Do we need Attention?

Presented by Sasha Rush

This talk is a survey of work done by:

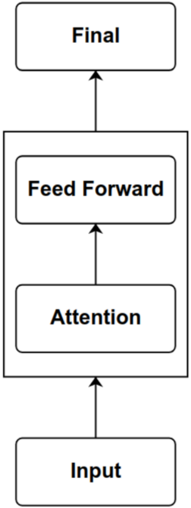
Albert Gu, Ankit Gupta, Tri Dao, Dan Fu, Shuangfei Zhai, Antono Orvieto, Michael Poli, Chris Re, Yuhon Li, Tianle Cai, Harsh Mehta, Jimmy Smith, Scott Linderman, Xuezhe Ma, Chunting Zhou, Xiang Kong, Bo Peng, Eric Alcaide, Anthony Quentin, Andrew Warrington, Yi Zhang, Stefano Massaroli, and many others

Preface: Transformers and Attention

Transformers for Sequence Modeling

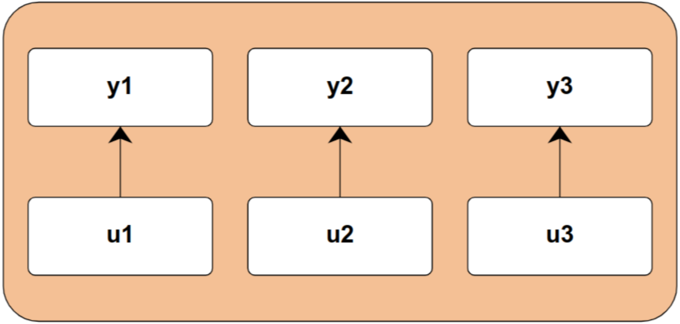
Repeated components

- Feed Forward
- Attention



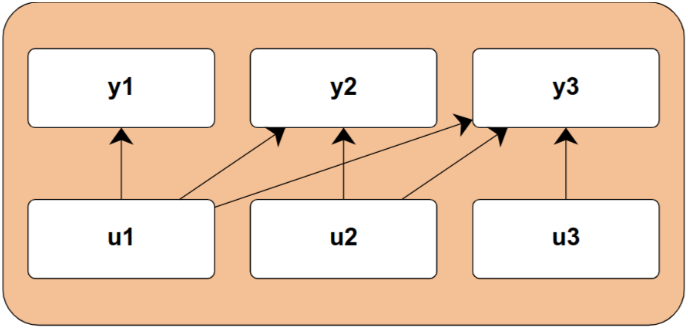
Feed Forward

- Acts on each position independently.



Attention

- Fully connected interactions.



Task: Language Generation

Predict the next word.

Final: The dog walked to the park

Input: The dog walked to the ?

Task: Long Range Arena (ListOps)

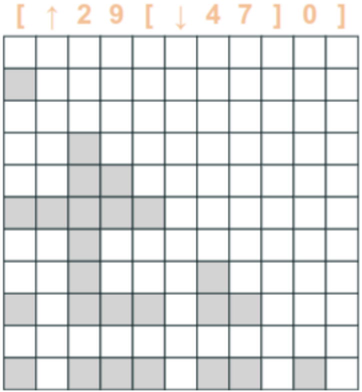
Calculate the equation (↑=max ↓=min)

Final: [↑ 2 9 [↓ 4 7] 0] 9

Input: [↑ 2 9 [↓ 4 7] 0] ?

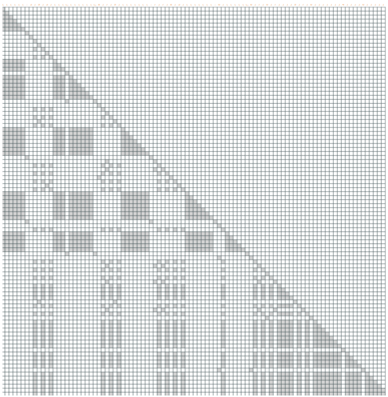
Attention Matrix

All quadratic interactions possible.



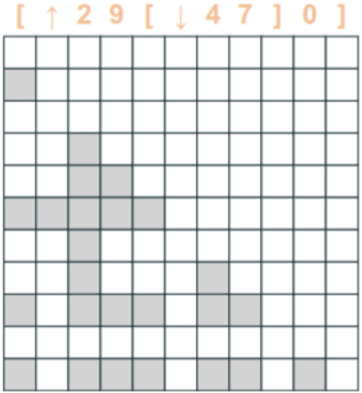
Attention for Realistic Examples

Listops goes to 2,000 steps. This is 100.



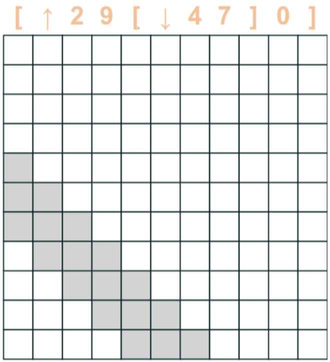
The Challenge

Do we need Attention?



Do we need Attention?

Or can we use something simpler...



Proposition - One year ago

On January 1, 2027, an Attention-based model will be state-of-the-art in natural language processing.

Proposition - One year ago

On January 1, 2027, an [*Attention-based*]() *model will be state-of-the-art in natural language processing.*

Even President Biden has tried ChatGPT

Grace Mayer and Aaron McDade May 4, 2023, 12:43 PM EDT

Algorithmic Goal

GPT models are growing, but still limited by context length.

Algorithmic Goal

GPT models are growing, but still limited by context length.

- Training Speed - Cost is quadratic in length
- Generation Speed - Attention requires full lookback

Survey: Progress on Attention Alternatives

Recent research has made significant progress.

