֹקָנים הקודמים (=זיכרונו) בוקטור זיכרונו אחד, המתעדכן עם המערכת הדינמית הלינארית. מצד שני במהלך האימון (כשל הטוֹקָנים ידועים) הוא אפשר חישוב בו בזמן של כל הטוֹקָנים המומוסכים.

דו-ائيות עצמאית זו התאפשרה על ידי ייצוגה של זיכרונו בתור מערכת לינארית שניית לבטא את הזיכרונו המצחבר לכל טוֹקָן כפולה לינארית. ככלומר ניתן לתאר את הפלט של עבור טוֹקָן k על ידי הנוסחה באחת התמונות (הקטנה יותר).

מטריצות בנוסחה הן הגרסאות המודוסקרוטות של המטריצות המופיעות בנוסחה של המערכת הדינמית המתארת את התקדמות הזיכרונו בזמן (טוֹקָנים). ניתן לראות כי מה שיש לנו כאן זו רשות קונבולוציה (שעלולה להיות מאוד ארוכה) שמאפשרת חישוב הייצוג של כל טוֹקָן.

קיבלנו את הארכיטקטורה הדואלית המתאימה גם לאימון וגם להיסק. עבור אורך הקשר גדול מספיק נדרשת כמות גדולה מאוד של זיכרונו. קודם כל אלו צריכים מטריצה A בגודל $N \times N$ (נגיד עבור $N=64$) עבור כל מימד של ייצוג הקלט (כי זה מה שהמערכת הדינמית שלו "צריכה לזכור"). אז חישוב קונבולוציה זו בצורה הישירה עבור מטריצה A כלילית של OPOHiP (עבור מקרה של פולינומי Legendre שנקרא LegT תחת מכסה המנווע של המערכת הדינמית) הוא מאוד כבד ודורש הרבה זיכרונו.

אז מה ניתן לעשות? קודם כל אם מטריצה A היא אלכסונית החישוב ודרישות הזיכרונו היו הרבה יותר צנעות. המחברים גם שמו לב כי conjugation של מטריצה A במערכת הדינמית (הכפלתה מימין ומשמאלי במטריצה אוניטרית V) מוביל למערכת דינמית שקולה עם התוצאה $A \cdot V$. הבעה שמטריצה A - M - P - O - H - I לא ניתן לתאר בצורה LV^* כאשר L היא מטריצה אלכסונית, ו V היא מטריצה אוניטרית (נובע מכך ש A אינה קומוטטיבית עם A^* כלומר לא נורמלית - זה השם אין מה לעשות).

از הכל אבד? מתרברר שלא. מתרברר ש A מ- $OPOHiP$ ניתן לתאר בתור סכום של מטריצה נורמלית ומטריצה בעלת רנק נמוך (עבור LegT הרנק אפילו שווה ל-1 כלומר תוספת זו כי מכפלה חיונית של שני וקטורים בעלי מימד $N \times 1$). ואז המאמר מציע אלגוריתם די לא טריויאלי עבור חישוב של קרナル קונבולוציה ארוך המבוסס על עקרונות מתמטיים:

- במקום לחשב A^*A עבור כל i ניתן לחשב z -transform (מקוטע עד L) של A ואז לחשב בצורה i פשוטה את A^*A על ידי הצבה של שורש שונים של 1 (המרוכבים) ב z -transform (הזה).

- כאשר A הוא הפרש של מטריצה אלכסונית L ומטריצה בעלת רנק נמוך מאד ניתן לחשב את z -transform בצורה יعلاה דרך $\text{Z}(\text{Woodbury})$ שמסתכם בהיפוך של מטריצה אלכסונית.

- ניתן לבצע את כל החישובים העולים כאשר מפעלים בזאת Woodbury בצורה יعلاה מאוד עם Cauchy Kernel; שזה בגודל מטריצה שנבנית בצורה מסוימת משני וקטורים

לבסוף, מבצעים את החישובים האלו עבור כל מימד של ייצוגי הטוֹקָנים בנפרד ואז מערבבים עם שכבה לינארית (או כמה). מטריצות אלכסוניתות L (למעשה וקטורי), וקטוריים C, B, P ו- Q שמכפלתם היא מטריצה בעלת נמוך מאומנות בנפרד עבור כל מימד של ייצוג הטוֹקָנים.

זהו, יצא אורך - הסקירה הסקירה תהיה קצרה יותר.

Review 189: Simplified State Space Layers For Sequence Modeling(S5)

<https://arxiv.org/abs/2208.04933>

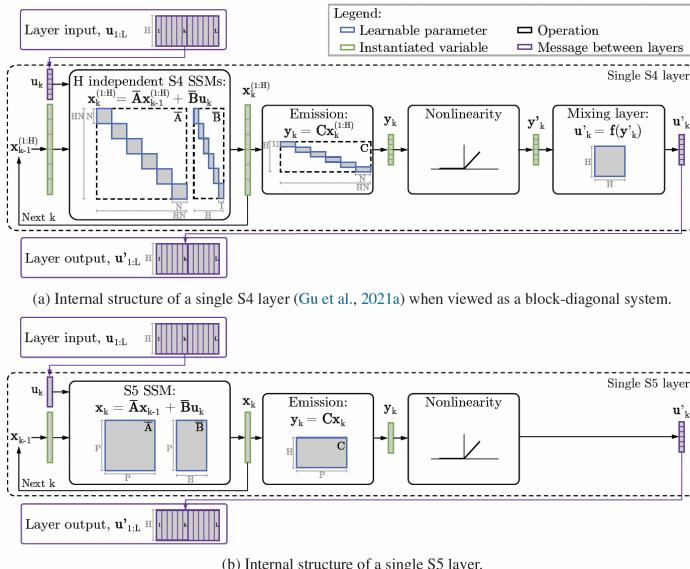


Figure 2: Schematic of the internal structure of a discretized S4 layer (Gu et al., 2021a) (top) and S5 layer (bottom). Note D is omitted for simplicity. We view an S4 layer as a single block-diagonal SSM with a latent state of size HN , followed by a nonlinearity and mixing layer to mix the independent features. (b) In contrast, the S5 layer uses a dense, MIMO linear SSM with latent size $P \ll HN$.

משיכים עם הסקירה החמישית כל הדרך למ מבה. סקירה זו תהיה די קليلת כי היא בסך הכל מציעה שכלול לארכיטקטורת S4 שדיברנו עליה בהרחבה בסקירה הקודמת. למעשה S4 בנויה מ- H (מימד של ייצוג הטוקון) SSMים של אחד מהם מומש עם מערכת דינמית ליניארית שdone עליה בהרחבה בסקירה בסיסית. כל SSM מהווה בעצם זכרון עבור כל מימד של וקטור ייצוג הטוקון לאורך זמן. זמן כאן ציר הטוקונים שאנו רוצחים לזכור כדי לקבל החלטה מושכלת עבור הטוקון הנוכחי.

אם נביט בನוסחאות המתארות SSM ניתן לראות כי H מערכות SSM האלו אפשר לתאר כ-SSM אחד גדול המתואר על ידי מטריצה A בлокית אלכסונית שבאלכסון שלה נמצאות מטריצות $H = i, j, A_i, A_j$ המתארות כל SSM. וקטורי B ו- C של ה- SSM הגדול זהה ניתן לבנות על ידי שרשרת של וקטורי i, j, B_i, C_j של H המערכות SSM האלה.

כਮון שכלי הסיפור הזה דורש לא מעט זכרון ולא מעט חישובים במילויים כאשר H (מימד ייצוג הדאטה) הוא סדר גודל של כמה מאות או כמה אלפיים. אז המאמר המשוכן מציע להשתמש באותה מטריצה A עבור המערכות הדינמיות המתוארות זיכרון של כל מימד שיקיים ייצוג הדאטה. גודל של מטריצה A נבחר הרבה יותר קטן מ- PH שזה גודל של מטריצה A עבור כל המימדים של ייצוג התוקן יחד (= גודל המטריצה הבלוקים האלכסוניים). כਮון שבדרך זו נחסכים לנו גם הזיכרון וגם כמות החישובים הנדרשת גם בהיסק וגם באימון.

כמובן שהקטנה שצזו של מימד מטריצה A עלול לפגוע בביוצוי המודל (כי אידיאלית זיכרון של מימדים שונים של ייצוג דאטה עשויים להכיל אופיינים שונים של זיכרון; נגיד, זיכרון ארוך וקצר טווח). המחברים בוחנים מספר דרכים לצמצום פגיעה זו על ידי עדכון חכם של A ועוד כמה טריקים נחמדים. המחברים למשל בוחרים אופצייה של מטריצה A בעלת מימד KP כאשר K הרבה יותר קטן מ-H.

5בקיצור מאמר קליל וקל לקרוא...

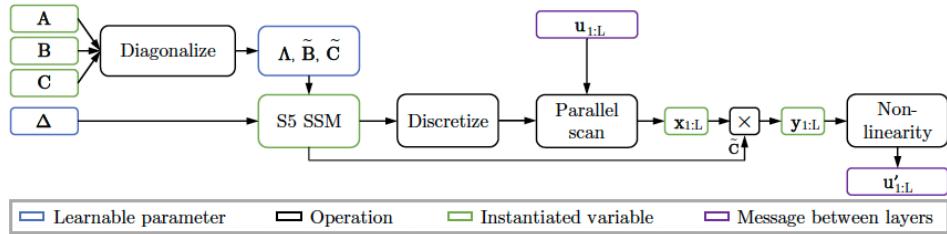


Figure 1: The computational components of an S5 layer for offline application to a sequence. The S5 layer uses a parallel scan on a diagonalized linear SSM to compute the SSM outputs $\mathbf{y}_{1:L} \in \mathbb{R}^{L \times H}$. A nonlinear activation function is applied to the SSM outputs to produce the layer outputs. A similar diagram for S4 is included in Appendix B.

Review 190: Hungry Hungry Hippos: Towards Language Modeling with State Space Models(H3)

<https://arxiv.org/abs/2212.14052>

Algorithm 1 H3 Layer

```

Require: Input sequence  $u \in \mathbb{R}^{N \times d}$  from the previous layer, weight matrices  $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V, \mathbf{W}_O \in \mathbb{R}^{d \times d}$ , a shift
SSM  $\text{SSM}_{\text{shift}}$ , a diagonal SSM  $\text{SSM}_{\text{diag}}$ , head dimension  $d_h$ .
1: Compute  $\mathbf{Q} = u\mathbf{W}_Q, \mathbf{K} = u\mathbf{W}_K, \mathbf{V} = u\mathbf{W}_V \in \mathbb{R}^{N \times d}$ .
2: Pass  $\mathbf{K}$  through the shift SSM:  $\bar{\mathbf{K}} = \text{SSM}_{\text{shift}}(\mathbf{K}) \in \mathbb{R}^{N \times d}$ .
3: Split  $\mathbf{Q}, \bar{\mathbf{K}}, \mathbf{V}$  into  $H$  “heads”  $(\mathbf{Q}^{(h)}, \bar{\mathbf{K}}^{(h)}, \mathbf{V}^{(h)})$  for  $h = 1, \dots, H$ , each a sequence of  $N$  vectors of size  $d_h = d/H$ .
4: for  $1 \leq h \leq H$  do
5:   Take the batched outer product  $\bar{\mathbf{K}}^{(h)}(\mathbf{V}^{(h)})^\top \in \mathbb{R}^{N \times d_h \times d_h}$  (batched in the  $N$ -dimension) and pass it through a
      diagonal SSM:  $\mathbf{K}\mathbf{V}^{(h)} = \text{SSM}_{\text{diag}}(\bar{\mathbf{K}}^{(h)}(\mathbf{V}^{(h)})^\top) \in \mathbb{R}^{N \times d_h \times d_h}$ .
6:   Batch-multiply by  $\mathbf{Q}$ :  $\mathbf{O}^{(h)} = [\mathbf{Q}_1^{(h)}\mathbf{K}\mathbf{V}_1^{(h)}, \dots, \mathbf{Q}_N^{(h)}\mathbf{K}\mathbf{V}_N^{(h)}] \in \mathbb{R}^{N \times d_h}$  (batched in the  $N$ -dimension).
7: end for
8: Concatenate the output  $\mathbf{O}^{(h)}$  of each head, and multiply by the output projection matrix  $\mathbf{W}_O \in \mathbb{R}^{d \times d}$ .

```

עד עכשוו ראיינו מאמרים שמיישו את ארכיטקטורת SSM בתור רכיב הזיכרון של המערכת. אף אחת מהמאמרים שסקרנו לא ניסה לשלב גישה זו(SSM) יחד עם מנגנונים אחרים שמכירים לנו מעולם שלUIDOT סדרות דאטה עם רשותות נוירונים. המאמר המסתוקר משלב את גישת SSM, המושמת באמצעות מערכות דינמיות לינאריות, עם מנגנון תושמת הלב הלינארי.

דברנו על מנגנון attention הלינארי בסקירה השלישי של המאמר: Fast Autoregressive Transformers with Linear Attention (Fast AT). המאמר זהה אליו הceilip אט מנגנון תושמת הלב הרגיל עם softmax של הטרנספורמים בחישוב לינארית: $(q^*f)^*(k) = f(q^*k)$ כאשר $*$ מסמן מכפלה פנימית ו- f היא פונקציה לא לינארית. המאמר מראה כי ניתן לתאר טרנספורמר עם מנגנון זה בתור RNN ולהימנע מסיבוכיות חישוב ריבועית הרגילה של הטרנספורמים. ככלומר אין צורך להתחשב בצורה מפורשת בכל פיסות הדטה לפני טוקן או בשביל לחזות אותו אלא כל הזיכרון של הטוקנים הקודמים נדחס ושמור בשני וקטורים.

אוקי', אבל למה צריך בעצם לשלב ארכיטקטורות מבוססות SSM עם מנגנון אחרים? התשובה היא פשוטה - ארכיטקטורות אלה לא מספיק טובות לכמה מטריות. למשל מחברי המאמר שמו לב כי במשימות כמו *Induction Head* נדרש לעקוב על טוקן שבא אחרי טוקן מסוים, ארכיטקטורה זו מפגינה ביצועים לא מרשימים במיוחד. כדי להתמודד עם סוגיה זו המחברים הציעו לשלב SSM עם מטריצות A מסוימות עם מנגנון תשומת הלב הלינארית.

از איך כל הסיפור הזה עובד? בשלב הראשון מכפילים את "צוגי הטוקנים" במטריצות K, Q ו- V כמו בטרנספורמרים. בשלב השני מפעילים SSM על המפתח k (עבור כל הטוקנים) עם מטריצה A המדמה "זיכרון" של הטוקן הקודם" (בערך $A_{j,j} = 1$ ו- 0 אחרת). מבחינת מנגנון תשומת הלב הלינארית זה "מקביל" ל(k) f למרות ש f כאן "די לנארית".

