状態空間モデルとMamba

NTTドコモビジネス
片島　健博

内容

状態空間モデル（SSM: State Space Model）ベース手法をMambaを中心に解説していく：

1.状態空間モデルの概要

2.S4 [1]

3.Mamba [3]

4.Mamba-2 [4]

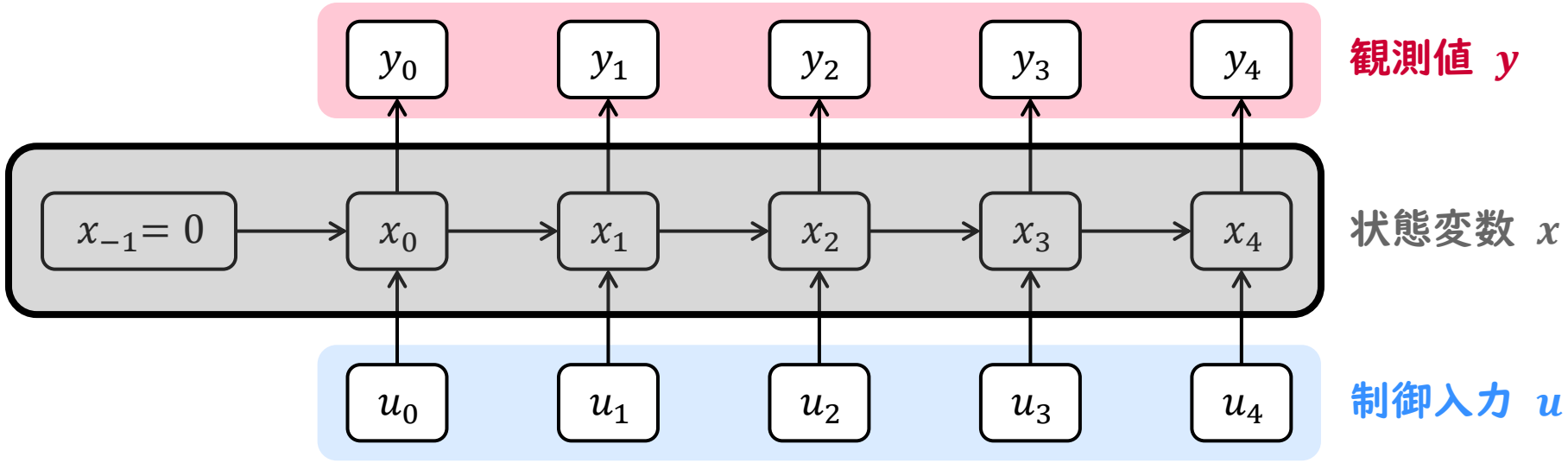
状態空間モデル：概要

状態空間モデル（SSM: State Space Model）は下式により記述されるモデル：

$$\begin{cases} x'(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \xrightarrow{\text{discretize}} \begin{cases} x_k = \bar{A}x_{k-1} + \bar{B}u_k, \\ y_k = Cx_k. \end{cases}$$

制御入力uを入力すると内部状態xが更新され、内部状態xが観測値yを通じて観測されることを表現する。

近年、ニューラルネットワーク内に状態空間モデルによる変換（$u \mapsto y$）を組み込むモデルの研究が進んでいる。



状態空間モデル（離散時間モデル）

状態空間モデル：Attentionとの比較

つなごう。驚きを。幸せを。
docomo Business

状態空間モデル（SSM）は**学習時の計算量がトークン長Nについて線形**である。

そのため、計算量がトークン長Nの2乗オーダーとなるAttentionと比較して、高速に学習を行える。

また、SSMは過去の入力履歴を固定サイズの状態変数に圧縮するため、**推論時のメモリ使用量も少ない**。

	SSM	Attention
計算量（学習時）	N （or $N \log N$）	N^2
メモリ使用量（推論時）	過去のKVを全て保持	固定サイズの状態のみ保持

SSMでは各時刻での出力を計算するために直前の状態を必要とするので本来なら逐次計算が必要となるが、
畳み込みやParallel scan [6]、SSDアルゴリズム [4] を用いた並列化により高速化可能である。（詳細は後述）

状態空間モデル：畳み込みによる計算

つなごう。驚きを。幸せを。
docomo Business

離散時間モデルの出力yは、

$$\begin{cases} x_k = \bar{A}x_{k-1} + \bar{B}u_k, \\ y_k = Cx_k, \end{cases} \longrightarrow \begin{aligned} y_0 &= C\bar{B}u_0, \\ y_1 &= C\bar{A}\bar{B}u_0 + C\bar{B}u_1, \\ y_2 &= C\bar{A}^2\bar{B}u_0 + C\bar{A}\bar{B}u_1 + C\bar{B}u_2, \\ &\vdots \end{aligned} \longrightarrow \begin{cases} \bar{K} = [C\bar{B} \quad C\bar{A}\bar{B} \quad \cdots \quad C\bar{A}^k\bar{B} \quad \cdots], \\ y = u * \bar{K}, \end{cases}$$

のように畳み込み $y = u * \bar{K}$ の形で表現できる。

これにより、最初に$\bar{K}$ を1回だけ計算しておけば効率よく出力 u を計算できる。

S4モデル [1] ではさらにFFTなどを用いてこの畳み込みを高速化する技法が取り入れられている。

S4モデル：論文概要

つなごう。驚きを。幸せを。
NTT docomo Business

"Efficiently modeling long sequences with structured state spaces “ (ICLR2022)

Gu, A., Goel, K., and Ré, C.