S4 [Gu et al., 2022a]
DSS [Gupta, 2022]
GSS [Mehta et al., 2022]
S4D [Gu et al., 2022b]
H3 [Dao et al., 2022]
S5 [Smith et al., 2022]
BiGS [Wang et al., 2022]

QRNN [Bradbury et al., 2016]
Mega [Ma et al., 2022]
SGConv [Li et al., 2022]
Hyena [Poli et al., 2023]
LRU [Orvieto et al., 2023]
RWKV [Peng et al., 2023]
MultiRes [Shi et al., 2023]

Survey: Progress on Attention Alternatives

Recent research has made significant progress.

S4 [Gu et al., 2022a]
DSS [Gupta, 2022]
GSS [Mehta et al., 2022]
S4D [Gu et al., 2022b]
H3 [Dao et al., 2022]
S5 [Smith et al., 2022]
BiGS [Wang et al., 2022]

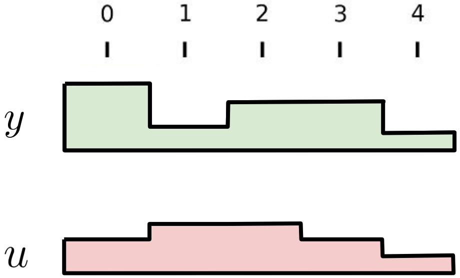
QRNN [Bradbury et al., 2016]
Mega [Ma et al., 2022]
SGConv [Li et al., 2022]
Hyena [Poli et al., 2023]
LRU [Orvieto et al., 2023]
RWKV [Peng et al., 2023]
MultiRes [Shi et al., 2023]

Note: Just one research direction.

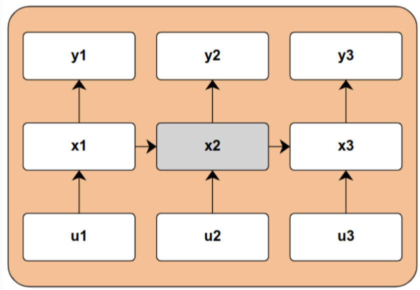
An RNN Revival

Discrete Time Sequence

From scalar sequence $u_1, \ldots, u_L$ to $y_1, \ldots, y_L$.

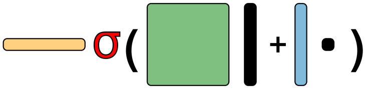


Review: RNN for Language Generation

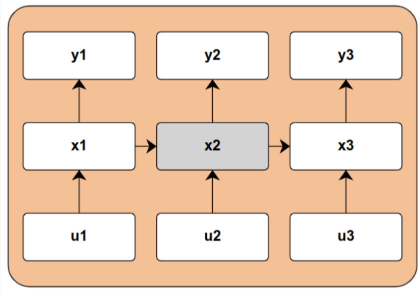


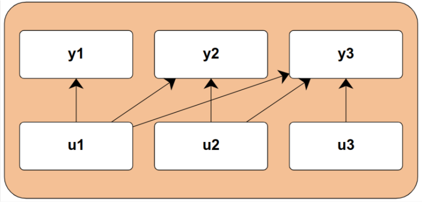
$$x_k = \sigma(\overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k)$$

$$y_k = \overline{C} x_k$$



Review: Elman RNNs

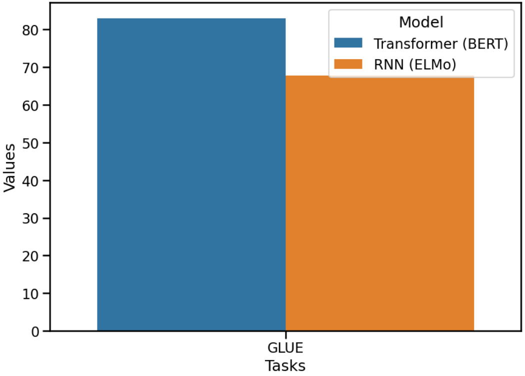




- Training Speed: Slow (Serial bottleneck)
- Generation Speed: Fast (constant-time per step)

Didn't we try this RNN thing?

The last major RNN model in NLP - ELMo

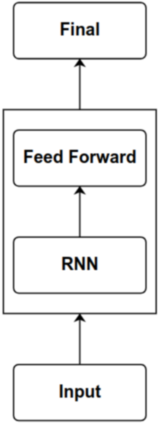


[Peters et al., 2018, Devlin et al., 2018]

RNN Revival: Two Differences

1. Efficient Models

2. Effective Long-Range Parameterizations



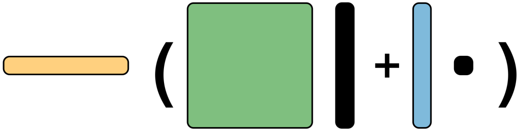
Component 1: Linear RNN

$$x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$
$$y_k = \overline{\boldsymbol{C}} x_k$$

Component 1: Linear RNN

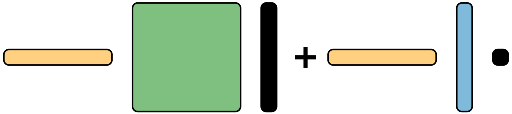
$$x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

$$y_k = \overline{\boldsymbol{C}} x_k$$



Component 1: Linear RNN

$$x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$
$$y_k = \overline{\boldsymbol{C}} x_k$$



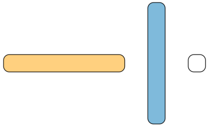
Expansion Of Terms

$$y_k = \overline{\boldsymbol{C}} x_k \quad x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

Expansion Of Terms

$$y_k = \overline{\boldsymbol{C}} x_k \qquad x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

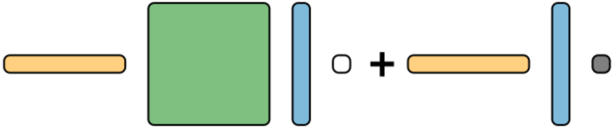
$$u_1 \rightarrow y_1$$



Expansion Of Terms

$$y_k = \overline{\boldsymbol{C}} x_k \qquad x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