בשלב השלישי לוקחים v, והتوزואה של השלב הקודם ל h חתיכות (= "ראשים" במנגנון ה-*attention*). לאחר מכן מכפילים כל חתיכה של q בחתיכה של התוצאה של השלב הקודם (עם A) ו"מעבירים" את התוצאות דרך SSM עם מטריצה A אלכסונית. את התוצאה מכפילים ב-q, מאחדים את כל התוצאות ומכפילים במטריצה W כמו שמקובל בטרנספורמרים מרובי ראשים(multi-head transformers).

בסוף המאמר מציע מנגנון הנקרא *FlashConv* לחישוב חייזי הטוקנים באופן מקביל במהלך האימון. כמו שאתם הזכירם הקרןל קונבולוציוני שם מאוד ארוך וחישובו יכול להיות יקר גם מבחינת הזמן וגם מבחינת המקום העשוי בזיכרון נאייה. המחברים משלבים את המנגנון כאשר העיקרונו המוביל הוא ניצול מקסימלי של זיכרון SRAM ומהיר שיש ב-GPUs תוך מזעור של העARBוטה דאטה לשם (זה איטי ובד"כ מהו צואר בקבוק). החייזון הזה לא גדול ולא ניתן לדוחוף שם יותר מדי אז נדרש שיטות מתוחכבות המפרקות את חישוב הקונבולוציה לחלקים תוך ניצול תכונות של FFT ו- IFFT. נזכיר שהחישוב הקונבולוציה מתבצע בצורה: $((x \text{IFFT}(c) \text{FFT}(c))^*$

Algorithm 2 State Passing Algorithm

Require: Input $u \in \mathbb{R}^N$, SSM parameterized by matrices $\mathbf{A} \in \mathbb{R}^{m \times m}$, $\mathbf{B} \in \mathbb{R}^{m \times 1}$, $\mathbf{C} \in \mathbb{R}^{1 \times m}$, $\mathbf{D} \in \mathbb{R}^{1 \times 1}$, chunk size N' where N is a multiple of N' .

- 1: Precompute $\mathbf{A}^{N'} \in \mathbb{R}^{m \times m}$, $\mathbf{M}_{ux} = [\mathbf{A}^{N'-1}\mathbf{B}, \dots, \mathbf{B}] \in \mathbb{R}^{m \times N'}$, $\mathbf{M}_{xy} = [\mathbf{C}, \mathbf{CA}, \dots, \mathbf{CA}^{N'-1}] \in \mathbb{R}^{N' \times m}$.
- 2: Split the inputs $u_{1:N}$ into $C = N/N'$ chunks $u_{1:N}^{(c)}$ for $c = 1, \dots, C$.
- 3: Let the initial state be $x_{N'}^{(0)} = 0 \in \mathbb{R}^m$.
- 4: **for** $1 \leq c \leq C$ **do**
- 5: Compute $y^{(c)} = \mathbf{M}_{xy}x_{N'}^{(c-1)} + \text{BLOCKFFTConv}(f, u_j) + \mathbf{D}u^{(c)} \in \mathbb{R}^{N'}$.
- 6: Update state: $x_{N'}^{(c)} = \mathbf{A}^{N'}x_{N'}^{(c-1)} + \mathbf{M}_{ux}u^{(c)}$.
- 7: **end for**
- 8: Return $y = [y^{(1)}, \dots, y^{(C)}]$.

Review 191: Hyena Hierarchy: Towards Larger Convolutional Language Models

<https://arxiv.org/abs/2302.10866>

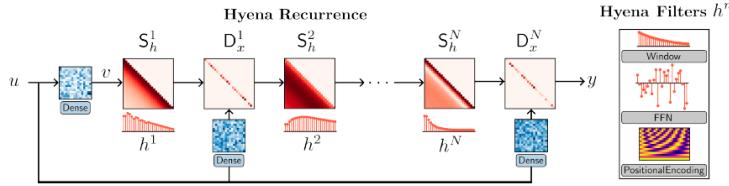


Figure 1.1: The Hyena operator is defined as a recurrence of two efficient subquadratic primitives: an implicit long convolution h (i.e. Hyena filters parameterized by a feed-forward network) and multiplicative element-wise gating of the (projected) input. The depth of the recurrence specifies the size of the operator. Hyena can equivalently be expressed as a multiplication with *data-controlled* (conditioned by the input u) diagonal matrices D_x and Toeplitz matrices S_h . In addition, Hyena exhibits sublinear parameter scaling (in sequence length) and unrestricted context, similar to attention, while having lower time complexity.

היום סוקרים את המאמר השביעי בסדרה וכן אנו חיב להודות שלקוח לי הרצה מאד זמן לצலול למאמר זהה לעומק למרות שטכנית המאמר לא מורכב במיוחד (בטח לא קרוב ל Hippo(H)). אבל המאמר כתוב בצורה נוראית: מצד אחד הוא עמוס בפרטים לא מהותיים ומצד שני געשה ממש ניכר (על ידי המחברים) להסתיר את הפרטים המהותיים עם מלל אינסופי. לא יודע האם זה נעשה בזדון או לא אבל המאמר הזה לוקח לי בערך פי 4 יותר זמן ממאמר ממוצע שעזה הרבה סטיות תקן מה ממוצעו (יש לי מדגם די גדול).

אחרי שהחרתתי את הקיטור אפשר לשקור את המאמר זה שמציע הכללה חמודה ל 3H שסקרנו קודם. 3H היה די נחמד אבל עדין הביצועים שלו לא היו בשמיים עבור כמה שימושות על הדטה בעלי אורך הקשר ארוך מאוד. אז באו לנו ממחברי Hyena והציגו לשפר את ביצועי 3H אך לא במחair של עלייה ניכרת במשאב חישוב והזיכרון.

אוקי, אז מה הם הציעו בעצם? אתם זוכרים שב-3H אנו לקחנו וקטורי מפתח עבור הטוקנים בתוך חלון ההקשר (=מטריצה K) העברנו אותו דרך דרך SSM (State-Space Models) ואז הכפלנו אותו בווקטורי שאלתה (=מטריצה Q) והעברנו את התוצאה דרך SSM נוספת עם מטריצה A אחרת ואת התוצאה הכפלנו בווקטורי ערך עבור כל הטוקנים בתוך חלון ההקשר (=מטריצה V)? כל המנגנון הזה הוא למעשה *attention mechanism*.

از הכללה הראשונה המוצעת במאמר היא הגדלת מספר הוקטורים שעליהם מופעלת SSM (בצורה לא מפורשת - מדובר על זה עוד מעט) ל N. קלומר יש לנו $N+1$ הטלות של ייצוג הטוקנים (אחד עבור מטריצת הערך V). אחרי שיש לנו את ההטלות האלו מפעילים עליהם מה שבמאמר נקרא Short Convolution (קונבולוציה קצרה) ביציר הטוקנים. זה געשה כנראה כדי ללמד את האינטראקציות בין הטוקנים הסמוכים (המאמר לא מסביר כלום לגבי זה).

מפה העניינים קצת מסתובבים. אנו לוקחים מטריצת הערך V מההטלה האחורונה ומפעילים עליהם SSM (אותה מערכת דינמית לינארית) אבל בצורה לא מפורשת. מה זה אומר אבל? אנו יודעים שהפעלת SSM לסדרה של L טוקנים שקופה להפעלה של קרNEL קונבולוציה באורך L על ייצוג טוקנים אלו. קרNEL קונבולוציה זה מוגדר על ידי המטריצות המגדירות את ה-SSM (שהה A, B, C). איז ניתן להגיד SSM בצורה לא מפורשת דרך דרך ה الكرNEL. צריך לזכור פועלה זו שקופה להכפלת וקטורים, המרכיבים מטריצת ערך V, במטריצת קונבולוציה גדולה (= שעזה אותו מנגנון של *attention mechanism*).

למשל ב-3H (שסקרנו בפעם הקודמת) הוי לנו שני SSMs (עם מטריצה אלכסונית ועם מטריצת חזזה - B^{-1}) ומתרבר שנית ליעיג אותם בצורה לא מפורשת עם קרNEL שהוא מכפלה של שתי מטריצות שכיל אחת מהן היא מכפלה של מטריצה אלכסונית במטריצת Toeplitz. מה שמיוחד במטריצת Toeplitz היא שכיל שורה בה כי הזרה שמאליה של השורה הקודמת. תוכנה מעניינת של כל מטריצה Toeplitz היא שהיא ייצוג של קרNEL קונבולוציה.

از המחברים לקחו את היצוג הלא מפורש של SSM ובנו אותו מ- N מכפלות של מטריצות אלכסוניות ומטריצות Toeplitz (שונות). כזכור מתחילה מטריצה V עבור הטוקנים מפעלים עלייה מייפוי $\$H\$$ לינארי (= קרナル קוונבולוציה) די מסורבל. כזכור H הוא הרכבה של N מייפוי $\$N...1,i\$$ לינאריים שכל אחת מהן היא קוונבולוציה המיצגת על ידי מטריצה Toeplitz (מס' 0) ומכפלת התוצאה איבר-איבר בהטלה מספר 0 של וקטורי הטוקנים. במאמר כל הסיפור זהה נקרא Hyena operator מסדר N .

אוקי, מה הבעיה העיקרית עם הגישה זהה? זה דורש הרבה זיכרון בטח עבור N גדול יחסית. אז המאמר מציע פתרון מאד אלגנטי. במקום ללמידה את כל N קרナルים אלו בצורה מפורשת נגדיר אותם באמצעות רשות ניירונים רדודה (fully-connected). גם נוכל לשנות על מספר פרמטרים וכך לשומר על זיכרון קבוע פחות או יותר לכל ערך של N . כך ניצור את כל N קרナルים עם רשות אחת בלבד. ארכיטקטורת רשות רדודה זאת היא די מינימלית והיא מכילה פונקציות אקטיבציה מוחזקות (כדי ליצור קוונבולוציות עם תדרים גבוהים).

בנוסף מכפלים קרナル זה (איבר איבר) בפונקציה מעריכית עם פרמטר חיובי דelta $\$t\$$ ביציר הטוקנים. המכפלה זו באה לשקף דעיכה בהתהבות הטוקנים(=attention) ככל המרחק בין הטוקן החזו'י ועד. המאמר משתמש במקרה אופרטורי Hyena (ערוצים) במקביל עם מקדים $\$delta\$$ שונים המבטאים קצב דעיכה שונים של attention. כל אופרטור זה מופעל על וקטורי קידוד מיקומי (positional encoding).

ודבר אחרון: כל הקוונבולוציות מחושבות דרך FFT(Fast Fourier Transform) וגם IFFT(Fast Inverse Fourier Transform) כמו במאמר של 3 H (כי זה פשוט יותר מהיר). כמובן כל SSM (גם לא מפורש) מופעל על כל מימד של ייצוג הטוקנים שטיפה מסבר את התיאור אבל עדין הכל נשאר לינארי.

Algorithm 1 Projection

Require: Input sequence $u \in \mathbb{R}^{L \times D}$

1. In parallel across L : $\hat{z} = \text{Linear}(u)$, $\text{Linear} : \mathbb{R}^D \rightarrow \mathbb{R}^{(N+1)D}$
2. In parallel across D : $z = \text{DepthwiseConv1d}(h, \hat{z})$, h is a short convolution filter
3. Reshape and split z into x^1, \dots, x^N, v . Dimensions of one element are $x^n \in \mathbb{R}^{D \times L}$

Return x^1, \dots, x^N, v, x^n

Algorithm 2 Hyena Filter

Require: Sequence length L , positional embedding dimension D_e

1. $t = \text{PositionalEncoding}(L)$, $t \in \mathbb{R}^{L \times D_e}$
2. In parallel across N, L : $\hat{h} = \text{FFN}(t)$, $\text{FFN} : \mathbb{R}^{D_e} \rightarrow \mathbb{R}^{ND_e}$, $\hat{h} \in \mathbb{R}^{L \times ND_e}$
3. Reshape to $h \in \mathbb{R}^{N \times D \times L}$
4. $h = \hat{h} \cdot \text{Window}(t)$, $h \in \mathbb{R}^{N \times D \times L}$
5. Split h into h^1, \dots, h^N

Return h^1, \dots, h^N

Algorithm 3 Forward pass of Hyena

Require: Input sequence $u \in \mathbb{R}^{L \times D}$, order N , model width D , sequence length L , positional embedding dimension D_e

1. $x^1, \dots, x^N, v = \text{Projection}(u)$
2. $h^1, \dots, h^N = \text{HyenaFilter}(L, D_e)$
- for $n = 1, \dots, N$ do
 3. In parallel across D : $v_t \leftarrow x_t^n \cdot \text{FFTConv}(h^n, v)_t$
- end for

Return $y = v$

Review 192: RWKV: Reinventing RNNs for the Transformer Era

<https://arxiv.org/abs/2305.13048>

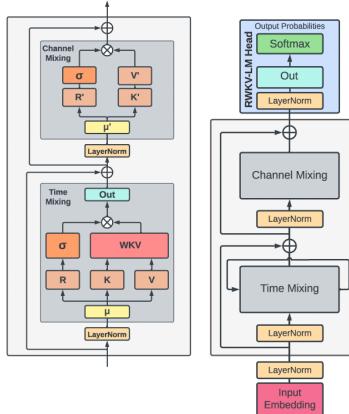


Figure 2: Elements within an RWKV block (left) and the complete RWKV residual block, equipped with a final head for language modeling (right).

אוקי', אחרי כמה מאמרם 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` ניתן לנארו 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 כהו הטרנספורמר מן המניין. ככלומר יש בו את הדואליות שרצינו: חיזוי מקבילי של טוקנים ממושכים במהלך האימון וחיזוי טוקנים בעל סיבוכיות לנארית במהלך ההיסק (inference).