- SSMを組み込んだS4モデルの提案
- SSMのパラメータAに特殊な構造を持たせることで、勾配消失・勾配爆発を抑制しつつ計算も高速化（後述）
- FFTなどのテクニックを用いて、畳み込み演算を高速化している

S4モデル：HiPPO [5] の概略

つなごう。驚きを。幸せを。
docomo Business

時刻tにおいて、過去の入力$u(\tau)\quad(0 \le \tau \le t)$を$N$次未満の多項式$g^{(t)}$で近似することを考える。

この時、最適な多項式$g^{(t)}$をある測度$\mu^{(t)}(\tau)$から定まる直交多項式$\{g_n\}_{n=0}^{N-1}$の線形結合で表すとする：

$$g^{(t)}(\tau) = \sum_{n=0}^{N-1} x_n(t) g_n(\tau)\,.$$

時刻tが進むにつれて考慮すべき過去の入力も長くなり、$x_n(t)$も変化していく。

例えば測度$\mu^{(t)}(\tau)$が一様測度のとき、 $x_n(t)$は以下の微分方程式を満たす：

$$\frac{d}{dt}x(t) = -\frac{1}{t}Ax(t) + \frac{1}{t}Bu(t). \quad ※A, Bは定数行列$$

→　上式で現れる定数行列A（HiPPO行列）を状態空間モデルに用いることで、長期記憶が可能になるのでは？

S4モデル：HiPPOからDPLRへ

つなごう。驚きを。幸せを。
docomo Business

[1] のTheorem 1によると、HiPPO行列はユニタリ変換により

$$\Lambda - PQ^T$$

の形で表される。（Λ：対角行列、P, Q：低ラング行列）

→　SSMは入出力の関係を変えずに状態変数をユニタリ変換できるので、HiPPOの代わりに$\Lambda - PQ^T$でもよい！

S4ではSSMの行列Aを$\Lambda - PQ^T$というDPLR（diagonal plus low-rank）の形で扱う。
これにより長期記憶を可能にしながら、FFTなどを用いた計算の高速化を実現している。
（詳細は[1]のAppendix Cを参照されたし）

Mamba：論文概要

つなごう。驚きを。幸せを。
docomo Business

"Mamba: Linear-Time Sequence Modeling with Selective State Spaces“
(COLM2024)

A. Gu and T. Dao

- **SSMのパラメータを入力xに依存させる**ようにし、xに応じて情報の選択的取捨選択を行えるように変更
  - 例：言語処理で、意味の薄い単語が来たら状態をあまり更新しない

- **Parallel scan**や、ハードウェアのメモリを意識したアルゴリズムにすることで高速化を実現

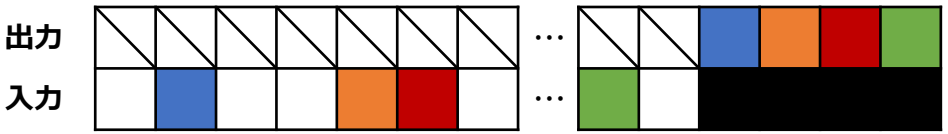
Mamba：時不変モデルの課題

S4モデルで用いられる畳み込み表現

$$\begin{cases} x_k = \bar{A}x_{k-1} + \bar{B}u_k, \\ y_k = Cx_k. \end{cases} \longrightarrow \begin{aligned} y_0 &= C\bar{B}u_0, \\ y_1 &= C\bar{A}\bar{B}u_0 + C\bar{B}u_1, \\ y_2 &= C\bar{A}^2\bar{B}u_0 + C\bar{A}\bar{B}u_1 + C\bar{B}u_2, \\ &\vdots \end{aligned} \longrightarrow \begin{cases} \bar{K} = [C\bar{B} \quad C\bar{A}\bar{B} \quad \cdots \quad C\bar{A}^k\bar{B} \quad \cdots], \\ y = u * \bar{K}, \end{cases}$$