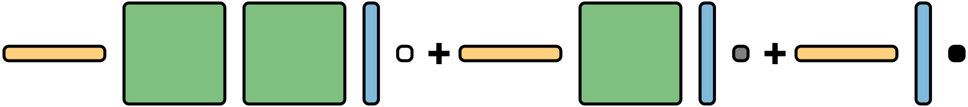
$$u_1, u_2 \rightarrow y_2$$



Expansion Of Terms

$$y_k = \overline{\boldsymbol{C}} x_k \qquad x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

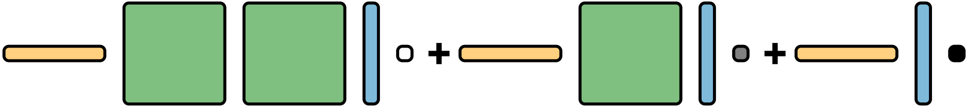
$$u_1, u_2, u_3 \to y_3$$



Expansion Of Terms

$$y_k = \overline{\boldsymbol{C}} x_k \qquad x_k = \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k$$

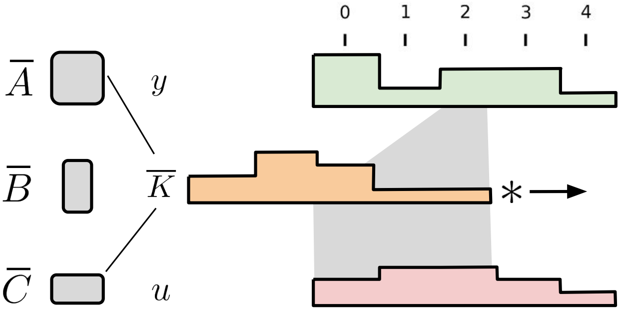
$$u_1, u_2, u_3 \to y_3$$



$$\overline{K} = (\overline{\boldsymbol{C}}\overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}, \dots, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}})$$

Convolutional Form

$$\overline{K} = (\overline{\boldsymbol{C}\boldsymbol{B}}, \overline{\boldsymbol{C}\boldsymbol{A}\boldsymbol{B}}, \dots, \overline{\boldsymbol{C}\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}})$$



Method: FFT

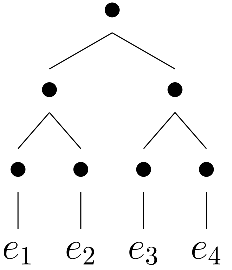
Compute convolution in Fourier space,

$$y_{1:L} = \overline{\boldsymbol{K}} * u_{1:L}$$

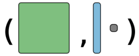
- $O(L \log L)$ for padded FFT of K and u, mult, then iFFT

Alternative: Associative Scan (S5)

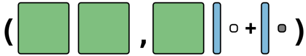
Associative $e_1 \bullet \dots \bullet e_L$



$$e_k = (\boldsymbol{E}_k, \boldsymbol{e}_k) = (\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}} u_k)$$



$$e_i \bullet e_j = (\boldsymbol{E}_i \boldsymbol{E}_j, \boldsymbol{E}_j \boldsymbol{e}_i + \boldsymbol{e}_j)$$



[Blelloch and Reif, 1990, Martin and Cundy, 2018, Smith et al., 2022]

Linear RNN Computational Profile

$$
\begin{aligned}
x_k &= \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k \\
y_k &= \overline{\boldsymbol{C}} x_k
\end{aligned}
$$

- Training Speed: ~~Weak~~ Strong (Parallelizable convolution)
- Generation Speed: Strong (constant-time per step)

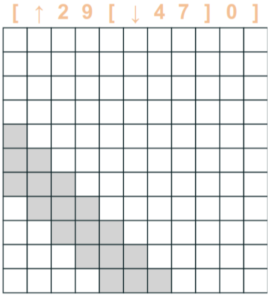
Linear RNN Computational Profile

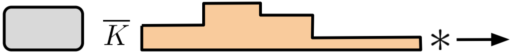
$$
\begin{aligned}
x_k &= \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k \\
y_k &= \overline{\boldsymbol{C}} x_k
\end{aligned}
$$

- Training Speed: ~~Weak~~ Strong (Parallelizable convolution)
- Generation Speed: Strong (constant-time per step)
- Accuracy: Extremely Poor... Barely learns.

Interactions

Routing here must be static and regular (conv).





Component 2: Model Parameterization

Long convolution behavior is highly dependent on $\overline{\boldsymbol{A}}$

$$\overline{K} = (\overline{\boldsymbol{C}}\overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}\overline{\boldsymbol{B}}, \dots, \overline{\boldsymbol{C}}\overline{\boldsymbol{A}}^{L-1}\overline{\boldsymbol{B}})$$

Initialization of $\overline{\boldsymbol{A}}$ is critical: stable and informative.

Mathematical Model: State Space Model (SSM)

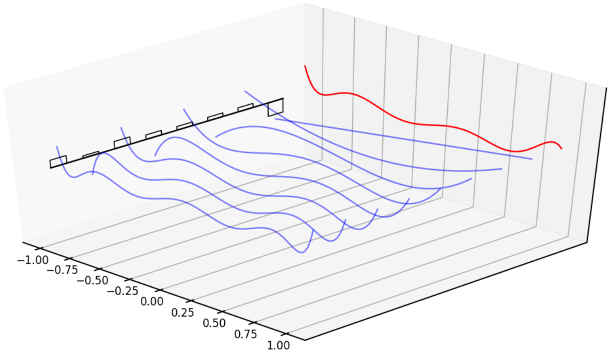
A SSM is a continuous-time, differential equation.

$$
\begin{aligned}
x'(t) &= \boldsymbol{A}x(t) + \boldsymbol{B}u(t) \\
y(t) &= \boldsymbol{C}x(t).
\end{aligned}
$$

- Used to explore Linear RNN parameterization.
- Recall x is a vector-valued hidden state.