המאמר שנתקoor היום מזכיר את הטרנספורמר `-RNN` באופן מפורש אפילו קצת יותר. המחברים לוקחים טרנספורמר עם מנגנון "choice" "פשות יותר" ומוסיפים קצט RNN לאופן בו מחושבים מטריצות מפתח K ומטריצת ערך V. אבל קודם אוסף לכמה פרטיטים על מנגנון "attention" שלו הוכיחו המחברים בתור בסיס ולמה אני שם אותו כאן בಗרשים. אז מנגנון זהה 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 התחילה כמו אותו ללא מונע). `AFT` מחליף את המנגנון הרגיל של חישוב `choice` של הטרנספורמר ביצה שדורש משמעותית פחותה זיכרון מהטרנספורמר הרגיל (בגרסתו הפשוטה גם סיבוכיות חישובית מוקטנת עד כדי לנארית במונחי אורך הקלט) ועשה את זה בדרכו מאד הגיוני. `AFT` מחליף את המכפלות הפנימיות בין וקטורי שאלתה q וקטור המפתח k באקספוננט של סופטמקס (זהה הלב של המנגנון והסיבה לסיבוכיות הריבועיות) בסכום של וקטורי המפתח עם מטריצת משקלים נלמדת $z_j w$ (מנורמל). ככלומר לא מתחשבים בוקטורי שאלתה q אלא משתמשים במקדים קבועים ומחושבים על סמך סט האימון. לאחר מכן צירוף לנארו עם וקטור הערך V כמו בטרנספורמר הרגיל.

כלומר מיקדי `attention` בין טוקן j לא תלויים באופן מפורש ביצוג טוקן i אלא רק $b - j$. בחרות חכמות (פרמטריזציה) של $z_j w$ מאפשרות להקטין את דרישות זיכרון והסיבוכיות החישובים כאשר המחיר הוא כפונקציית expressiveness של המודל. אחת הבחרות של $z_j w$ היא פונקציית דעכת מעריכית כאשר הארגומנט הוא מרחק בין הטוקנים (המאמר המסור מסתמש בה).

אוקי', אז איך מליבשים על זה `RNN`? לוקחים את המנגנון ה-`choice` מהפסקה הקודמת עם שפuzzו כל ליציבות נורմית - הוספה של וקטור w (המנגן הנקריא w) ומפעלים אותו עם וקטורי מפתח K ו- V מחושבים כמו ב-`RNN`. ככלומר בונים וקטורים אלו (K ו- V) תלויים באופן מפורש ביצוג הטוקן הנוכחי **וגם ביצוג הטוקן הקודם**(זה כל הקטע). במקום להכפיל את ייצוג הטוקן j במטריצות K_w ו- V_w (כמו בטרנספורמר הרגיל)

מכפילים אותם בסכום ממושקל (עם משקלים נלמדים) של ייצוג הטוקן t x הנוכחי ויצוג בטוקן הקודם $\{t-1\}x$. בנוספ' מחשבים וקטורי z (הנקרא receiptance) באוטה הצורה (עם t x ו- $\{t-1\}x$ ומטריצת ZW). וקטורי z למשהו שימושים לנו כדי "לשערך" עד כמה אנו צריכים להתחשב בה (מחושבת עם הסיגומואיד כמו בזמנים הטובים ב-RNN). כל הסיפור זהה נקרא באופן לא מפתיע rwkv.

בסוף משלבים את התוצאה של rwkv עם וקטורי מפתח וערך המוחשבים באוטה צורה כמו ב-rwkv (התוצאות ב- t x ו- $\{t-1\}x$ אבל עם מטריצות הטלה נלמדות אחרות). איך משלבים? כרגע בצורה של ResNet.

זהו זה. שמח לבשר שהמאמר הבא שנסקור בדרך למבנה גם יהיה קליל (Retentive Network).

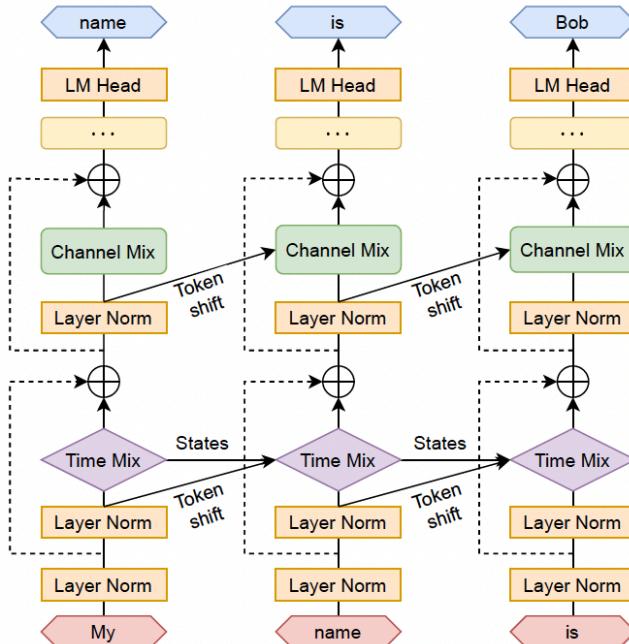


Figure 3: RWKV architecture for language modeling.

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

<https://arxiv.org/abs/2307.08621>

זה הולכת להיות הסקירה הקללה ביותר (אך קצר ארוכה). המאמר משתמש באופן די אלגנטי בReLUוניות שהוצע ב-8 המאמרים שכבר סקרונו. אזכיר שהמבנה המשותף במאמרים שסקרנו הינה מטרה למצוא ארכיטקטורה בעלת דואליות הבאה:

- ← 1. ניתנת לאימון באופן מקביל כמו הטרנספורמרים
- ← 2. היסק (inference) מהיר (=לינארי במונחי אורך חלון הקשר) שלא מצריך התחשבות מפורשת בכל טוקני של חלון הקשר.

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

אם לא אזכיר בקצרה. עבור סדרת טוקנים נתונה יש לנו מערכת דינמית לינארית (DMS) שבאמצעותה אנו מיצגים בצורה איטרטיבית את דיכאון ch_s הנזכר בחרטוטוקנים הראשונים בסדרה. בעזרה DMS ניתן לחשב את ch_s מיצוג הדיכאון קודם $\{1\text{-}\text{ch}_s\}$ ומיצוג של טוקן ה- ch , מסומן ch_v .

לאחר מכן באמצעות וקטור ch_s אנו מגדירים פלט המודל ch_o עבור טוקן ch (= "יצוג תלוי" הקשר או contextualized embedding) דרך הטלטוט ch עם מטריצה Q . נציין כי DMS מגדירה את מעבר(הLINEAR) בין ייצוג של הדיכונות ch_1 ו- ch אפשר חיזוי במקביל עבור כמה טוקנים במהלך אימון.

אותה DMS מוגדרת באמצעות מטריצות A ו- K וכאמור הפלט ch_o מוגדר באמצעות מטריצה הטלה Q . מטריצות Q ו- K הן אלו שנקראות בטרנספורמר מטריצות שאילתת וערך ומחושבות באותה צורה: $\text{Q} = \text{X}^* \text{W}_{\text{Q}}$, $\text{K} = \text{X}^* \text{W}_{\text{K}}$, כאשר X הוא ייצוג הטוקנים.

עכשו השאלה איך אנו מגדירים חישוב מקבילי של ch_o עבור כמה ch ? הרי עבור ch גדול מספיק העלה של מטריצה A בחזקה עלולה להיות יקרה גם מבחינת דיכאון וגם מבחינת מאשי חישוב. אך פותחים אחד הפרקים הראשונים של ספר של אלגברה לינארית ומגלים שניתן לתאר מטריצות ריבועיות (לא כולן) בתור $\{\text{A}^* \text{L}^* \text{D}^* \text{L} = \text{A}\}$ כאשר D היא אלכסונית עם ערכיים מרוכבים $\text{d}_{j,j} = \exp(\lambda_j)$.

מה בעצם טוב בייצוג הנחמד זהה? זה מאפשר לנו להעלות את מטריצה A בחזקה והבעיה שלנו עם חישוב A^* נראית פתרה. המאמר גם מניח ש $\text{d}_{j,j} = \lambda_j$ וזה מאפשר את היצוג הבא של המודל שם:

למעשה המחברים מחליפים את מנגנון $\text{h}\text{-attention}$ הממומש עם סופטמקס בטרנספורмерים עם $\text{h}\text{-hops}$ הדוער לצורך מעריכת כפונקציה של בין הטוקנים. כדי העין שקראו את הסקירה הקודמת שלי ישימו לב שעיקרון דומה ממומש גם ב-RWKV אבל די מוצע מעריכי של המידע מהטוקן הקודם. וכמובן ייצוג זהה חישוב מהיר עבור כל טוקן במהלך היסק (זהה תכונה 2 שלנו).

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

שכלול נוספת הוא שימוש במקדמי gamma שונים ל"ראשים" (heads) שונים של RetNet. זה למעשה מקנה למודל יכולת יותר להתמקד בטוקנים קרובים יותר (λ_{gamma} גבוהה) ו"לפזר" את $\text{h}\text{-attention}$ גם טוקנים רחוקים (λ_{gamma} נמוך). שילוב של ראשים בעלי λ_{gamma} שונים "לחקות" את הטרנספורמר (פחות במידה מסוימת).

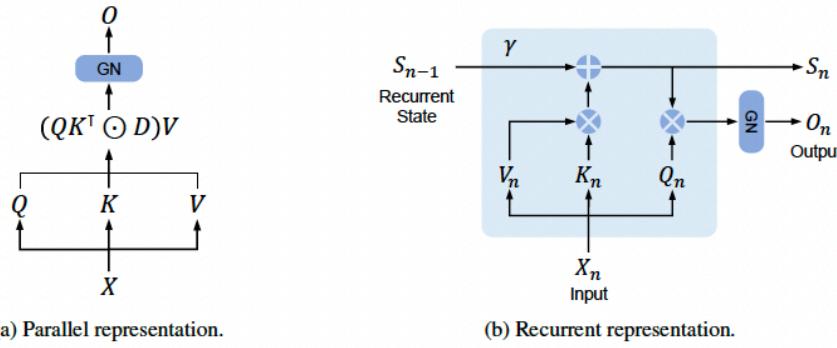


Figure 3: Dual form of RetNet. “GN” is short for GroupNorm.

Review 194: Mamba: Linear-Time Sequence Modeling with Selective State Spaces

<https://arxiv.org/abs/2312.00752>

זה קורה עכשו, אחרי 9 סקירות שחלקים היו די לא פשוטות הגענו למטרתנו הקדוצה שזה Mamba. מכיוון שאנו מפרנס סקירות בשלושות (באתר MDL) אני אוציא עוד 2 סקירות נוספות של שכלי מomba (אחד מהם EMo והשני עוד הוחלט).

האמת שהחרי לנו הבנו מה- (space-state models) SSM ואיך ניתן לבנות ארכיטקטורה מבוססת עליהם לעיבוד DATA סדרתי, השכלול המוצע על ידי mamba הוא די אינטואיטיבי ומתבקש. כמו שאתם זוכרים SSM מומשת בתור מערכת דינמית(DLS) לינארית כאשר הקטלט למערכת זו היא ייצוג וקטורי (embeddings) של איברי הסדרה (=טוקנים).

בשלב הראשון המערכת הדינמית מחשבת וקטור s הוא הוא ייצוג דחוס של זיכרון כלומר וקטור “הזכור” את המידע הרלוונטי עבור כל הטוקנים הקודמים לטוקן הנוכחי. בשלב השני מחשבים את הפלט עבור טוקן זה המזון לשכבה הבאה (שיכולה להיות גם שכבת שמיירת פלט סופי). כל חישובים אלו מתבצעים באמצעות מיפויים לינאריים כלומר מכפלות במטריצות. חשוב להבין שכל המעברים בין ייצוגי הזיכרון בין הטוקנים הם לינאריים ונשלטים על ידי מטריצה A וקטורי C, B וקטור delta B, C וקטור delta A (באופן גס) את קצב דעיכה של הזיכרון (כלומר ככל ש delta גובה יותר אנו נוטים “לזכור” פחות מהטוקנים הקודמים).

מה היתרונות של הארכיטקטורה זו? היא בעלת תכונה הדואלית המיזכרת המשלבת 2 התכונות הבאות:

- ניתן לחזות באופן מקביל (בו זמן) כמה טוקנים במהלך אימון (כמו בטרנספורמרים)
- חייזי מהיר של טוקן במהלך היסק (לא התחשבות בכל הטוקנים בחלון ההקשר כמו בטרנספורמרים שמביא לנו את הסיבוכיות הריבועית).

כלומר הארכיטקטורות מסוג זה הם יעילים בזמן האימון ומהירים בזמן ההיסק. אבל כמו שאתם יכולים לנחש יש לנו מחר לשלים על כל התכונות הנחמודות האלו. ומהיר הוא כמובן יכולת של המודל למדוד תלויות מורכבות של הדטה. עקב כך מאמרים כמו S4, H3, Hyena ניסו ליצור את הפרמטרים של DLS (המגדירה מעברים בין ייצוגי הזיכרון ויצירת הפלט) בצורה חכמה (ודי מורכבת).

אבל מתרברר שזה לא מספיק. מעברים לינאריים עם פרמטרי DLS קבועים לא מסוגלת לממדל>Data מורכב (כמו שפה טבעיות). אחד המשימות שמודל זהה נכשל עליה הוא העתקת טוקנים הבאים אחריו טוקן ספציפי (זהה די' הגינוי לאור הפרמטרים הקבועים של DLS). כמו שאתם יכולים כבר לנחש אולי מחברי מבנה מציעים לעשות חלק מהפרמטרים (C, B, B, delta) תלויים ביצוג הטוקן הנוכחי. התוצאות הזו היא לינארית עם מטריצות נמלדות. וזה עוזר לנו להתחשב בפישט הקלט הנוכחי בצורה יותר טובה. כאמור C, B, מגדרות את האופן בו יציגו הזכרן והפלט עבור הטוקן הנוכחי בהתאם ואז יש לנו סיכוי יותר טוב להצליח במשימות מסווג שתיארתי לפני. בנוסף של delta ביצוג הטוקן הנוכחי מפנה לנו אפשרות לשחק עם קצב דעתכה בצורה יותר גראנולרית שמקנה לנו יכולת "לשכוח" ו"לזכור" איפה שצרכיך.

אבל האם איבדנו את הדואליות שלנו בדרך. מתרברר שלא, הרि המעבר בין יציגו הזכרן של הטוקנים עדין לא תלוי במיקום של הטוקן אלא ביצוג. ככלומר אנו עדין יכולים לחזות מספר טוקנים בו זמן-תא Ci אנו יכולים לחשב את כל הפרמטרים מראש (לא צריך לחשב את המצב הזכרן הקודם באופן מפורש). וכמוון אין צורך להתחשב בכל הטוקנים בתוך חלון ההקשר במהלך ההתקין Ci הזכרן עדין מיוצג על ידי וקטור אחר. אז יש דואליות!