は $\bar{A}, \bar{B}, C$ が時間に依存しないパラメータであることを仮定している。（→　時不変モデルを仮定している）

だが、時不変なモデルでは以下の2つのタスクを解くことができない：



Selective Copying

入力中の重要なトークンを記憶。
ただし、間にランダムな長さのノイズトークンがある。

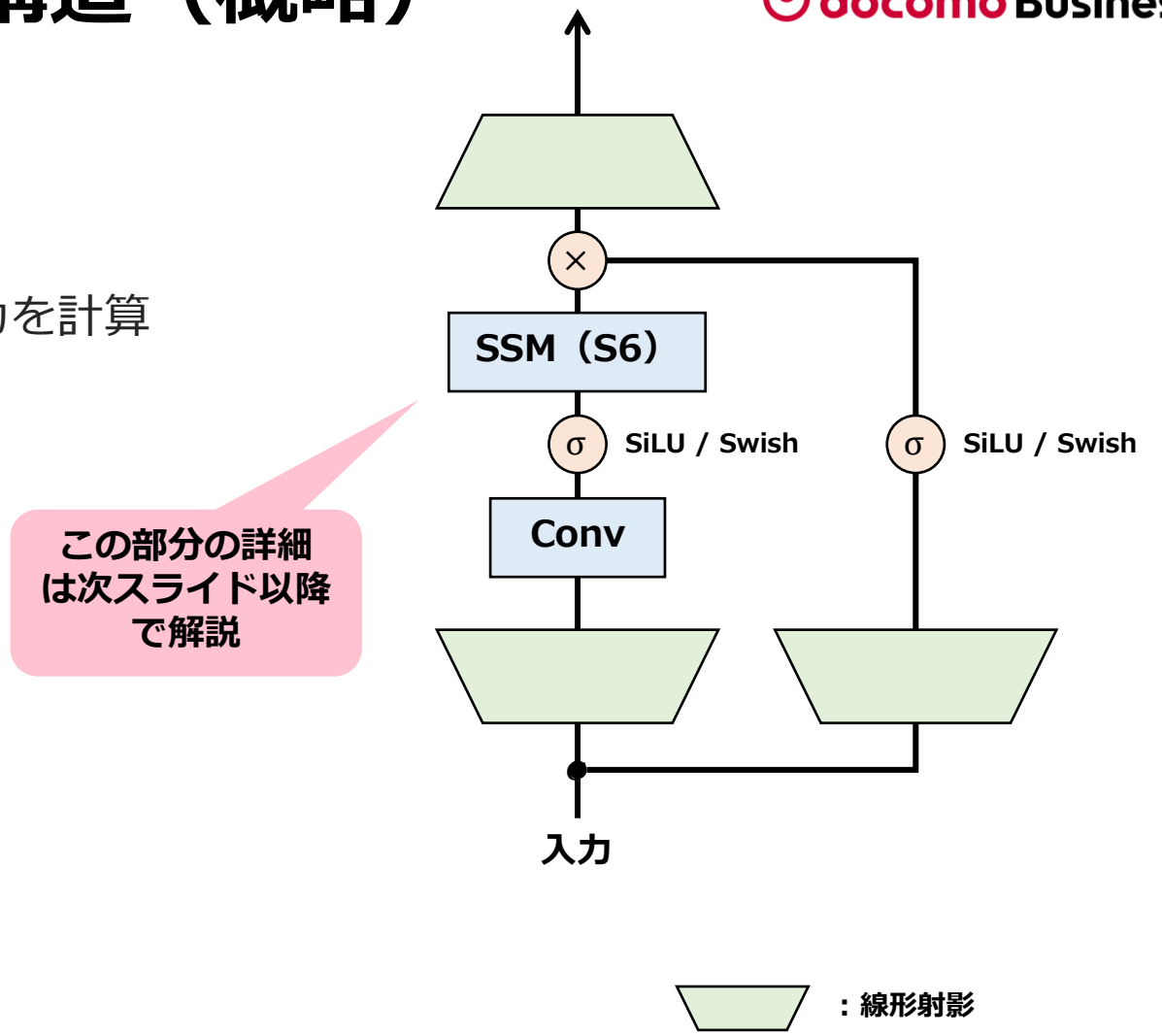
Induction Heads

入力からパターンを見出し、次のトークンを予測。
図では「黒の次は青」というパターンをもとに推論を行っている。

Mamba：Mambaブロックの構造（概略）

Mambaブロックは右図の構造をもつ。

- Conv層で局所的な時間依存性をとらえる
- SSMブロックでは入力を状態空間モデルに通し、出力を計算（次スライド以降解説）
- もう一本のパス（ゲート機構）で情報を取捨選択



Mamba：S6の概要

つなごう。驚きを。幸せを。
docomo Business

Mambaブロック内のSSM（S6）では、状態空間モデル

$$\begin{cases} x'(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases}$$

の**パラメータB, Cおよび離散化幅Δを入力uに依存して変化させる**ことで、情報の選択的な書き込み、読み取りを可能にした。

※Aは対角行列という制約があり、かつ時不変だが、離散化された$\bar{A} = \exp(\Delta_t A)$は$\Delta_t$ の影響で入力依存となる