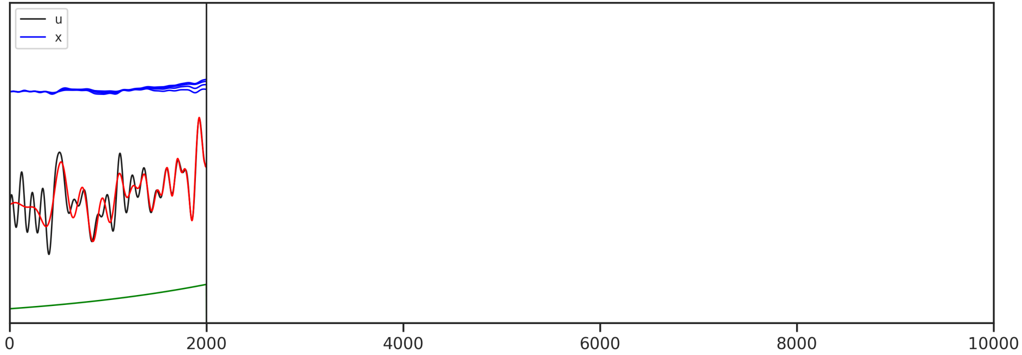
Hidden State Form [Gu et al., 2020]

Summarize history in vector x with Legendre coefficients



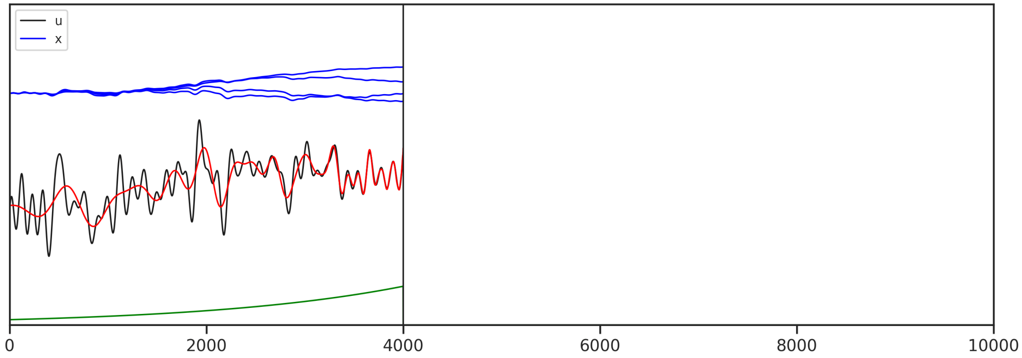
Choice of Parameters [Gu et al., 2020]

Intuition: Hidden state vector x should summarize past u.



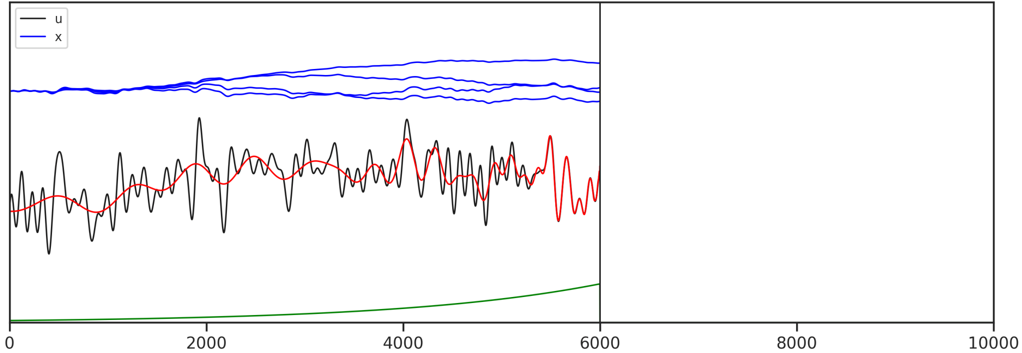
Choice of Parameters [Gu et al., 2020]

Intuition: Hidden state vector x should summarize past u.



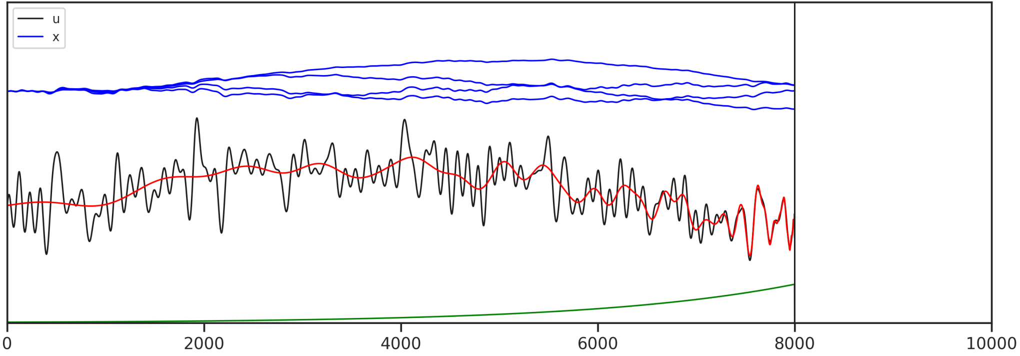
Choice of Parameters [Gu et al., 2020]

Intuition: Hidden state vector x should summarize past u.