מה שכנן קורה הוא זה החישובים הופכים לקצת יותר מורכבים (שים לב שהחלק הci בעיתי בחישוב שהוא הульאה של מטריצה בחזקה לא השונה Ci A נותרה קבועה). בסוף המאמר מציע כמה שכליים לאופן חישוב המיעילים ומזרדים אותו (המשחקים בין זכרון מהיר ואייטי של GPU).

זה וזה עכשו אתם יודעים מה זה ממנה. נתראה בMoE Mamba עוד כמה ימים.

Review 195: Can Mamba Learn How to Learn? A Comparative Study on In-Context Learning Tasks

<https://arxiv.org/abs/2402.04248>

"אוק", סוקרים מאמר הבא בסדרת מנסה (מה שבא אחריו). בוגיגוד להצהרותי בסוף סקירה הקדמת לא מהי MoE Mamba אלא מאמר אחר. הסיבה היא שלדעתני כמהות המאמרים על MoE היא גדולה מאוד והמאמר הזה רק מציע להלביש אותו על Mamba ללא חידושים מעוניינים אחרים אז החלטתי לדלג.

המאמר שנסקור היום בודק את האם מודלים המבוססים על ארכיטקטורת ממנה על למידת in-context (או ICL). למעשה ICL היא יכולה של מודל לבצע במידה על בסיס כמה דוגמאות בלבד (גמ' נקרא למידת few-shot) ללא שניי של משקל המודל. בגודל יכולת זו של הטרנספורמים לא מאוד מפתיעה Ci "חיזויים" שלהם תלויים ביחסים בין חלק הדטה השונים (טוקנים) באופן מפורש באמצעות מנגןון-h-attention שלהם. כמובן יש מחקרים לא מעטים ומעוניינים שחוקרים את התופעה המרתקת הזו ואני ממליץ לכם בחום להעיף מבט.

לעומת זאת הארכיטקטורה של ממנה לא לוקחת את היחסים בין הטוקנים השונים של הדטה באופן מפורש ודוחשת את ה"עברית" בוקטור אחד איז היכולת שלה לבצע ICL היא פחות אינטואיטיבי. זה אכן פחות קורה. המאמר בדק כמה ארכיטקטורות מבוססות (state-space models) SSM כדוגמת S4 וגם S4-mamba מנגנון h-attention של הטרנספורמים והשו את יכולות ICL שלהם עם ארכיטקטורות הברידיות: ככלומר שילוב של ממנה עם מנגנון h-attention של הטרנספורמים.

איך משלבים ממנה עם הטרנספורמר? המאמר בדק שתי גישות (די דומות). בגישה הראשונה הוא החליף את MLP שיש בבלוקי טרנספורמר אחרי h-attention במנגנון של ממנה. הגישה השנייה (הכי מוצלחת) הנקראת

כמו כן מוסיף מחליפה את הקידוד המיקומי (positional encoding) של עוברים הטוקנים במ מבה נוספת.

כאמור MambaFormer הגיעו לביצועים הטוביים ביותר מכל הארכיטקטורות הלא היברידיות (הטרנספורמר הטהור וכמה וריאנטים של SSM) באופן לא מפתיע בכלל. הררי MLP (ריך 2 שכבות) ממודלים הפעולה די פשוטה -mamba היא למעשה מגנון של זכרון הדוחש את המידע המקורי (בתקווה) של העבר (בטוקנים הקודמים). לא פלא שזה ניצח את כלום.

נשאר לנו רק לציין איזה משימות ניתן למודלים אלו כדי לבחון את יכולות ICL שלהם. אחת המשימות היא לתת למודל כמה זוגות של (x, f) עבור פונקציה f לינארית ולבקש ממנו לחשב $(x)f$ עבור א-ים נוספים. משימה אחרת הייתה לתת לה נקודות שנגדמו מ Gaussian Mixture מסויים ולבקש ממנו לדגום עוד נקודות. טבלה עם כל המשימות מצורפת לפופוט.

נתראה בסקירה מבה הבאה והאחרונה (לא בחרתי עדיין).

Review 196: VMamba: Visual State Space Model

<https://arxiv.org/abs/2401.10166>

מתחלים את הסקירה الأخيرة בסדרת מאמרי מבה. באופן די טבעי המאמר הזה מדבר על שילוב של ארכיטקטורה זה למודלי הראייה הממוחשבת (או ייזן בקצרה). הסקירה הולכת להיות די קצרה וקלילה.

הפעם לא אספק לכם סקירה על ארכיטקטורות מבוססות SSM (היתה זאת לפחות ב 3 סקירות הקודמות). כמו שאתם יודעים הטרנספורמרים השתלטו היום גם על תחום הייזן והחלק הארי של מודלי SOTA בתחום מבוסס על הטרנספורמרים. הטרנספורמרים החליפו את רשתות קונבולוציה(CNN) שלשלטו בתחום הייזן עד 2020 בערך. למרות שיש טענים שבכל מודל ייזן עובד יש או איזה backbone מבועס CNN או שמקיל ריבב כמו attention לקלאי (דרך להתחשב bias-inductive שיש בדתא ויזואלי שאותה מנצלים CNNs), עדין השליטה של הטרנספורמרים בייזן גראית די מוחלטת.

אוקי, אתה זכרם שהמטרה של מבה שהמודלים שקדמו לה היה הדואליות(ראו הסבר מפורט בסקירת הקודמות) המאפשרת אימון מקבילי לצד היסק (inference) מהיר. בגדיל מחליפים את מגנון-חיסון attention שיש בטרנספורמרים במודל מבועס SSM שמאפשר דחיסה של כל הזכרונות(עד טוקן הנחזה) בוקטור אחד ובכך אפשרים היסק מהיר (וגם אימון מקבילי).

از למה לא נעשה את אותו הדבר עבור הדטה היזואלי? זה בדיק מה שהמאמר מנסה לעשות. למעשה המאמר משלב CNNs (הרי נפטרים מהטרנספורמרים) עם מגנון דחיסת הזכרונות המבועס SSM. מה שקצת משעשע שבמהלך האימון מודלי בסגנון מבה מופעלים דרך קונבולוציה ארוכה (הוסבר בסקירות הקודמות בהרחבה) אז קיבלנו בסוף רשת קונבולוציה טהורה (פחות באימונו).

מה הבעיה העיקרית עם הכנסה של קצת-SSM -ים למודלים ויזואליים? היכוון!! הררי עבר שפה טבעית וגם עבר אודיו די ברור שעבור טוקן נתון צריך "לזכור" את הטוקנים מתחילת הטקסט/אודיו. בתמונה לטוקן נתון (פאיץ') ניתן לבנות זיכרון מכיוונים שונים (הסדר חשוב במ מבה וב-SSM-ים אחרים). אפשר להתחיל מלמעלה, או מלמטה שלה תמונה, לכת ימינה או שמאליה. לא ברור מה הכי טוב מבחינת ביצועים ואז VMamba משלב אותם. לפחות

נתון מתחילה מפוץ' השמالي והעליון והפוצ'ים נכנסים ל-SSM בכיוון ימין-מטה. משלבים את ה-SSM זהה עם ה-SSM שנבנה החל מהפוץ' התיכון מימין כאשר ההפוך הפוצ'ים נכנסים ל-SSM מכיוון שמאלי-למעלה. ככה בונים את החלק ה-SSM של VMamba. מעניין לזכור שהמאמר נקרא מבה הוא טוען שהוא משתמש בארכיטקטורה של S6 שקדמה למבה (הבדל הוא תלוות של מטריצות B ו- C של SSM בייצוג הטוקן הנוכחי).

שאר הדברים הדיא סטנדרטיים: חלוקת תמונה לפוצ'ים, הפעלה כמה סובי downsampling המבוצע עם בלוקים המכילים קובולוציות SSM, 3x3, חיבור resnet וכמה שכבות לינאריות. זהה זה, סוף הסקירה מקווה שנחנכתם לקרוא את סדרת מבה...

Review 197, Short: LLM4Decompile: Decompiling Binary Code with Large Language Models

<https://arxiv.org/abs/2403.05286>

המאמר מציע LLM4Decompile, משפחה של מודלים LLM לדיקומפלציה בגישה פתוחה שנעים מ-B1 עד B33 פרמטרים. מודלים אלו מאומנים על 4 מיליארד טוקנים של קוד מקור בשפת C וקוד אסמבלי מתאים. המחברים גם מציגים את Eval-Decompile-Eval, הדאטאטס להערכת דיקום הדיקומפלציה המבוצע על ידי מודל (מקמלים חדש ובודקים את הפונקציונליות של הקוד).

LLM4Decompile מצליח לבצע דיקומפלציה בצורה מדויקת 21% מקוד האסמבלי, עם שיפור של 50% ביחס ל-4-GPT. מודל שפה לקימפול ולדקמפלול של קוד נראים כמו תחום מחקר חשוב במיחש לבניית סוכני AI חסינים יותר נגד התקפות אדווורסריות שיפלו בשכבות העמוקות יותר של Software Stack.

Review 198: Improving Text Embeddings with Large Language Models

<https://arxiv.org/abs/2401.00368>

הסקירה זו תהיה לא סטנדרטית ואתחל אותה מ שאלה: למה בחרתי לסקור את המאמר זהה?

לא בಗל שמדובר במודלי שפה - הרי כל יום יוצאים عشرות מאמרים על LLMs. גם לא בಗל שהמאמר הזה מציע שיטה לשיפור ביצועים של פתרונות (hacks) (Retrieval Augmented Generation (RAG)). הסיבה האמיתית היא שימוש בטכניקה שאני מאד אוהב הנקראת (CI) או במידה ניגודית בשפת הקודש.

בזמן האחרון אני לוצרר לא רואה יותר מדי עבודות שימושísticas בפרדיגמה היפה זו שתיארתי אותה בהרחבה בלי כמעט סקירות בשנים האחרונות. בד"כ משתמשים בגישות השונות של הלמידה הניגודית כדי להפיק "צוגי" דатаה עצמאיים במרחב בעל מימד נמוך או embeddings. "צוג" דатаה עצמאי - הוא יציג שמליח "לשמור" על פיצרים האינרנטיים של פיסת דатаה, ככל מרצהה שדוחש את הדטה בצורה עיליה.

air השיטה זו עובדת? העיקרון הוא די פשוט - לכל פיסת דטה יוצרים פיסת דטה קרובה (נגיד סמנטיית בשפה הטבעית או שתי פוצ'ים של אותה תמונה בראייה הממוחשבת). לאחר מכן יוצרים מספר של זוגות של פיסות דטה שהן לא קשורות אחת לשניה (נגיד פוצ'ים מתמונות שונות). בגודל מאוד המטרה של CI היא

למבנה (= לאמן מודל המפיק אותו) יציג הדата הממצער מרחק בין פיסות דатаה דומות(זוגות חיובים) וממוקסם אותו בין פיסות הדата הלא דומות (זוגות שליליים). לעיתים משתמשים במרחב קוויין, לפעמים מרחק אוקלייד' יש עוד וריאנטים; יש מגוון שיטות לבנות זוגות חיוביים ושליליים אבל העיקרן נותר על כן

אחד המאמרים הראשונים שהשתמשו בلمידה הניגודית בהקשר של בניית ייצוג דатаה הוא [InfoNCE](#) של Oord ושותפיו. לאחר מכן השיטה שימשה מחברים של מאמרם מאד מפורטים כמו [MoCo](#)-[SimSCE](#). עכשו אתם בוחח רוצחים לשאול איך CI קשור למאמר שנסקורו היום? כמובן כדי לשפר את האמצעים של הטקסט עבור משימות RAG.

ל-RAG לרוב יש שתי בעיות משמעותיות:

1. לא תמיד טקסט בעל ייצוג קרוב (בד"כ לפי מרחק קוויין) לייצוג השאלה מכל תשובה על השאלה (או שהוא שנייתן לגוזר ממנו תשובה). יתרון מאוד שהtekst שייר לאותו התחום(דומיין) אבל לא מכל תשובה על השאלה.
2. המרחב של ייצוגי הטקסטים (המופקים על ידי LLM-ים עצמותי) הוא מרחיב לא טריוויאלי מבחינת המרחק בין הייצוגים ולפעמים מרחקים בין זוגות טקסט קשורים ולא מאוד קשורים עלולים להיות קרובים זה לזה. זה אicasר אתכם בוחרים פיסות טקסט עם ייצוג הקוראים ביוטר לייצוג השאלה לא תמיד מקבלים פיסות טקסט רלוונטיות.

از הפתרון הגיוני ביותר הוא לכיל (fine-tune) מודל שפה על דאטasset שאלות ותשובות מדויין (משימה) כך שיקרב את ייצוגים של שאלות ותשובות רלוונטיות וירחיק את הייצוגים של השאלות והתשובות הלא רלוונטיות. הבעה הגדולה עם הגישה הזו היא שבניהו ידנית של דאטasset כזה היא יקרה ו לוקחת הרבה זמן.

از למה לא לרטום LLMs למשימה הזו? זה בדיק מה שעשו המחברים של המאמר המסורק. הם ביצעו LLM מצוי ליצור זוגות של שאלות עם תשבות נכונות ובנוסף גם זוגות עם תשבות לא נכונות, אבל נראות "דומה לנכונות" (hard negatives). היה שם הנדסת פרומפטים חמודה אבל לא ממשו מהפכני במיוחד.

ולבסוף השתמשו בגישה CI סטנדרטית כדי לכיל מודל שפה בבדיקה באופן הסברתי לפניכן. לא מצאת במאמר איך בבדיקה מתבצע האימון (הואיפו שכבות או אימנו כמה מהן) אבל הרעיון ברור.

זה וזה....

Review 199: LLM2Vec: Large Language Models Are Secretly Powerful Text Encoder

<https://arxiv.org/abs/2404.05961>

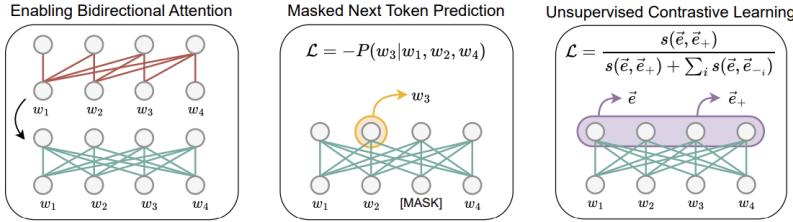


Figure 1: The 3 steps of LLM2Vec. First, we enable bidirectional attention to overcome the restrictions of causal attention (**Bi**). Second, we adapt the model to use bidirectional attention by masked next token prediction training (**MNTP**). Third, we apply unsupervised contrastive learning with mean pooling to learn better sequence representations (**SimCSE**).