→ S5 [2] からさらにB, Cも入力 u に依存するようにしたモデルとなっている

これだと畳み込みによる効率の良い計算ができないが、下記の工夫により高速化を実現している：

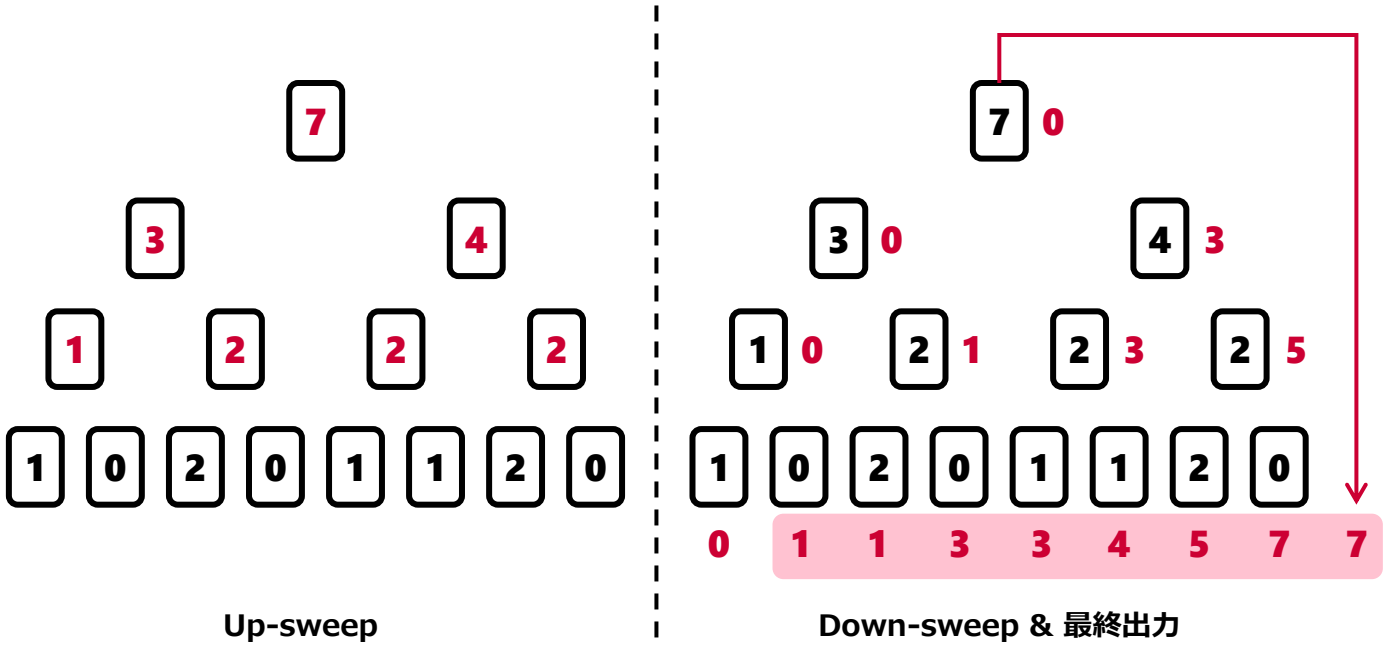
- **Parallel scan**： S5で導入された、畳み込み演算の並列アルゴリズム（次スライドにて解説）
- **Kernel fusion**：SSMの計算時、途中の変数をメモリに保存することなく高速なSRAM上で全て完結させる
- **Recomputation**：順伝播時の中間状態を保存せず、逆伝播の際にSRAM内で再計算する

Mamba : Parallel scan [6]

Parallel scanは、累積和の並列計算を可能にするアルゴリズム。

入力の長さがnのとき、逐次計算なら$\mathcal{O}(n)$ステップかかるところを、$\mathcal{O}(\log n)$ステップで計算可能にする。

通常の足し算以外にも、**結合法則　$a \circ (b \circ c) = (a \circ b) \circ c$　が成り立つ演算 $\circ$ について適用可能**。



Parallel scanアルゴリズム

1. 隣接する要素を足し合わせて二分木を構築(up-sweep)
2. 根を0に置き換え、二分木の根から葉に向かって以下の処理を繰り返す(down-sweep)：
   - 左の子を親の値、
     右の子を親の値＋左の子の値
     でそれぞれ書き換える
3. 末尾に合計値を付加し、1つずらす

Mamba：SSMにおけるParallel scan

つなごう。驚きを。幸せを。
docomo Business

SSMにおいては、各ステップの状態を$(\bar{A}_t, \bar{B}_t u_t)$のペアで管理し、演算$\circ$を

$$(A_1, b_1) \circ (A_2, b_2) = (A_2 A_1, A_2 b_1 + b_2)$$

で定義するとこの演算は結合法則を満たし、

$$(\bar{A}_0, \bar{B}_0 u_0) \circ (\bar{A}_1, \bar{B}_1 u_1) \circ \cdots \circ (\bar{A}_k, \bar{B}_k u_k) = (\bar{A}_k \cdots \bar{A}_1 \bar{A}_0, \quad \bar{A}_k \cdots \bar{A}_1 \bar{B}_0 u_0 + \bar{A}_k \cdots \bar{A}_2 \bar{B}_1 u_1 + \cdots + \bar{B}_k u_k)$$