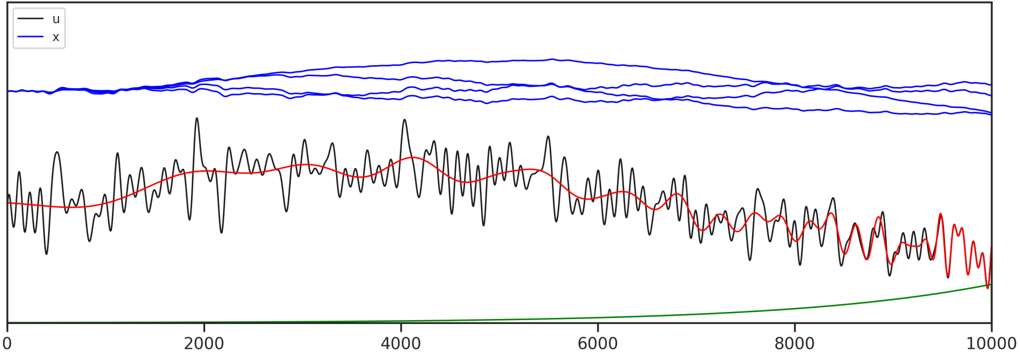
Choice of Parameters [Gu et al., 2020]

Intuition: Hidden state vector x should summarize past u.



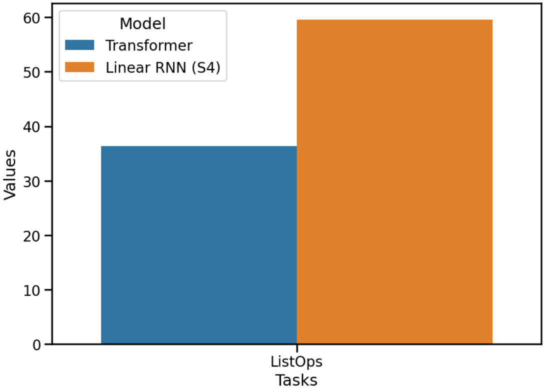
Choice of Parameters [Gu et al., 2020]

Intuition: Hidden state vector x should summarize past u.



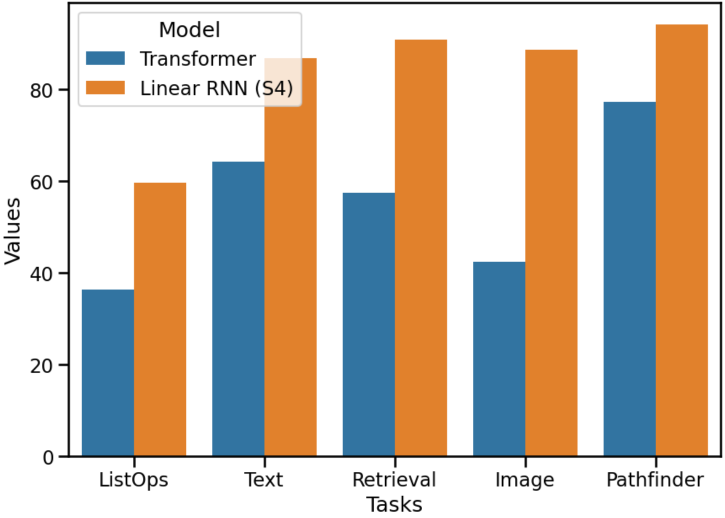
Results: ListOps [Gu et al., 2022a]

Example: [$\uparrow$ 2 9 [$\downarrow$ 4 7] 0] 9



Requires communication over 2,000 steps

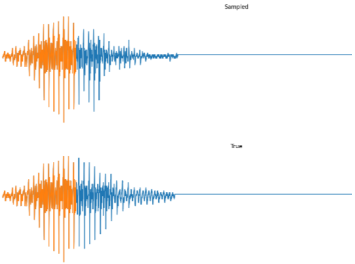
Results: Long-Range Arena [Gu et al., 2022a]



Are we GPT yet?

Applying Linear RNNs

- Speech [Goel et al., 2022]
- Video [Nguyen et al., 2022]
- RL [Lu et al., 2023]
- NLP

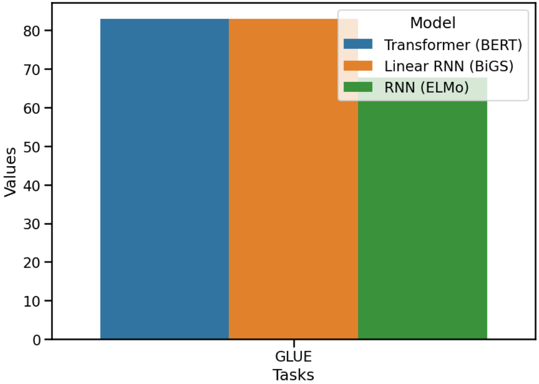


NLP Results

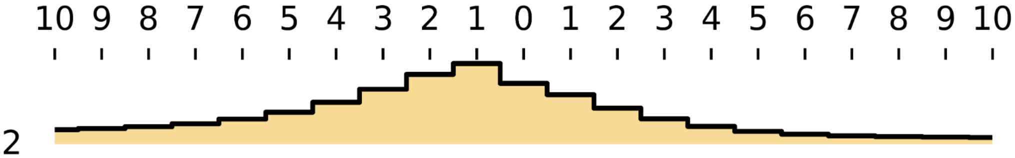
Two types of model

- Bidirectional LM (BERT)
- Unidirectional LM (GPT)

Results: Bidirectional LM [Wang et al., 2022]