המאמר זהה תפס את תשומת ליבו עקב העובדה שהוא דין בנושא שמאוד מעניין אותו לאחרונה (בנוסף למ מבה וחידושים למודלי דיפוזיה 😊). והנושא הזה הוא התאמת מודלי שפה מאומנים לביצוע משימות דיסקרימינטיביות, למשל משימות זיהוי נושא או סנטימנט, זיהוי חלקו ובודמה. הרוי רוב מודלי שפה בתקופה האחורה מאומנים לגנרטט טקסט, כמו לבצע משימה גנרטיבית(מבוססים על דקORDER בלבד).

אתם יכולים להגיד מהו ציריך מודלים למשימות דיסקרימינטיביות אם ניתן די בקהלות להפוך הרבה המשימות דיסקרימינטיביות לגנרטיביות? למשל משימת זיהוי סנטימנט ניתן להחליף במשימת גנרטיבית של גנרטוט הסנטימנט לטקסט נתון (כלומר "הסתימנט בטקסט זה היה חיובי"). אבל נשאלת השאלה האם הchèלה זו היא אופטימלית מבחינה הגדול, הביצועים והמאץ הנדרש לאימון מודל כזה למשימה נתונה. בלי כמעט מקרים (למשל כאשר יש דרישות קשותות לצריכת זיכרון או ליטנסי מקסימלי של המודל).

האם אפשר לעשות יותר טוב? כאמור רוב המודלים החזקים שייצאו ב-3 השנים האחרונות הם מודלים גנרטיביים בעלי ארכיטקטורת הדקORDER (gpt, gemini, claude etc). המודלים שאומנו למשימות דיסקרימינטיביות בעלי ארכיטקטורה הכללת אנקודר הפכו להיות די נדירים לאחרונה. לאור זה המאמר שנסקורו היום מנסה להתאים (לכайл) מודל שפה גנרטיבי (דקORDER) למשימות דיסקרימינטיביות.

עכשו נשאלת השאלה למה לא לקחת מודל שאמן דקORDER ושר לעשות לו פינטיין (fine-tune) למשימה דיסקרימינטיבית? כדי להבין למה זה עלול להיות לא אופטימלי צריך להרחיב טיפה על איך בדיק מאומנים מודלי אנקודר ומודלי דקORDER.

במהלך אימון האנקודר אנו ממסכים טוקנים מסוימים ומאמנים את המודל לחזות אותם. ככלומר אנחנו משתמשים בכל הטוקנים בטקסט כדי לחזות את הטוקנים המקוריים. אם הدادהסת שאמנו מאומנים עליו גדול ומוגן מספיק המודל לומד "להבין" (לאפין סטטיסטי) את השפה. לעומת זאת מודל הדקORDER הינו מודל גנרטיבי כלומר המודל יוצר פיסות דעתה חדשות. זה מצריך איפון אימון שונה מהאנקודר. הדקORDER מאמן לגנרטט דатаה חדש: המודל מאומן לחזות את המילה (טוקן) הבא. ככלומר להבדיל מאיפון אימון האנקודר אנו **מסתירים מהמודל את הטוקנים שבאים אחרי הטוקן הנחוצה**, ככלומר חוסמים ממנו את העתיד.

מכאן ניתן לראות עקב אוף אימון שונה קשה וקצת נאיי לצפות מהמודלים שמאומנים דקORDERים להציגן במשימות דיסקרימינטיביות אחרי פינטיין (אני לא טוען שההבדקה אפשרי וכונראה יש משימות שעבוריהם לא רע, כמובן זה תלוי בכמה דטה מתוויג יש). נגד למשימה זיהוי של חלקו ובודמו היצוג של מילה במודל הדקORDER המאומן (pretrained) לוקח בחשבון רק את המילים הקודמות שכמובן לא אופטימלי עבור משימה זו.

אחרי הקדמה ארוכה זו בוא נתמקד במאמר המסורק. כאמור הוא מציע דרך להתאים מודל דקORDER מאומן למשימות דיסקרימינטיביות. המאמר מציע 3 שלבים לה'פיכה' של מודל דקORDER למודל האנקודר:

1. ביטול איפוס הטוקנים העתידיים במנגנון attention של המודל חופשי לנצל את כל הטוקנים לבניית ייצוג של כל טוקן . ד"א המאמר טוען הביצועים של המודל לאחר מכון יורדים (בגלל זה יש עוד 2 שלבים בתהילר).
2. במהלך האימון במקום לחזות את הטוקן הממוסך מייצגו ההקשרי (contextualized) אנו עושים זאת מייצגו של הטוקן הקודם. לא ברור לי ב 100% מה היגיון מאחורי זה.
3. שימוש בلمידה ניגודית (contrastive learning). גישות למידה ניגודית משמשות לאימון של ייצוג דата (לא מתייג בד"כ) כאשר מטרת האימון לקרב ייצוגים של פיסות דataset קרובות ולהרחק ייצוגים של דataset לא דומות/לא קשורות (מבחינת דמיון קוויין). אז המאמר מציע לאמן את המודל לקרב ייצוגים של אותו המשפט עם drop-outs שונים (בגדול מודול dropout הוא מעשה איפוס קשרים/משקלים בין נוירונים שונים במודול. לעומת זאת ייצוגים של משפטים מאומנים להיות רחוקים אחד מהם למרחב אמבריגן).

לטענת שילוב שלבים אלו הופך את המודל שלכם לאנדוור המסוגל לרופיק ייצוגים דата חזקים המפיגנים ביצועים לא רעים בכמה ממשימות דיסקרימינטיביות.

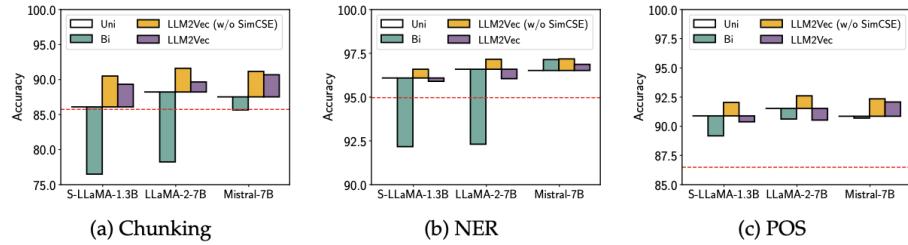


Figure 2: Evaluation of LLM2Vec-transformed models on word-level tasks. Solid and dashed horizontal lines show the performance of Uni and DeBERTa-v3-large, respectively.

Review 200: SiMBA: Simplified Mamba-based Architecture for Vision and Multivariate Time series

<https://arxiv.org/abs/2403.15360>

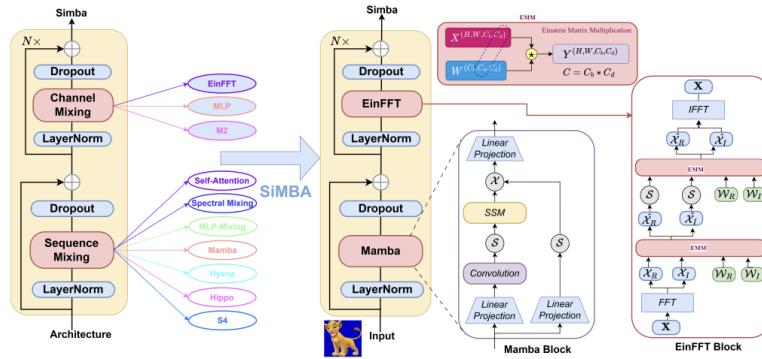


Fig. 1: Simplified Mamba Based Architecture.

המאמר הזה משר את תשומת לבי כי מצד אחד יש בו שימוש נרחב בהתרמת פוריה ויביצוגים של דата בתחום התדר. החולשה שלי בתחום התדר נבעת מכך שביליתי כמה מהשנים הראשונות של הקריירה בתחום עיבוד של

אותות אלחוטיות. מצד שני המאמר גם משתמש בארכיטקטורת מבנה שסקרטט' בהרבה בחודשים האחרונים (וכנראה אמשיך עם זה כי מאמרם מעוניינים בנושא מרתך זה לא מפסיקים להגיע).

אוקי', אז מה יש לנו במאמר זהה? המאמר מציע שדרוג יפה לארכיטקטורה של מבנה המערב כאמור התמורות פוריה וקצת משחקים בתחום התדר. הארכיטקטורה המוצעת מתאימה גם לדאטה ויזואלי וגם לסדרות זמן multivariate. המאמר כתוב בצורה די מסורבלת והיה לי לא טריויאלי לגלוות מה הם באמת עשו עקב הסברים וסימונים לא ברורים. אבל כאמור הרעיון מאחורי המאמר הוא די חמוד.

המחברים מנסים לשפר את מבנה על ידי הוספת שכבה שבגדול לוקחת את הייצוגים המופקים על ידי מבנה ו"מצפקת" אותם על ידי פלטור תדרים מסוימים מהם (הייצוגים). קודם כל נציג שמעילים את המנגנון המוצע, שקיבל שם EinFFT, על כל איבר סדרה בנפרד (פאנ' של תמונה) בצורה ממוקבלת. כאמור הטיספור מתחילה מהפעלת התמרת פוריה על הפלט (=ייצוג פיסת דאטה) של שכבת מבנה. ואז המאמר הופך להיות די לא ברור ודבר הזה גזל מני בערך שעوتאים כדי להבין שלא אני מפספס משהו אלא המאמר עצמו קצת לא מד'יק (בתקווה עמדתי במשימה זו).

כאמור הרעיון הוא מפלטן תדרים(תלוויות) הלא נוחצים (לביצוע המשימה) בייצוג איברי הסדרה. הפלטור מתבצע במחרב הייצוג של הדאטה (כלומר אמבידג') ונקרא channel-mixing. ככלומר שכבה זו היא משמשת בתור תוסף/חלפה ל-MLP שלפעמים משמשת לאותה המטריה.

אבל איך הוא עושה את זה משתנה בין נוסחה לבין נוסחה במאמר. במאמר עצמו (נוסחה 4) קודם כל מפעלים שכבה לינארית במישור המרוכב ולאחריה סיגמוד (גם במישור המרוכב). ב-`append` (בתחילה עמוד 22) זה כבר מופיעה שכבה אחת של ReLU, לאחר מכן עוד שכבה לינארית, לאחר מכן מפעלים פונקציית `softshrink` ולאחר מכן פונקציית `lambda`. ככלומר שטח λ עבור ReLU ושהז' כאלו מעבר לזה ב-`lambda`. ככלומר איזה `stop-band filter` מוזז.

הגרסה השלישייה מגיע מהדף האחרון של ה-`append` שם יש רק `ReLUs`. לא הסתכלתי בקוד אז לא ברור מה באמת קורה שם. כל הפעולות הללו מתבצעות בצורה נפרדת במישור ממשי ובמישור המדומה לאחר מכן משלבים אותם. בשלב האחרון מבצעים התמרת פוריה הפוכה (IFFT).

אוקי', אז בואו נחזור לעיקר. המנגנון שבא אחריו שכבת מבנה נקרא EinFFT וכבר הבנו ש- FFT מתאים להtamרת פוריה. אבל מה זה `Ein`? באופן לא מפטיע ALSO 3 האותיות הראשונות נלקחו מאינשטיין. אז מה בעצם איינשטיין עושה כאן?

למעשה המאמר משתמש בסכימת איינשטיין שהוא דרך לרשום מכפלות הטמורות או המטריצות במקרה פרט. למעשה במקומות לרשום כל איבר z של המכפלת מטריצות A - B בתור מכפלה פנימית של שורה ? ועמודה j סכימת איינשטיין כתובה אותו ללא סימן של סכום(=סיגמה) אלא על ידי ציון של מספר שורה ?, מספר עמודה j. ואינדקס סכימה k.

از איך המאמר משתמש בסכימה זו? הר' אמרתי שהסכמה זו מוגדרת גם לטמורות ומתרבר שלחbillות תוכנה כמו pytorch יש חבילות שיודעות לבצע מכפלת טנזוריים רב ממדיים המבוטאים דרך סכימת איינשטיין בצורה די יعلا. זה בדיק מה שעושים במאמר. המאמר מפרק את המטריצות מהשכבות הלינאריות של EinFFT לכמה מטריצות ביממד נמוך יותר ובונה מזה טנזור רב ממד' הבניי ממטריצות בלוקיות (אפסים מחוץ לבлокים). הטענה במאמר זהה מאפשר לבצע את המכפלות (בaimon אבל כמובן גם באינפרנס) בצורה מהירה יותר על ידי ניצול טוב יותר של משאבי החומרה.

Review 201: ZigMa: A DiT-style Zigzag Mamba Diffusion Model

<https://arxiv.org/abs/2403.13802>

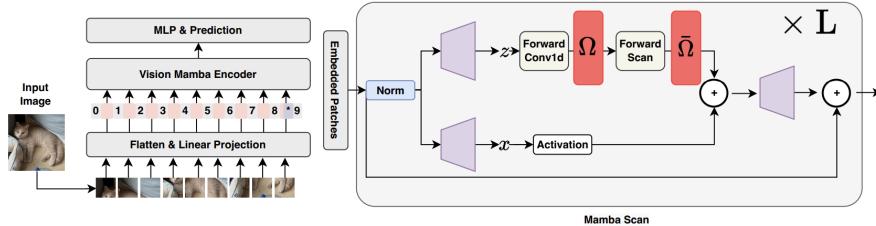


Figure 2: ZigMa. Our backbone is structured in L layers, mirroring the style of DiT [65]. We use the single-scan Mamba block as the primary reasoning module across different patches. To ensure the network is positionally aware, we've designed an arrange-rearrange scheme based on the single-scan Mamba. Different layers follow pairs of unique rearrange operation Ω and reverse rearrange $\bar{\Omega}$, optimizing the position-awareness of the method.

המאמר זהה משך את תשומת ליבי מכמה סיבות:

- יש מודלי דיפוזיה - האהבה הקודמת של שבקروب מואוד אוחד את הקשר איתם
- יש כאן (zs) SSMs (State-Space Models) בדמות Mamba - האהבה הנוכחית של שתיכף אני מסיים להcin עליה מצגת די רצינית ובתקווה ישמעו אותו מציג אותה בפורומים השונים
- המאמר פורסם בראשון לאפריל ובהתחלה קצר חמדתי 😊

בנוסף יש במאמר גם קצר מהטרנספורמרים (cross-attention) שעוד מוסיף לשலמותנו. אוקי, אז מה יש לנו במאמר הזה מעבר לכמה מילים "באזיזות". המאמר מציע ארכיטקטורה מעניינת המиועדת לגינורוט תמונהות וגם ידאו. כאמור הארכיטקטורה היא שיכת למשפה של מודלי דיפוזיה גנרטיביים אבל מכילה חלקים המורכבים מ-Mambas (ממבא) בנוסף ל-cross-attention הלב של הטרנספורמרים. ויש כאן חידוש מעניין לגבי הסדר שבו מכניםים פאצ'ים של תמונהות (או פרימים של ידאו) במהלך אימון המודל.