左からC_kを掛けるとy_kとなる

となるので**scanアルゴリズムによる並列化が可能**。
（S6以前にも、S5モデルで導入された実績あり [2]）

詳細は[2]のAppendix Hを参照されたい

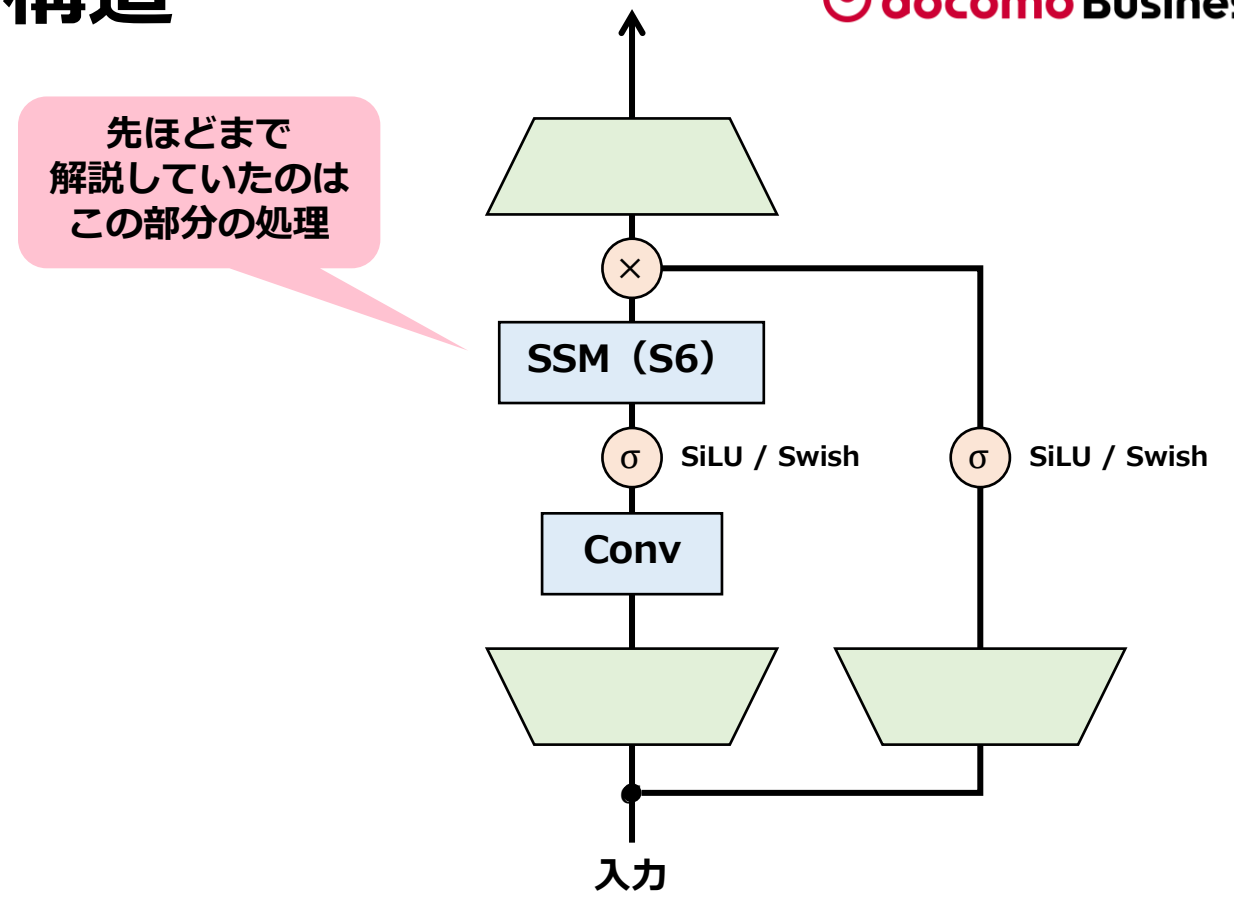
Mamba：Mambaブロックの構造

Mambaブロックは右図の構造をもつ。

- Conv層で局所的な時間依存性をとらえる

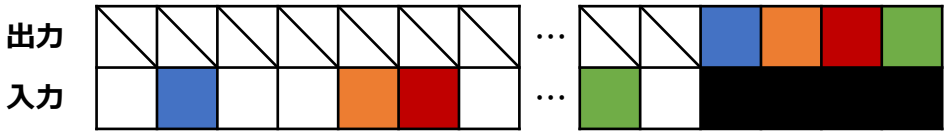
- **SSMブロック内で以下の手順で出力を計算：**
  1. **入力 u_t から Δ_t, B_t, C_t を計算**
  2. **Δ_t で離散化し、$\overline{A}_t, \overline{B}_t$ を計算**
  3. **parallel scanで出力 y_t を計算**

- もう一本のパス（ゲート機構）で情報を取捨選択



Mamba：Selective Copyingの実験結果

- アーキテクチャに依らず、**selective SSMを用いたモデルで高い精度**
- 時不変モデルでは、強力なアーキテクチャでも失敗している
  - gating機構があっても失敗しており、**gating機構のみでは選択性の獲得には不十分であることが示唆**される



Selective Copying

入力中の重要なトークンを記憶するタスク。
ただし、間にランダムな長さのノイズトークンがある。

MODEL	ARCH.	LAYER	ACC.
S4	No gate	S4	18.3
-	No gate	S6	**97.0**
H3	H3	S4	57.0
Hyena	H3	Hyena	30.1
-	H3	S6	**99.7**
-	Mamba	S4	56.4
-	Mamba	Hyena	28.4
Mamba	Mamba	S6	**99.8**

各モデルのSelection Copyingのaccuracy [3]

Mamba：Induction Headsの実験結果

- Mambaは訓練データの約4000倍もの長さのテストデータに対しても完璧に外挿できている
  - 選択的でないモデルでは訓練データの2倍の長さ程度が限界
- Attention系のモデルではメモリの制約から$2^{14} = 16384$までのみでの実験となっている