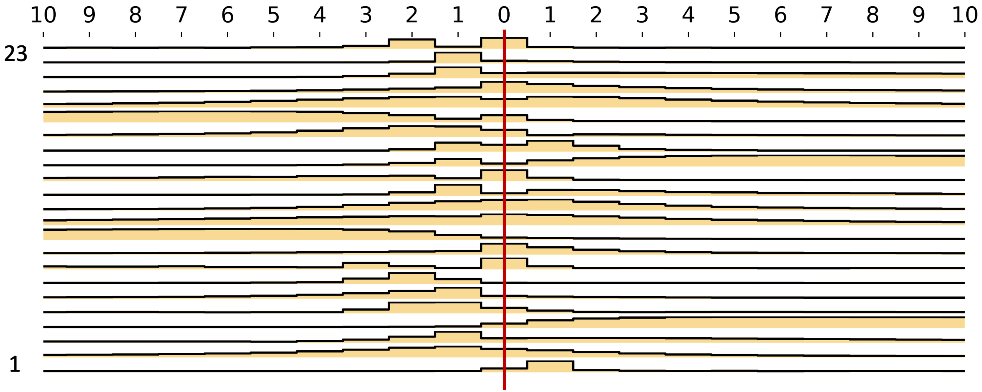


Analysis: Kernel Visualization $\bar{K}$



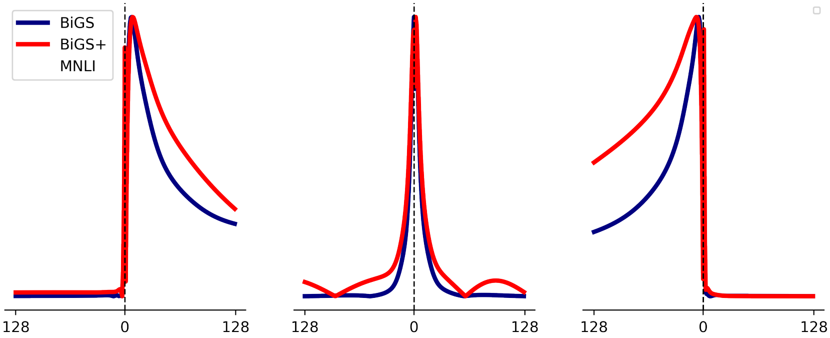
- Replaces Attention Matrix
- Single Kernel per layer

Analysis: All Kernels

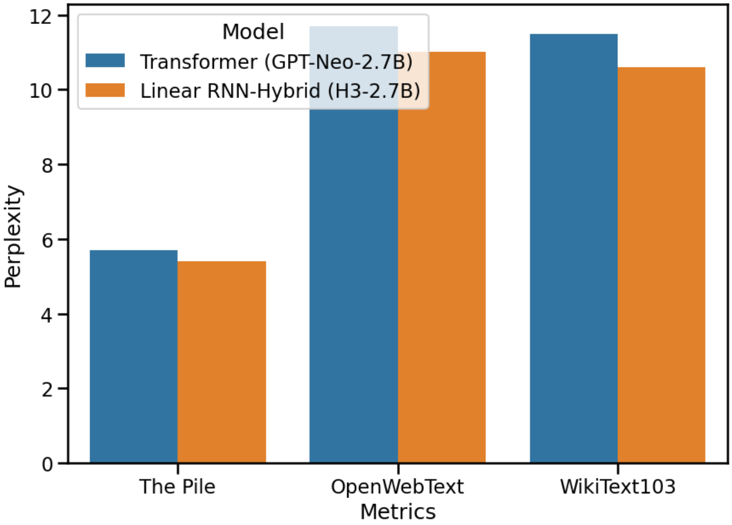


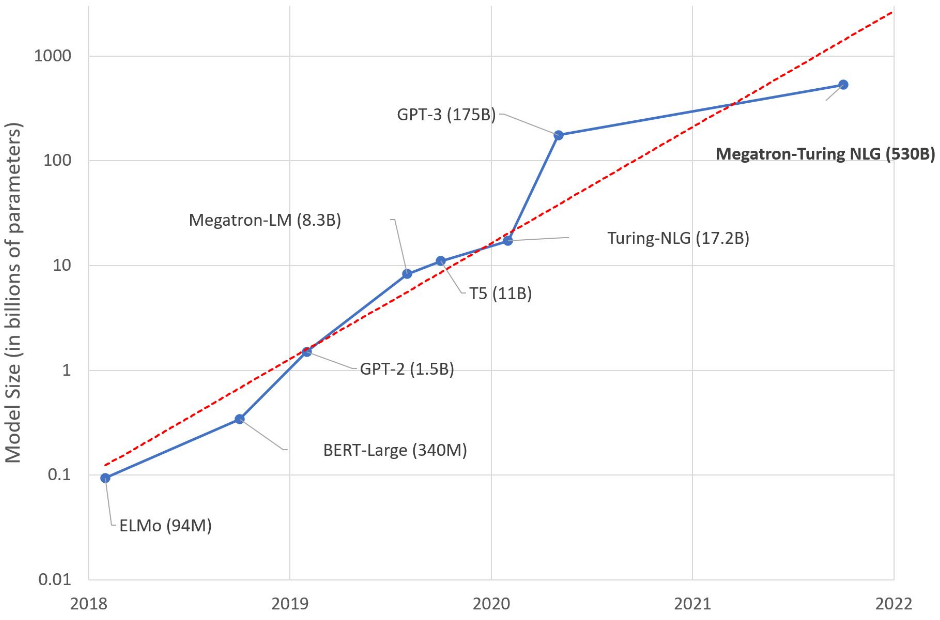
Analysis: Change in Kernels during Finetuning

Task: Long-Range Sentence Matching



Results: Unidirectional LM [Dao et al., 2022] ↓

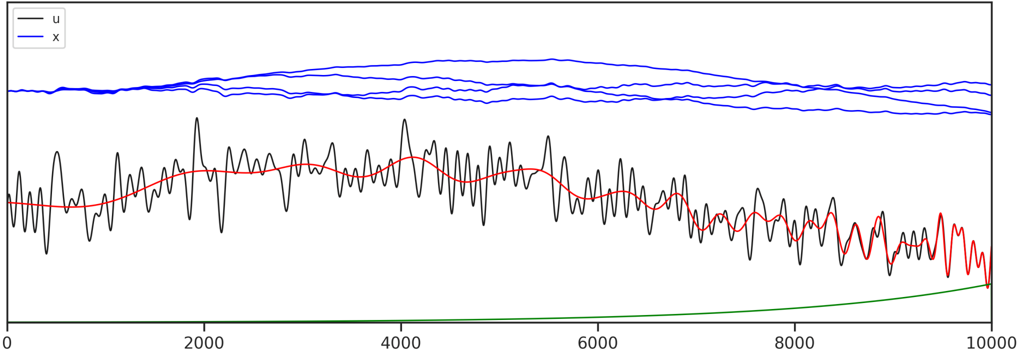




Model Size (in billions of parameters)
1000
100
10
1
0.1
0.01
2018
2019
2020
2021
2022
GPT-3 (175B)
Megatron-Turing NLG (530B)
Megatron-LM (8.3B)
Turing-NLG (17.2B)
T5 (11B)
GPT-2 (1.5B)
BERT-Large (340M)
ELMo (94M)