נתחל מהסביר קצר על מודלי דיפוזיה גנרטיביים. בהינתן DATASET (של תמונהות או/ו סרטוניים) אנו מאמנים את רשת באופן הבא:

- מוסיפים כמויות קטנות של רעש גausi לפיסט דאטה עד שהיא היא הופכת לרעש טהור
 - מאמינם רשת נירונים (עם Mamba ו-cross attention במקורה שלנו) כדי למדל של התהיליך ההפוך.
- כלומר מפיסט דאטה מושעת מאייטרציה ח לחזות אותה באיטרציה 1-ח.

כאשר יש בידינו מודל זהה אנו למעשה מסוגלים לנרטת תמונה מרעש גausi טהור בצורה הדרגתית, איטרציה אחריו איטרציה. עם השנים צצו שיטות רבות ומגוונות מאוד לאריך להוסיף רעש ומה לבדוק כדי לחזות עם הרשת שלנו.

בשנה וחצי האחרונות היו כמה חידושים מעניינים במודלי דיפוזיה ומכוון שהמאמר משתמש בהם אני חייב לספר לכם בגודל כמה מדובר (כאמור הולך לדבר על זה בהרחבה בסקירות הבאות).

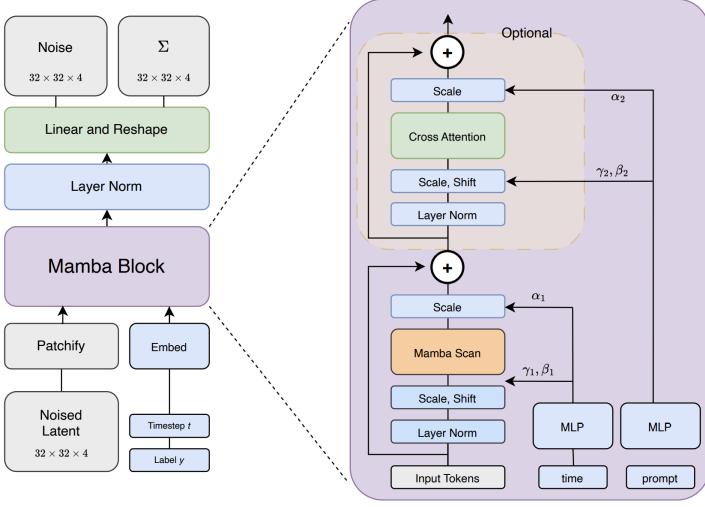
לאחרונה יצאו כמה מאמרים מעוניינים (למשל <https://arxiv.org/abs/2210.02747> ו- <https://arxiv.org/abs/2303.08797>) אבל יש עוד عشرות אחרים) המכילים מודלי דיפוזיה לתהיליך רציף של מיפוי התפלגות פשוטה (כגון גאוסית סטנדרטית) להתפלגות של הדטה (התפלגות המורכבות). תהיליך זה נקרא זרימה רציפה (flow continuous) הדיסקרטיזציה שלו (במישור הזמן כלומר האיטרציות) היא מודל דיפוזיה גנרטיבי עבור מיפויים מסוימים. יש לנו צורך לבחור את המיפוי (זרימה) בין ההתפלגות דעתה להתפלגות הפешטה ויש לא מעט מחקרים על איך לבחור אותו בצוורה אופטימלית (למקסם את איקוֹת הדטה המוגנרטת, ליצב את התהיליך, ליצור מיפוי כמו שיטור פשוט או ישיר וכדומה).

از איך כל המתמטיקה זו קשורה לגנרטוט דעתה? אז יש כאן עוד קצת מתמטיקה שנוצרה לצלול בה. בגדול הזרימה הרציפה בין להתפלגות הפешטה להתפלגות הדטה (לפעמים נקראת reverse-time) ניתן לתאר על ידי משואה דיפרנציאלית סטוכסティות (SDE) שמכילה:

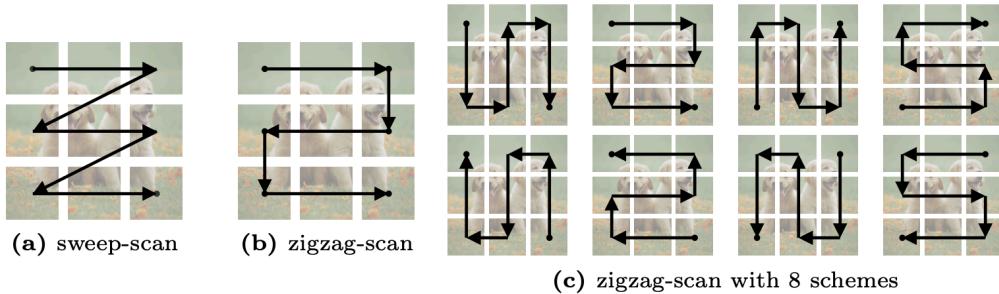
1. הדטה המורעש עצמו x
2. מהירות או השتنות(נגזרת בזמן) של הזרימה בזמן $(x)_t'$ (תחשבו על זה כמו על תנועה במרחב בין t ו- s עננים של נקודות שניתן להגדיר אותה על ידי מהירות הциינית ונקודת התחלת או הסוף).
3. פונקציית $(x)_t(s)$ שהיא בעצם לוגריתם של $(x)_s - (x)_t$ פונקציית ההתפלגות של הדטה המורעש
4. יש גם תהיליך reverse-time Wiener המהווה את החלק אקראי (סטוכסטי) ב-SDE זהה

אז מה אפשר לעשות עם ה-SDE זהה, למה צריך אותו? מתרברר כי עבור פרמטרים של הזרימה בין ההתפלגות הדטה להתפלגות הפешטה ניתן לנסח בעיות אופטימיזציה המאפשרות שערור של $(x)_s - (x)_t$ בהינתן דטה לאימן. אחרי שנשעך אותו ניתן לפתור את ה-SDE שדיברנו עליו נומריית (נגיד בשיטת אולר-מראימה) כולם מנוקודת התחלת הנדגמת מההתפלגות הפешטה (גאוסית) נוכל לגנרט דעתה צעד אחריו צעד. וזה בדיק מה שעושים במאמר.

אוקי, שרדנו את המתמטיקה - עכשיו מה הקשר ל-SSMs-caus? בשביל כך צריך להזכיר ארכיטקטורה של מודל או T, DiT, או Diffusion Transformer הבוסס המבוסס המבוסס של SoRA של OpenAI. למעשה DiT מרכיב מбалוקים של טרנספורמרים שמרתם היא למדל את הפרמטרים $(x)_s - (x)_t$ (כמובן לאחר דיסקרטיזציה במישור הזמן, כלומר איטרציות). המאמר המסורק מחליף את בלוקי הטרנספורמר ב-Mamba-b-attention (בנוסף הם גם לוקחים cross-attention cross-shape הלב של הטרנספורמר אך לפי הציור שלהם החלק הזה הוא אופציוני).



אבל כאן יש לנו בעיה. מכיוון שմבבה היא ארכיטקטורה מיועדת לסדרות בעלת מימד הזמן חד-מימדי (למשל טוקנים של טקסט) כאן יש לנו תמונות ובנה קיימים קשרים דו-מימדיים בין הפעצים (טוקנים ויזואליים) בתמונה וקשרים תלת ממדיים בוידאו (בנוסף בין הפריימרים). המאמר מתאר את המבנה של SSM עבור הקלט בעל קשרים רב ממדיים על שילוב של s-SSM שכל אחד מקבל את הקלט בסדר שונה (תראו בתמונה). לעומת שכבות של מمبבה מווערמות (stacked)ichert מעל השניה כל הקלט נכנס לכל אחד מהם בסדר שונה (למייבט הבנתי כל המ מבבות עובדות עם אותן מטריצות הפרמטרים A, B, C). זה מאפשר לנו ZiGMA (A, B, C) להתחשב בקשרים אלו. המאמר מרחיב את הגישה הזה לגנרטוט וידאו (עבור קשרים תלת-ממדיים).



אצין שבדומה ל-TiDi 모델 המוצע פועל במרקח הלטוני כמו דיפוזיה הוא יציג לטוני של הדאטה אחרי האנקודר. TiDi משתמש באנקודר ובדקורר של VAE (אחד השכלולים שלו) אך במאמר זה לא הצליח להבין האם המחברים לקחו VAE. במקומ אחד במאמר רומזים שהאנקודר גם מכיל SSM אבל לא מצאתי לדקה אזכורים נוספים.

התוצאות נראה לא רע, לשנות 2020 ככה אבל מכיוון שהוא אחד המאמרים הראשונים המשלבים SSM ומודול דיפוזיה נסלח להם על כן.

יצאה סקירה ארוכה אבל מובנת פחות או יותר בתקווה...

Review 202: SimPO: Simple Preference Optimization with a Reference-Free Reward

<https://arxiv.org/abs/2405.14734>

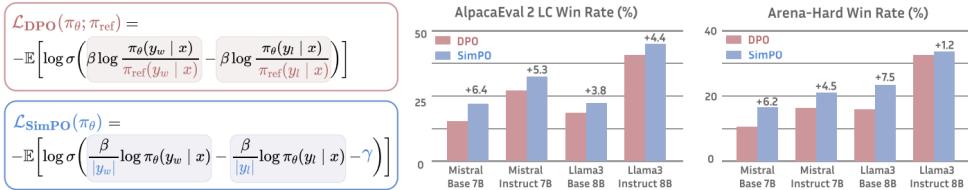


Figure 1: SimPO and DPO mainly differ in their reward formulation, as indicated in the shaded box. SimPO outperforms DPO across a wide range of settings on AlpacaEval 2 and Arena-Hard.

המאמר שנסקרו דן בנושא אימון של מודלי שפה. אתם בטוח יודעים שאימון מודל שפה alfafoundational מורכב מ- 3 שלבים עיקריים:

1. אימון מודל self-supervised על דאטסהט ענק
2. אימון(fine-tuning) מפוקח (supervised fine-tuning) או SFT על דאטסהט מתואג קטן יותר (בד"כ מכיל תשובות רצויות למגוון שאלות) במטרה למודל לעקב אחריו הוראות המשמש (following)
3. שלב RLHF: מתרבר שרוב המודלים לא מצליחים ללמידה רק מהתשבות ה"טובות" ואנו נדרשים לספק לו גם את התשובות ה"לא טובות". השלב האחרון נעשה באמצעות שימוש בטכנייקות שונות של למידה עם חיזוקים.

המודלים הראשונים (גוגל, OpenAI) שהשתמשו ב-RLHF ליישור (alignment) של המודלים התבוססו על טכנית שנקראת PPO (Proximal Policy Approximation) או RLHF (Reward Function Alignment). במהלך האימון אנו מעדכנים את המודל שלנו כך שהוא ייתן תגמול (=reward) גבוהה לתשובה ותגמול נמוך לשובנה לא טוביה תוך שמירה של המודל החדש קרוב (מבחינת התפלגיות הטוקנים) שהוא מוציא להתפלגות המתקבלת בשלב 2.

אבל איך נמדוד את התגמול זהה? עבור PPO אנו צריכים לאמן מודל תגמול שבහינתו פרומפט ותשובה יחזיר לנו ציון (סקלר). עבור תשובה טובה הציון יהיה גבוה ועבור תשובה לא טובה הוא יהיה נמוך. מאמנים את המודל הזה על הדאטסהט של התשובות הטובות ולא טובות משלב 3.

כמובן שאם היה אפשר להסתדר ללא מודל תגמול מצבונו היה טוב יותר. קודם כל זה חוסך לנו את זמן ומשאבים ובנוסף אנו לא צריכים להפעיל אותו לאינפראנס במהלך אימון RLHF זהה גם יכול להפחית את דרישות הזיכרון וכוח חישוב. אך הוצעה שיטות כמו DPO (Direct Preference Optimization) או ORPO (סקרינו אותו באנגלית לפני כחודש) הסתדר גם בלי להשתמש במודל תגמול. לאחר מכן יצא מודל הנקרא ORPO (סקרינו אותו באנגלית לפני כחודש) הסתדר גם בלי להשתמש במודל משלב 2 במהלך האימון (משמש רק לאתחל המודל משלב 3).

עכשו הגיעו למאמר המסורק. הוא הציע שכלול ל-DPO הנקרא SimPO. כמו OrPo הוא לא צריך מודל רפרנס ב佐ורה מפורשת במהלך אימון שלב 3 ומצביע לאמן את המודל על ידי מKeySpec ההפרש בין התגמול של התשובה הטובה והתשובה הלא טובה (עם הסיגמואיד) עם איזשהו מרג'ין מסוים. החידוש העיקרי של המאמר שבתוור פונקציית תגמול המחברים לוקחים את הנראות המירבית של תשובה בהינתן שאלה, המנורמלת באורך התשובה (בטוקנים). המחברים טוענים שדבר זה (נורמל) בין השאר מונע מהמודל לגנרט תשבות ארוכות מדי וזה אכן נשמע די הגיוני.

דרך אגב בוגר להמרא'ין נטען המאמר שמספר עבדות קודמות ציינו שהMargin מיטיב עם תהליכי האימון (למרות זהה ד' הוספה קבוע).

"The margin between two classes is known to influence the generalization capabilities of classifiers [1, 9, 19, 27]. In standard training settings with random model initialization, increasing the target margin typically improves generalization".

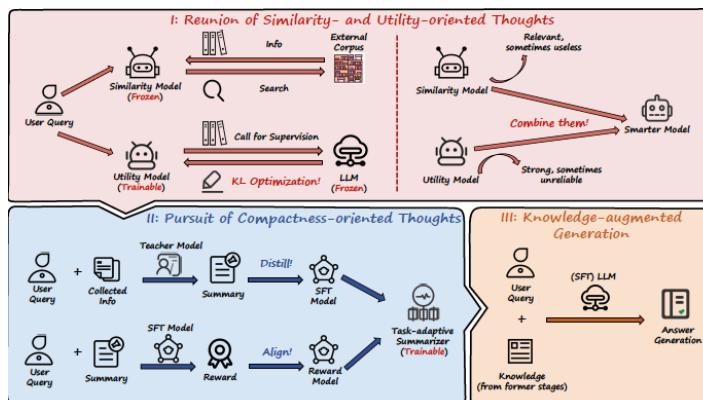
יש גם את הטבלה החמודה זו המסכםת את רוב החוקרים האחרונים בתחום RLHF למודלי שפה.