Induction Heads

入力からパターンを見出し、次のトークンを予測するタスク。
図では「黒の次は青」というパターンをもとに推論を行っている。

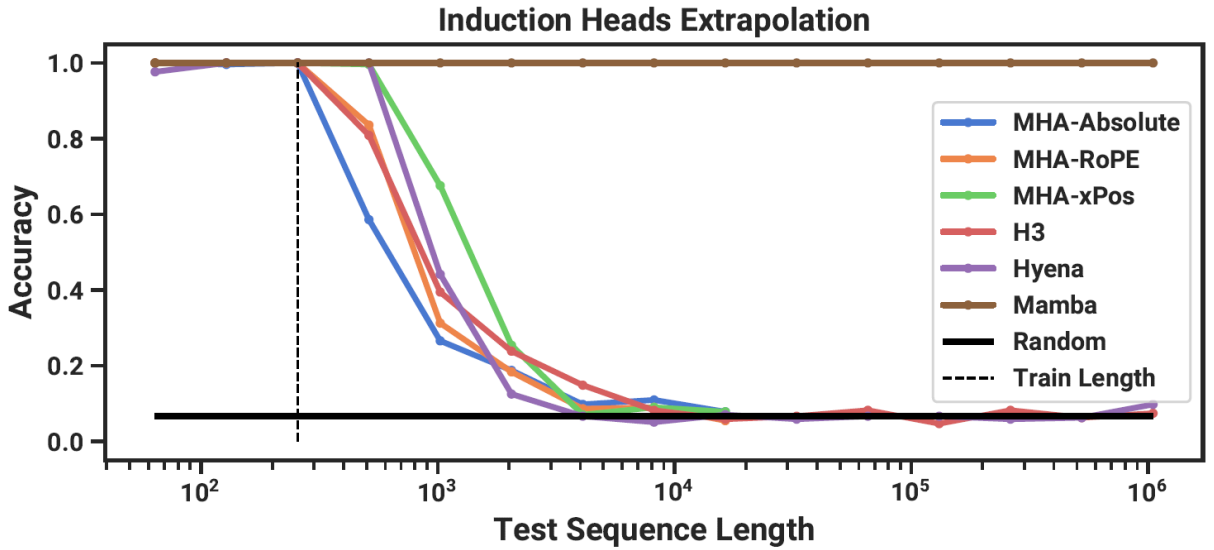


Figure 3: (**Induction Heads.**) Models are trained on sequence length $2^8 = 256$, and tested on increasing sequence lengths of $2^6 = 64$ up to $2^{20} = 1048576$. Full numbers in Table 10.

Induction Headsのaccuracy。横軸はテスト時のシーケンス長。 [3]