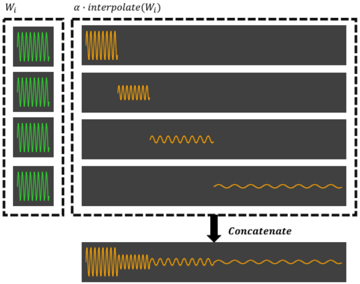
Alternative Parameterizations

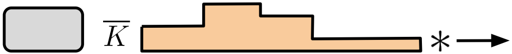
Do we need the SSM?



CNN Param: Decaying Structure [Li et al., 2022]

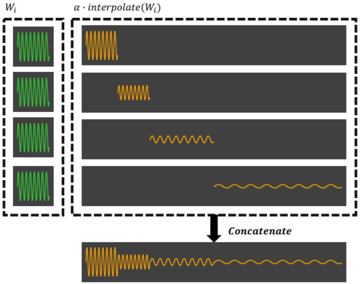
Parameterization should decay $\bar{K}$ over time.

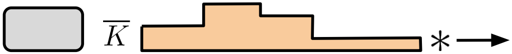




CNN Param: Decaying Structure [Li et al., 2022]

Parameterization should decay $\bar{K}$ over time.



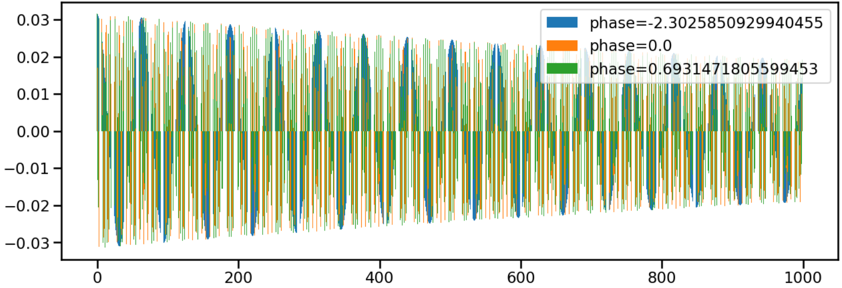


However, non-trivial to convert to finite linear RNN form.

RNN Param: LRU [Orvieto et al., 2023]

Stable diagonal parameterization of Linear RNN

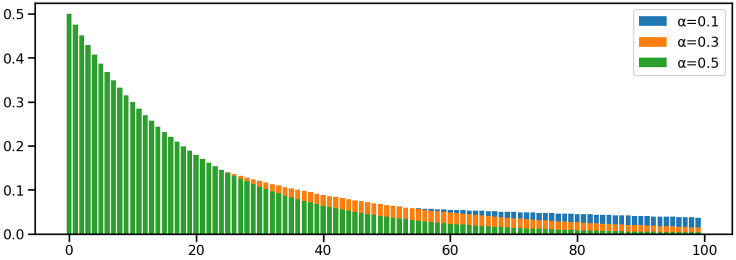
$$\bar{A}_{j,j} = \exp(-\exp(\nu_j) + i\exp(\theta_j))$$
$$\bar{B}_j = (1 - |\bar{A}_{j,j}|^2)^{1/2}$$



RNN Param: MEGA [Ma et al., 2022]

Use a parameterized damped, exponential moving average

$$\bar{A}_{j,j} = 1 - \alpha_j \times \delta_j$$
$$\bar{B}_j = \alpha_j$$



Strong results on NLP tasks

RNN Param: AFT / RWKV [Peng et al., 2023]

Inspired by Attention-approximation [Zhai et al., 2021]

$$R_i, K_i, V_i$$

RNN Param: AFT / RWKV [Peng et al., 2023]

Inspired by Attention-approximation [Zhai et al., 2021]

$$R_i, K_i, V_i$$

Then compute averaged values normalized by keys.

$$R_i \frac{\sum_{i'=1}^{i} \color{green}{\exp(w)}^{i'} \exp(K_{i'}) V_{i'}}{\sum_{i'=1}^{i} \color{green}{\exp(w)}^{i'} \exp(K_{i'})} = R_i \frac{\mathsf{LR}_1(\exp(K_i) V_i)}{\mathsf{LR}_2(\exp(K_i))}$$

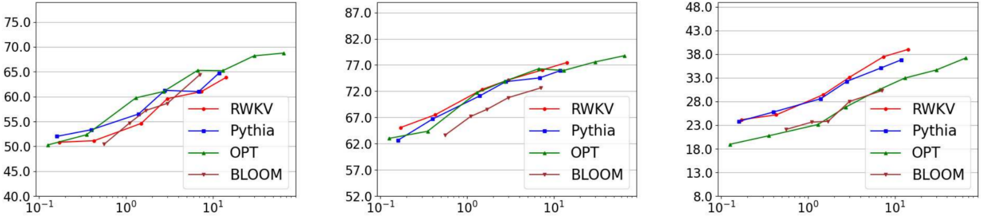
Yields a product of Linear RNNs (Computed directly).

Results: RWKV [Peng et al., 2023]

Largest RNN. Trained up to 14B parameter scale.

Results: RWKV [Peng et al., 2023]

Largest RNN. Trained up to 14B parameter scale.



(a) Winogrande (b) PIQA (c) ARC-Challenge

- Lots of practical interest and community.