Table 3: Various preference optimization objectives given preference data $\mathcal{D} = (x, y_w, y_l)$, where x is an input, and y_w and y_l are the winning and losing responses.

Method	Objective
DPO [62]	$-\log \sigma \left(\beta \log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - \beta \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} \right)$
IPO [6]	$\left(\log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} - \frac{1}{2\tau} \right)^2$
KTO [25]	$-\lambda_w \sigma \left(\beta \log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - z_{\text{ref}} \right) + \lambda_l \sigma \left(z_{\text{ref}} - \beta \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} \right)$, where $z_{\text{ref}} = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\beta \text{KL}(\pi_\theta(y x) \pi_{\text{ref}}(y x))]$
ORPO [38]	$-\log p_\theta(y_w x) - \lambda \log \sigma \left(\log \frac{p_\theta(y_w x)}{1-p_\theta(y_w x)} - \log \frac{p_\theta(y_l x)}{1-p_\theta(y_l x)} \right)$, where $p_\theta(y x) = \exp \left(\frac{1}{ y } \log \pi_\theta(y x) \right)$
R-DPO [60]	$-\log \sigma \left(\beta \log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - \beta \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} - (\alpha y_w - \alpha y_l) \right)$
SimPO	$-\log \sigma \left(\frac{\beta}{ y_w } \log \pi_\theta(y_w x) - \frac{\beta}{ y_l } \log \pi_\theta(y_l x) - \gamma \right)$

Review 203: Similarity is Not All You Need: Endowing Retrieval-Augmented Generation with Multi-layered Thoughts

<https://arxiv.org/abs/2405.19893>



בזמן האחרון גישות המשלבות מודלי שפה עם בסיסי נתונים חיצוניים הפכו למאוד פופולריים. גישות אלו לרוב שייכות למשפחה Retrieval Augmented Generation או RAG בקצרה. בגין בהינתן מודל שפה ומסמכים העשויים להכיל תשובה על שאלת משתמש, RAG קודם מחפש כמה מסמכים הרלוונטיים ביותר לשאלה ואז מזינה אותם יחד עם השאלה למודל שפה. המודל מרכיב את תשובתו על השאלה בהתאם שהוזנו אליו.

אבל איך נבחר מסמכים הרלוונטיים יותר לשאלה? בדרך כלל בוחרים אותם לפי הקربה של האמביינט (= "יצוג וקטורי") שלו לאמביינט של השאלה. בדרך כלל הממציאות טיפה יותר מורכבת ממה שתיארתי: למשל אם המסמכים ארוכים צריך לחלק אותם לציאנקים אז הבחירה היא לפי דמיון האמביינט של הציאנקים לזה של השאלה. כמובן שיש עוד גישות.

הדמיון בין אמביינט לדמיון קויין (זווית בין הווקטוריהם). האם הבחירה זהו היא אופטימלית - זו השאלה שהמאמר שנסקור היום מנסה לענות עליה.

כדי להבין האם הבחירה אופטימלית צריך להגיד מدد אופטימליות. הרעיון בסופו של דבר מטרתנו היא לתת תשובה נכונה לשאלת המשתמש. המאמר טוען שבבחירה מסמכים רלוונטיים לפי דמיון אמביינט אינו אופטימלי' בהתאם המدد הזה. אז המחברים מציעים גישה לשכלול הבחירה של המסמכים הרלוונטיים לשאלה.

האמת הם מציעים משהו די טבעי - בגין המטרה שלהם היא לאਪטם את הביצועים של RAG דרך "מקסום הסיכוי לקבל תשובה טובה אחריו בחירת מסמכים רלוונטיים על ידי RAG". המחברים מנסים להשיג את המטרה בכמה שלבים:

שלב 1: אימון מודל utility. המטרה של מודל זה להעניק ציון ליכולת של מסמרק נתון "لتת' תשובה טובה לשאלה כאשר הם (המסמרק והשאלה) מזונים למודל שפה ייחד. אבל איך נדע לשערך את איכות התשובה? בשביב זה המחברים לוקחים מודל שפה חזק (gpt4) שמטרתו היא לתת ציון לתשובה עבור מסמרק ושאלה נתונים (כל שההתשובה טובה ציון גבוה יותר). המאמר לא מסביר איך זה נעשה אבל אני מאמין שעבור דאטאסתט המכיל תשובות ניתן למדוד דמיון סמנטי בין תשובה אמיתי לתשובה מופקת על ידי Who (כלומר בין האמביינט), גם למדוד אותה על ידי הזרותם של המסמרק, השאלה והתשובה ל-Who ומידדת נראות מירבית שלה (כלומר logits), בטח יש עוד שיטות. המחברים ממשנים model utility (שהוא מודל קל יחסית) להחזיר את אותה ההתפלגות של ציוני מסמכים (בהינתן שאלה) כמו המודל החזק. כלומר ממצערים divergence KL בין התפלגות ציוניים של utility model לבין זו של מודל השפה (שהוא מוקפא - לא מואמן).

שלב 2: בחירת מסמכים עבור שאלה נתונה בוחרים רק מסמכים שיש להם ציון דמיון או ציון של model utility מוסף (בין k הגבוהים ביותר כל אחד).

שלב 3: אימון מודל תמצות מסמכים. המחברים טוענים שב"כ המסמכים שנבחרים מכילים לא מעט מידע לא רלוונטי לשאלה שמקשה על מודל שפה לחתת תשובה טובה וגם מעלה עליות (צריכים להכניס הרבה טוקנים ל-LLM). בגין להתמודד עם הקושי הזה המחברים מציעים לאמן מודל שבהינתן שאלה מפיך מהמסמכים שנבחרו את המידע הרלוונטי לשאלה. זה נעשה ב 2 שלבים: בשלב הראשון עבור דאטאסתט של שאלות והמסמכים הרלוונטיים מתשאלים מודל שפה חזק (gpt4) לתמצת את המסמכים האלו (עבור שאלה נתונה). על הדאטאסתט הזה (שאלה, מסמכים ותמצית) עושים פיניטיון של מודל שפה לא כבד עם LoRa כמו מבון - כלומר עושים Fine-Tuning Supervised SFT או RLHF עם DPO כמו שמקובל היום 😊. בשביב באמצעות מודל שפה(הם לא מפרטים יותר מדי כאן) בונים דאטאסתט של תשובות נכונות ולא נכונות בהינתן שאלה ותמצית מסמכים. בניית פונקציית תגמול (reward) מتبוצעת בדיקן כמו ב- DPO הסטנדרטי.

אחרי שסימנו לאמן את מודל התמצות, ההיסק (אינפרנס) נעשה בצורה מאוד טبيعית. לוקחים שאלה, מפיקים את המסמכים הרלוונטיים משלב 1, מתמצחים אותם עם המודל שלב 3 ואז מזינים אותו לעוד מודל שפה (המחברים לא מפרטים עליו אבל מצינים שניתן לכיל אותו על דאטסהט כלשהו של שאלות ותשובות). והמודל מספק לנו את התשובה...

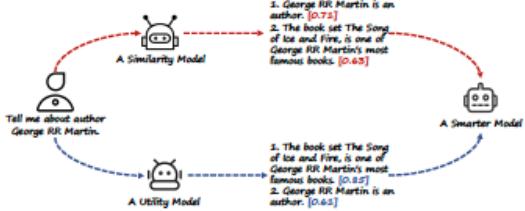


Figure 1: A toy example illustrating the difference between similarity and utility, where the score of similarity model is given by BGE¹. Can we reunite the virtues of both worlds and come up with a better model?

Review 204: Simple linear attention language models balance the recall-throughput tradeoff

<https://arxiv.org/abs/2402.18668>

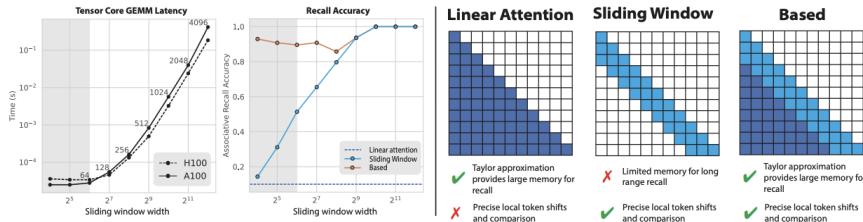


Figure 1: **Based** overview. Combining linear attention with *tiny* sliding window softmax attention (e.g., 64 or 128 tokens in width) enables improved recall accuracy with limited efficiency overhead vs. smaller tile sizes. (Left) Time to execute Cutlass GEMMs (y) vs. sliding window attention size (x), with batch size 512 on tensor cores. (Center) Model recall accuracy (y) vs. sliding window attention size (x). We compare linear attention alone (dark blue), sliding window attention alone (light blue), and their combination (BASED, orange). (Right) Schematic diagram of BASED illustrating how the two components complement each other.

מודלי שפה ענקים של היום מפגינים יכולת מרשיםה של למידת *out-of-context* יכולת לבצע משימות חדשות (שלא אומן עליהם באופן מפורש) בהתבסס על כמה דוגמאות המדגימות (מחישות) את המשימה. מבון דוגמאות אלו מזנות מודל שפה כפרומפט. המאמר שנסקרו היום מדבר על משימת *out-of-context* ספציפית הנדרשת *recall*. המטרה של משימה זו היא להוכיח חוקיות מסוימות בפרומפט ולענות על שאלות בנוגע אליו. למשל אם פרומפט המזון הוא "A 4 B 3 C 6 F 1 G 2". אם לאחר מכן אני מכניסים למודל שפה "B ? F ? C ? A ?". המודל צריך לענות 3 6 כולם המספר בא מיד אחרி כל אותן בפורמט השאלה.

ארQUITktורת הטרנספורmers מתמודדת בהצלחה עם משימות *recall* אף היא מתקשה עם אורך הקשר (context length) מאד ארכיים עקב מנגן self-attention שלהם. ד"א המימושים המודרניים של מנגן זה (כמו FlashAttention2 ו-Paged-Attention) הם בעלי סיבוכיות subquadratic במונחי אורך הסדרה אף עדין גם הם מתקשים "לעכל" אורך הקשר ממש ארכיים.

כדי לחתת מענה לסוגיה זו הוצעו מספר חלופות למנגנון-h-attention כמו h -option f attention (לינארי, שיטות המבוססות על חלון הדז (sliding window) ובנוסף לאחרונה משפחת ארכיטקטורות מובה (סקרטט) אותן בהרחבה לפני כחודשים).

מנגנון h -attention לינארי בגדול מחליף את הסופטמקס של המכפלה הפנימית של וקטור שאלתה (Q) וקטור ערך (K) למכפלה הפנימית של ($Q)f + (K)f$ עברו פונקציה לא לינארית f (יש לא מעט מאמרם המציעים לקחת פונקציות f שונות עבור החלפה זו). אחת הדוגמאות היא לבחור f בהתאם כמה איברים ראשונים של פיתוח טילור של סופטמקס.

פעולה זו מאפשרת להחליף סדר הפעולות בחישוב h -attention ולבצע את החישוב באופן לינארי במונחי אורך הסדרה. דרך אגב החלפה זו היא יכולה כמו reparameterization trick ב-SVMs אבל בכיוון הפוך. היא מאפשרת להיפטר מ"גירירה" של הייצוגים של כל הטוקנים הבודדים באופן מפורש באינפרנס ומאפשרת חישוב בסוגנון RNN. ככלומר כל הזכרון עד תוקן i נדחס לכדי 2 וקטורים (מליץ לקרוא על זה [כאן](#)) וכמובן זה אפשר לחסוך במשאבי חישוב הנדרשים לביצוע אינפרנס באופן משמעותי.

מנגנון h -attention עם החלון ה i הוא פשוט הגבלה גודל ההקשר במנגנון h -attention כאשר יש מגוון גישות ל"AIR לדוחס" את הדטה שלא נכנסת לחלון זה (העברית). בעוד החלון h -attention מחושב באופן רגיל ככלומר הגדלה משמעותית של החלון זה משפרת את הביצועים אבל גם כרוכה בבחירה של יותר חישובים.

מצד אחד ארכיטקטורות המבוססות על h -attention לינארי יודעות להסתדר לא רע עם אורכי הקשר ארוכים מאוד במשימות מסוימות אבל מתקשות לספק ביצועים גבוהים לשאלות בסגנון recall. מצד שני ארכיטקטורות המשמשות החלון h -attention זו מסתדרות יפה עם משימות recall בתחום החלון הזה אולם כדי להביא ביצועים גבוהים עם הקשר ארוך צריך להגדיל את גודל החלון ש כאמור כרוך בהקצאה של יותר משבבים וא/or גם ב-latencies גבוהים יותר באינפרנס.

אוקי דיברנו הרבה על הרקע למאמר אז הגיע הזמן לדבר על מאמר עצמו. קודם כל החוקרים מוכיחים באופן תיאורטי (את הקטע הזה hicci אהבתני [כאן](#)) כי ככל שאורך הקלט לשימוש recall "המודול צריך לזכור" (N O "מידע" כאשר N הוא "אורך" של פרומפט h -recall (זה גם נבדק אמפירית). ככלומר זה תקף לכל ארכיטקטורה והשאלה היחידה איך כל מודל (למשל טרנספורמר לינארי, mamba, s3, hyena ועוד) בונים ומנהלים את הזיכרון הזה ואיך הוא משפייע על ביצועי אינפרנס.

לגביה החידוש שהמאמר מציע: החוקרים שילבו את ה"טוב" שיש במנגנון h -attention הלינארי ובגישה של החלון ה i והציגו מנגנון h -option f חדש הנקרא Based. הם לקחו מנגנון h -attention הלינארי החסכו והיעיל מבחינה ניהול הזיכרון והוסיפו לו חלון i קצר יחסית הממשש מנגנון h -option f הקשור רגיל לטרנספורמים. זה עבד להם לא רע בכלל במשימות recall שונות המצריכות חלון קשר גדול. בנוסף גם הציעו מספר שכליים לשיטה זו המאפשרים להריץ אותה בזיכרון מאד עיליה על GPUs (למשל בחירת גודל החלון כדי שהיא ניתן לבצע את החישובים עבור על ידי שימוש רק הזיכרון המהיר של GPU).

בסוף הכל אומר די נחמד ...