Mamba：言語モデルでの実験結果

つなごう。驚きを。幸せを。
docomo Business

各種ベンチマークにおいて、Mambaは同等のモデルサイズのオープンソースモデルを超えるスコアを記録した

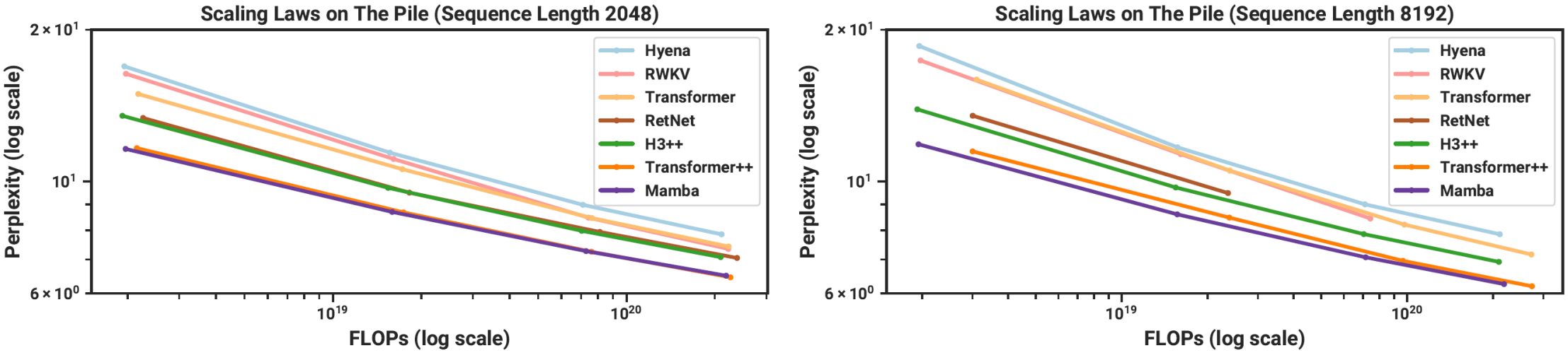
MODEL	TOKEN.	PILE PPL ↓	LAMBADA PPL ↓	LAMBADA ACC ↑	HELLASWAG ACC ↑	PIQA ACC ↑	ARC-E ACC ↑	ARC-C ACC ↑	WINOGRANDE ACC ↑	AVERAGE ACC ↑
Hybrid H3-360M	GPT2	—	12.58	48.0	41.5	68.1	51.4	24.7	54.1	48.0
Pythia-410M	NeoX	9.95	10.84	51.4	40.6	66.9	52.1	24.6	53.8	48.2
Mamba-370M	NeoX	**8.28**	**8.14**	**55.6**	**46.5**	**69.5**	**55.1**	**28.0**	**55.3**	**50.0**
Pythia-1B	NeoX	7.82	7.92	56.1	47.2	70.7	57.0	27.1	53.5	51.9
Mamba-790M	NeoX	**7.33**	**6.02**	**62.7**	**55.1**	**72.1**	**61.2**	**29.5**	**56.1**	**57.1**
GPT-Neo 1.3B	GPT2	—	7.50	57.2	48.9	71.1	56.2	25.9	54.9	52.4
Hybrid H3-1.3B	GPT2	—	11.25	49.6	52.6	71.3	59.2	28.1	56.9	53.0
OPT-1.3B	OPT	—	6.64	58.0	53.7	72.4	56.7	29.6	59.5	55.0
Pythia-1.4B	NeoX	7.51	6.08	61.7	52.1	71.0	60.5	28.5	57.2	55.2
RWKV-1.5B	NeoX	7.70	7.04	56.4	52.5	72.4	60.5	29.4	54.6	54.3
Mamba-1.4B	NeoX	**6.80**	**5.04**	**64.9**	**59.1**	**74.2**	**65.5**	**32.8**	**61.5**	**59.7**
GPT-Neo 2.7B	GPT2	—	5.63	62.2	55.8	72.1	61.1	30.2	57.6	56.5
Hybrid H3-2.7B	GPT2	—	7.92	55.7	59.7	73.3	65.6	32.3	61.4	58.0
OPT-2.7B	OPT	—	5.12	63.6	60.6	74.8	60.8	31.3	61.0	58.7
Pythia-2.8B	NeoX	6.73	5.04	64.7	59.3	74.0	64.1	32.9	59.7	59.1
RWKV-3B	NeoX	7.00	5.24	63.9	59.6	73.7	67.8	33.1	59.6	59.6
Mamba-2.8B	NeoX	**6.22**	**4.23**	**69.2**	**66.1**	**75.2**	**69.7**	**36.3**	**63.5**	**63.3**
GPT-J-6B	GPT2	–	4.10	68.3	66.3	75.4	67.0	36.6	64.1	63.0
OPT-6.7B	OPT	–	4.25	67.7	67.2	76.3	65.6	34.9	65.5	62.9
Pythia-6.9B	NeoX	6.51	4.45	67.1	64.0	75.2	67.3	35.5	61.3	61.7
RWKV-7.4B	NeoX	6.31	4.38	67.2	65.5	76.1	67.8	37.5	61.0	62.5

各種ベンチマークにおけるスコア [3]

Mamba：言語モデルにおけるScaling Laws

つなごう。驚きを。幸せを。
docomo Business

Mambaは1Bパラメータまでのスケーリング則において、Transformer++（PaLM, LLaMaベースの非常に強力なTransformerの学習レシピ）と同等以上の性能を示した。



Pileで学習されたモデルのスケーリング則 [3]

Mamba-2：論文概要

つなごう。驚きを。幸せを。
docomo Business

“Transformers are SSMs: Generalized Models and Efficient Algorithms through Structured State Space Duality“ (ICML2024)

Author: T. Dao and A. Gu

- Mambaの改良版である**Mamba-2**を提案
- Mambaからの変更点として、Aが対角行列→スカラー行列に制限されている
- **SSDアルゴリズム**により、学習の高速化を実現。