Open Question: In-Context Learning

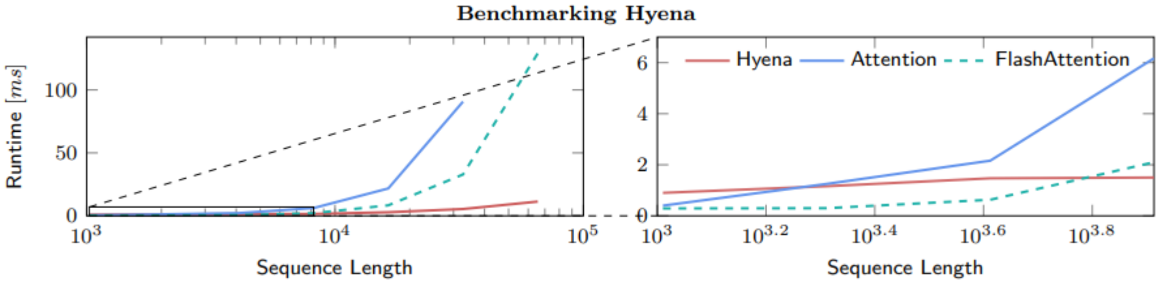
- Results show comparable loss at medium scales.
- Significant interest is in abilities such as in-context learning
- Current understanding relies of Attention mechanisms.

Call to Action: Scaling Linear RNNs

Benefits of Linear RNNs

- Methods for training (CNN) and generation (RNN)
- Potentially more FLOP efficient.
- However not yet used in practice

Current Efficiency with Scale [Poli et al., 2023]



Models become more efficient at long time-scales.

Better Accelerators can Help

Approaches require:

- Support for complex numbers
- Support for FFT (lower precision, TPU)
- Numerical Stability
- Fast Associative Scans

Hard to compete with pure MatMul in Attention.

Is Attention All You Need?

Current Status: Yes

Time Remaining: 1318d 0h 5m 37s

Let's go!

S4 [Gu et al., 2022a]
DSS [Gupta, 2022]
GSS [Mehta et al., 2022]
S4D [Gu et al., 2022b]
H3 [Dao et al., 2022]
S5 [Smith et al., 2022]
BiGS [Wang et al., 2022]
QRNN [Bradbury et al., 2016]
Mega [Ma et al., 2022]
SGConv [Li et al., 2022]
Hyena [Poli et al., 2023]
LRU [Orvieto et al., 2023]
RWKV [Peng et al., 2023]
MultiRes [Shi et al., 2023]

References I

Blelloch, G. E. and Reif, J. H. (1990).
Prefix sums and their applications.
http://shelf2.library.cmu.edu/Tech/23445461.pdf.
Accessed: 2023-5-30.

Bradbury, J., Merity, S., Xiong, C., and Socher, R. (2016).
Quasi-Recurrent neural networks.

Dao, T., Fu, D. Y., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. (2022).
Hungry hungry hippos: Towards language modeling with state space models.
arXiv preprint arXiv:2212.14052.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018).
Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805.

References II

Goel, K., Gu, A., Donahue, C., and Ré, C. (2022).
It's raw! audio generation with state-space models.
arXiv preprint arXiv:2202.09729.

Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. (2020).
Hippo: Recurrent memory with optimal polynomial projections.
Advances in Neural Information Processing Systems, 33:1474–1487.

Gu, A., Gupta, A., Goel, K., and Ré, C. (2022a).
On the parameterization and initialization of diagonal state space models.
arXiv preprint arXiv:2206.11893.

Gu, A., Gupta, A., Goel, K., and Ré, C. (2022b).
On the parameterization and initialization of diagonal state space models.

References III

Gupta, A. (2022).
Diagonal state spaces are as effective as structured state spaces.
arXiv preprint arXiv:2203.14343.

Li, Y., Cai, T., Zhang, Y., Chen, D., and Dey, D. (2022).
What makes convolutional models great on long sequence modeling?

Lu, C., Schroecker, Y., Gu, A., Parisotto, E., Foerster, J., Singh, S., and Behbahani, F. (2023).
Structured state space models for In-Context reinforcement learning.

Ma, X., Zhou, C., Kong, X., He, J., Gui, L., Neubig, G., May, J., and Zettlemoyer, L. (2022).
Mega: moving average equipped gated attention.
arXiv preprint arXiv:2209.10655.

References IV

Martin, E. and Cundy, C. (2018).
Parallelizing linear recurrent neural nets over sequence length.

Mehta, H., Gupta, A., Cutkosky, A., and Neyshabur, B. (2022).
Long range language modeling via gated state spaces.
arXiv preprint arXiv:2206.13947.

Nguyen, E., Goel, K., Gu, A., Downs, G. W., Shah, P., Dao, T., Baccus, S. A., and Ré, C. (2022).
S4ND: Modeling images and videos as multidimensional signals using state spaces.

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. (2023).
Resurrecting recurrent neural networks for long sequences.

References V

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., Gv, K. K., He, X., Hou, H., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau, H., Mantri, K. S. I., Mom, F., Saito, A., Tang, X., Wang, B., Wind, J. S., Wozniak, S., Zhang, R., Zhang, Z., Zhao, Q., Zhou, P., Zhu, J., and Zhu, R.-J. (2023).

RWKV: Reinventing RNNs for the transformer era.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018).

Deep contextualized word representations.

In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

References VI

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. (2023).
Hyena hierarchy: Towards larger convolutional language models.

Shi, J., Wang, K. A., and Fox, E. B. (2023).
Sequence modeling with multiresolution convolutional memory.

Smith, J. T., Warrington, A., and Linderman, S. W. (2022).
Simplified state space layers for sequence modeling.
arXiv preprint arXiv:2208.04933.

Wang, J., Yan, J. N., Gu, A., and Rush, A. M. (2022).
Pretraining without attention.

References VII

Zhai, S., Talbott, W., Srivastava, N., Huang, C., Goh, H., Zhang, R., and Susskind, J. (2021).
An attention free transformer.