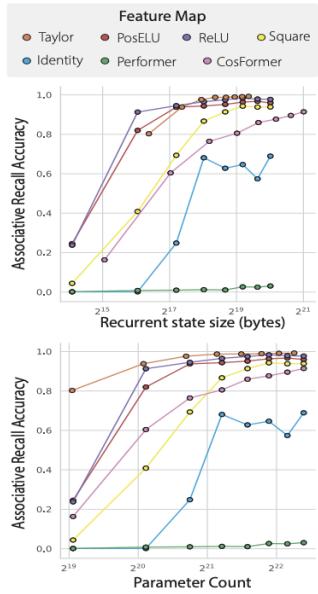


Figure 3: **Linear attention feature maps on AR.** x : state size (bytes) during generation or param. count; y : MQAR accuracy. This setting is harder than fig. 2 (256 key-value pairs).

Review 205: PanGu-π: Enhancing Language Model Architectures via Nonlinearity Compensation

<https://arxiv.org/abs/2312.17276>

היום סוקרים מאמר המציג שדרוג לארכיטקטורת הטרנספורמר. כמו שאתם בטח יודעים בлок של טרנספורמר מרכיב משני החלקים העיקריים (פרט לשכבות נרמול):

- מנגנון תשומת לב עצמי בעל ראשים רבים (multi-head self-attention or MSA)
- שכבת (fully connected) MLP המורכבת משכבה לינארית עם פונקציה אקטיבציה לא לינארית ולאחר מכן שכבה לינארית נוספת (ללא פונקציית אקטיבציה)

כמו שאתם זוכרים מטרת בлок הטרנספורמר היא להפיק "יצוגים תלויי" הקשר של טוקני הקלט. ככלمر כל "יצוג" של כל טוקן לוקח בחשבון את הטוקנים בתורו ההקשר. המחברים מנהיחסים את התכונות של "יצוגי טוקנים תלויי" הקשר הנוצרים על ידי הטרנספורмерים עלי ידי השוואתם עם "יצוגי הטוקנים המזונים לבlok הראשון של הטרנספורמר (כלומר "יצוגי הטוקנים ממטריצת embeddings של מודל שפה)". השיפורים המוצעים במאמר באים למנוע מצב שבו "יצוגי תלויי" ההקשר של טוקנים יהיו דומים מאוד אחד לשני.

תופעה דומה למתחזרת בפסקה הקדמת נקראת over-smoothing ברשותות נירוניים גרפיות (GNN). זה קורה שיש מספר גביה מדי של שכבות MSA שמוביל ל"יצוגים דומים למדי" של הקדקוד העולאים לגורום ל"קritisה" של הייצוגים לתת-מחרב קטן של מרחב הקלט. ד"א מטריצת משקל ה-attention בטרנספורмерים ניתנת לראות בתור מטריצה שכניות מנורמלת של גרף שלהם.

אבל איך נמדד את מידת שונות (diversity) בין ייצוגי הטוקנים? המאמר מגדיר את שונה של מטריצה M (=קבוצה של וקטורים) בתור מינימום נורמת פרובניאס של ההפרש של $A - M$ מעל כל המטריצות A בעלות רnk 1 (כל וקטורים במטריצה תלויים לינארית).

המחברים מראים כי עבור מודל המורכב מ A בלוקי MSA מוערמים (stacked) בלבד (לא MLP) השוני של ייצוג הפלט ניתן לחסום על מכפלה של הערכים הסיגולריים (הכללה של ערכים עצמאיים למטריצות לא ריבועיות) המקיים מליים של מטריצות משקלים השונות במנגן MSA ובשני של ייצוג הקלט (מטריצת אמבדינג של מודל השפה). ללא שכבות MLP ייצוגים אלו נוטים להתנוון ולהפוך להיות תלויים לינארית ככל מספר הבלוקים גדל. זו הסיבה להימצאות של MLP בטרנספורמרים.

בנוסף עבור המודל המורכבים מבלוק MLP מוערמים המאמר מוכיח כי השוני של ייצוג הפלט הינו מכפלה של שני ייצוג הקלט, הערכים הסיגולריים המקיים מליים של מטריצות המשקלים וקבעי ליישן של פונקציות האקטיבציה של MLP.

במטרה לשפר את תכונות ייצוגי הטוקנים בפלט של הטרנספורמר המאמר מציע שני שדרוגים, אחד ל MSA והשני ל-MLP. זוכרים בבלוק הטרנספורמר יש לנו חיבור שاري (residual or shortcut according to the paper) - כולם הפלט של MSA מחובר לייצוג הקלט ל-MSA, המחברים מציעים לפתח חיבור "קיצור דרך" נוספים. כל חיבור זהה הוא למעשה שכבה לינארית עם מטריצה נלדמת ופונקציה אקטיבציה לא לינארית. כדי לא להכביר מדי על העומס החישובי המתווסף בעקבות קר(מטריצות המשקלים בחיבור קיצור דרך אלו יכולות להיות 4096×4096 וזה די הרבה עם רצה להשתמש במקרה חיבורים כאלה) משתמשים במטריצות בעלי רnk נמוך. המחברים מוכיחים שהוספה של שכבות בלוקי הטרנספורמרים המקיימים אלו תורם להקטנת הפגיעה בשוני של ייצוגי פלט.

בנוסף המאמר מציע לשדרוג פונקציה אקטיבציה זהה החלק המהותי של מנגן הטרנספורמרים בנוסף ל-MSA. במקומ להשתמש בפונקציה אקטיבציה רגיל (כמו סיגמאoid או ReLU) המאמר מציע לשלב (חיבורית) ח פונקציות אקטיבציה בצורה הבאה:

$$\sum_{i=1}^n \sigma_i(a_i x + b_i),$$

כאשר a ו- b הם פרמטרים נלמדים. כמובן שיש הוכחה שהחלפה זו תורמת להגדלת השוני בין ייצוגי הפלט.

בנוסף השיפוצים המוצעים נבדקו על מספר נתונים מאקרים והראה ביצועים לא רעים.

Review 206: SSAMBA: SELF-SUPERVISED AUDIO REPRESENTATION LEARNING WITH MAMBA STATE SPACE MODEL

<https://arxiv.org/abs/2405.11831>

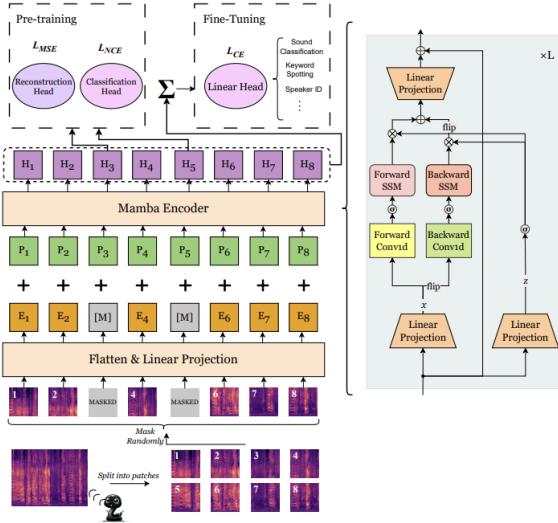


Fig. 1: A top-down view of Self-Supervised Audio Mamba

המאמר הזה משלב תשומת לבנו כי שמו דומה לממבה, ארכיטקטורה מעניינת שפרצה לתודעינו לפני חצי שנה וכבר יצאו עשרות מאמרם המשלבים אותה עבור מגוון דומיינים ומגוון משימות. והפעם התחום הוא אודיו והמחברים משתמשים בארכיטקטורת ממבה למטרת בניית "ցוג חזק" של אות אודיו.

השאלה הרשונה שצרכן לשאול כאן - מה הוא "ցוג חזק" של>Data. בהקשר זה באופן די טבעי "ցוג חזק" של>Data מכוון מזקודה את התכונות החשובות שיש>Data כולם דוחס את המידע המהוות שיש>Data בוצרה יעליה. "Ցוג" זה נבנה על ידי מודל (mboss ממבה כאמור) ויכול לשמש אותנו לאימון של משימות נוספות על אותות אודיו. ככלומר במקומם לאמן מודל למשימה מסוימת על>Data עצמו נאמן אותו על היצוג הלטנטי של>Data (אמבדיניג). דרך אגב התחום בלימידה מכונה העוסק בבניה של "ցוגים" אלו נקרא למידת היצוג או *representation learning*.

כמו שאתם בטח זוכרים ממבה אמרו לקבל קילוט אמבדיניגים של טוקנים. בשפה טבעית כל טוקן הוא תת-מילה או מילה מוגדרים על ידי המילון, עברו תמונה הטוקנים הם פאצ'ים של תמונה (בסדר מסוים) אבל מה אנו עושים עם אות האודיו? האמת שהוא די סטנדרטי - מחלקים את האות שלנו למקטעים זרים שכלי קטע הוא כמה שנויות. לאחר מכן מעבירים כל מקטע זהה דרך התמרת פורייה ולאחר מכן דרך טרנספורמציה מל (Mel transform). בגדול טרנספורמציה מל מדגישה את התדרים שהאזור האנושית מסוגל לשמע. לאחר מכן מעביר את התוצאה של מל דרך שכבה ליניארית ומוסיפים קידוד מיקומי (positional encoding) המזקודה מיקומו של כל טוקן אודיו בסדרה.

לאחר מכן מעבירים את התוצאה דרך שכבת ממבה (די סטנדרטית - ניתן למצוא את תיאורה בהרבה מקומות כולל בסקרים (لينك) הרבות בנושא זה). בדומה למודל ממבה לראייה ממוחשבת (שם המצב אפילו יותר מורכב כי הפאצ'ים של תמונה הם דו-מימדיים) כאן מכינים את "ցוגי הטוקנים" לממבה בשני "סדרים": מהתחלה עד הסוף (forward) ומהסוף להתחלה (backward) ומשלבים אותם כדי לבנות את הפלט.

מה שיצא אחרי כמה שכבות של ממבה הוא למעשה ייצוג תלוי הקשור (contextualized) של הטוקן ואמרור ניתן לנצל אותו לאימון מודלים למגוון משימות ייעודיות.

אבל איך מאמנים את המודל המפיך את הייצוג הזה. בצורה ד' סטנדרטית האמת. ממסכים חלק מהטוקנים (כמו באימון של מודלי שפה) ואז בונים לוס המרכיב משני חלקים:

1. הלוֹס הניגודי (contrastive loss): כאן המטרה לקרב את הייצוג של הטוקן הממוסך לייצוגו (מהאיטרציה הקודמת של אימון) ובאותו הזמן להרחק אותו מהייצוגים של הטוקנים האחרים. ניתן להשיג את היעד זהה עם פונקציית לוס, לראשונה הוצגה במאמר InfoNCE (لينك) לפני 8 שנים בעבר.
2. כאן מנסים לקרב את ייצוג הטוקנים הממוסכים עם ייצוגו (מהאיטרציה אימון הקודמת). המרחק בין חיזוי הייצוג והייצוג עצמו מוגדר כ L_2 כלומר אוקלידי.

Algorithm 1 Bidirectional Mamba Block Processing

Input: Audio embedding sequence: E_1, \dots, E_M

Output: Embeddings sequence: H_1, \dots, H_M

```

1: for  $i = 1$  to  $M$  do
2:   if initial layer then
3:      $E'_i \leftarrow E_i + P_i$   $\triangleright$  Add positional encoding to the initial
   layer input
4:   else
5:      $E'_i \leftarrow H_i$   $\triangleright$  For subsequent layers, use the output of the
   previous layer's corresponding patch
6:   end if
7:    $x \leftarrow \text{Linear}_x(E'_i)$ 
8:    $z \leftarrow \text{Linear}_z(E'_i)$ 
9:   for  $o \in \{\text{forward, backward}\}$  do
10:     $x'_o \leftarrow \text{SiLU}(\text{Conv1D}_o(x))$ 
11:     $B_o \leftarrow \text{Linear}_{B_o}(x'_o)$ 
12:     $C_o \leftarrow \text{Linear}_{C_o}(x'_o)$ 
13:     $\Delta_o \leftarrow \log(1 + \exp(\text{Linear}_{\Delta_o}(x'_o)))$ 
14:     $A_o \leftarrow \Delta_o \times \text{Parameter}_{A_o}$ 
15:     $B_o \leftarrow \Delta_o \times B_o$ 
16:     $y_o \leftarrow \text{SSM}(A_o, B_o, C_o)(x'_o)$ 
17:   end for
18:    $y'_{\text{forward}} \leftarrow y_{\text{forward}} \odot \text{SiLU}(z)$ 
19:    $y'_{\text{backward}} \leftarrow y_{\text{backward}} \odot \text{SiLU}(z)$ 
20:    $H_i \leftarrow \text{Linear}_T(y'_{\text{forward}} + y'_{\text{backward}}) + E'_i$ 
21: end for
22: return  $H_1, \dots, H_M$ 

```

Review 206, Short: Training LLMs over Neurally Compressed Text

<https://arxiv.org/pdf/2404.03626.pdf>

נתקלתי במאמר החמוד הזה של DeepMind and Anthropic

מה הוא בעצם מציע? לאמן מודל שפה לא על טקסט כמו שאנחנו רגילים היום אלא על טקסט מקומפרס. זה מגניב כי מודלי שפה ידועים יכולים לדחוס טקסט לייצוגים דחוסים אבל זה סיפור טיפה שונה.

از מה בעצם נתונים לנו אימון של M על טקסטים דחוסים. קודם כל אימון מהר יותר, או רוך הקשר ארוך יותר ויש עוד כמה. אז מה הבעה? זה קצת עדין - הרי אם אנו דוחסים דатаה עם אלגוריתם חזק התוצאה תהיה רעש רנדומלי (אחרת המודל ילמד וינצל את זה).

אז מה המאמר בעצם עשה? הוא לוקח מודל שפה M שאומן על סדרות ביטים שמייצגות את הטקסט ודוחס את הפלט שלו. כמובן M גם דוחס את הדטה (הרי זה מודל שפה) אבל לטענת המחברים בצורה רחוקה מושלמת. אז הם לוקחים שיטת דחיסת קלאסית הנקראת AC (arithmetic coding) ואומר דוחסו את הפלט של M . הם גם יוצרים טוקנים חדשים אבל הפעם כל טוקן מיוצג על ידי צ'אנק של ביטים (באורך קבוע) הדוחס את של ביטי הפלט. כאן AC לוקחים את ההסתברויות של M מוציא לול טוקן ודוחס אותם. לאחר הפיכתם של סדרות אלו לטוקנים "הדחוסים" מאמנים מודל שפה איתם בצורה הרגילה.

מעניין שנייה ביצועים נעשה על ידי השוואות של perplexity המנורמל עם מקדם דחיסת דטה (יודעים למה?). בסך הכל מאמר חמוד.