- SSMとAttentionを結びつける枠組である**SSD（structured state space duality）**を提案

Mamba-2：SSDアルゴリズムの概要

つなごう。驚きを。幸せを。
docomo Business

SSMの出力は

$$y_t = \sum_{s=0}^{t} C_t^T A_t A_{t-1} \cdots A_{s+1} B_s \, u_s,$$

なので、行列積を用いると $y = Mu$という形でまとめることができる。

Mamba-2ではこの行列演算を、Mの**semiseparability**を利用した**SSDアルゴリズム**により高速に計算する。

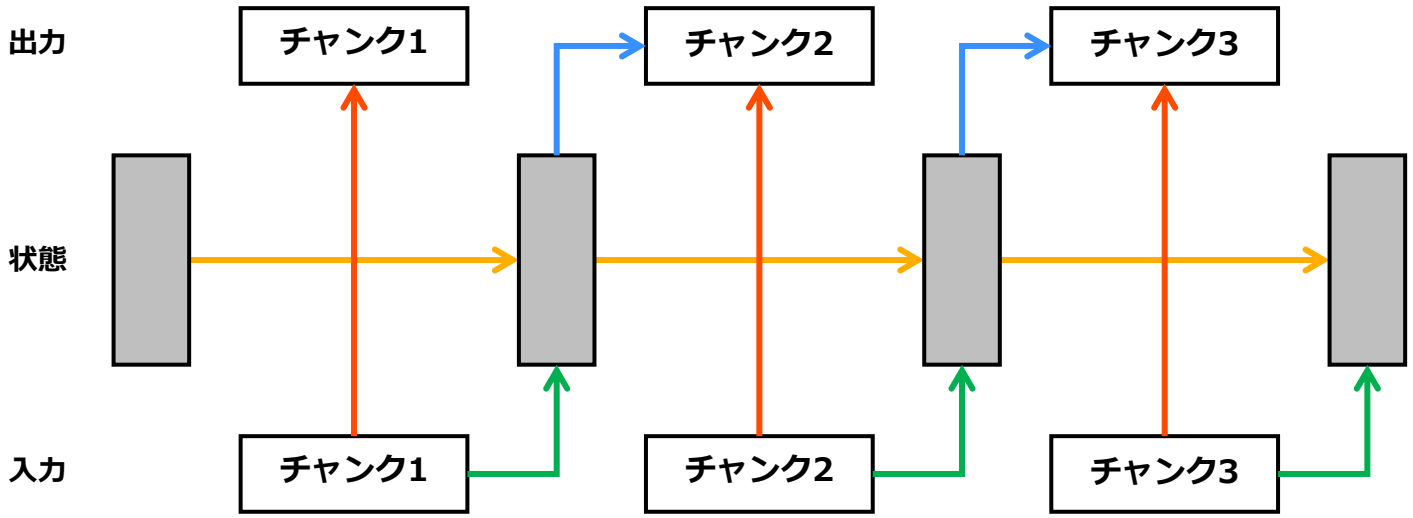
$$\left[\begin{array}{ccc|ccc|ccc}
C_0^\top A_{0:0} B_0 & & & & & & & & \\
C_1^\top A_{1:0} B_0 & C_1^\top A_{1:1} B_1 & & & & & & & \\
C_2^\top A_{2:0} B_0 & C_2^\top A_{2:1} B_1 & C_2^\top A_{2:2} B_2 & & & & & & \\
\hline
C_3^\top A_{3:0} B_0 & C_3^\top A_{3:1} B_1 & C_3^\top A_{3:2} B_2 & C_3^\top A_{3:3} B_3 & & & & & \\
C_4^\top A_{4:0} B_0 & C_4^\top A_{4:1} B_1 & C_4^\top A_{4:2} B_2 & C_4^\top A_{4:3} B_3 & C_4^\top A_{4:4} B_4 & & & & \\
C_5^\top A_{5:0} B_0 & C_5^\top A_{5:1} B_1 & C_5^\top A_{5:2} B_2 & C_5^\top A_{5:3} B_3 & C_5^\top A_{5:4} B_4 & C_5^\top A_{5:5} B_5 & & & \\
\hline
C_6^\top A_{6:0} B_0 & C_6^\top A_{6:1} B_1 & C_6^\top A_{6:2} B_2 & C_6^\top A_{6:3} B_3 & C_6^\top A_{6:4} B_4 & C_6^\top A_{6:5} B_5 & C_6^\top A_{6:6} B_6 & & \\
C_7^\top A_{7:0} B_0 & C_7^\top A_{7:1} B_1 & C_7^\top A_{7:2} B_2 & C_7^\top A_{7:3} B_3 & C_7^\top A_{7:4} B_4 & C_7^\top A_{7:5} B_5 & C_7^\top A_{7:6} B_6 & C_7^\top A_{7:7} B_7 & \\
C_8^\top A_{8:0} B_0 & C_8^\top A_{8:1} B_1 & C_8^\top A_{8:2} B_2 & C_8^\top A_{8:3} B_3 & C_8^\top A_{8:4} B_4 & C_8^\top A_{8:5} B_5 & C_8^\top A_{8:6} B_6 & C_8^\top A_{8:7} B_7 & C_8^\top A_{8:8} B_8
\end{array}\right]$$

Mの構造　　※$A_{t:s} = A_t A_{t-1} \cdots A_{s+1}$

Mamba-2：SSDアルゴリズム

SSDアルゴリズムでは入出力を固定長のチャンクに分割して以下の計算を行う：

1. 各チャンク内での入力から出力への影響を算出（チャンクごとに並列化可能）
2. 入力の各チャンクがチャンクの末端の状態に与える影響を算出（チャンクごとに並列化可能）
3. チャンク末端の状態が、次のチャンク末端の状態に与える影響を伝播（→各チャンク末端状態の計算完了）
   ※この部分は逐次計算になるが、ステップ数はチャンクの個数分でよい
4. 算出されたチャンク末端の状態から、次チャンクの出力に与える影響を算出（チャンクごとに並列化可能）



Mamba-2：SSDアルゴリズムの汎用性

つなごう。驚きを。幸せを。
docomo Business

SSDアルゴリズムはSSMの計算だけではなく、より一般に**1-semiseparable行列Mによる行列演算$y = Mu$**の高速演算を可能にする汎用性の高いアルゴリズムである。

（下三角）行列Mの下三角成分に含まれる任意の部分行列の階数が高々Nのとき、Mは**N-semiseparable**であるという。

※1-semiseparable行列Mは、SSMの計算式$y = Mu$で現れるMと同様の構造に分解可能という性質がある

SSMの出力計算においてはMが1-semiseparableであるため、SSDアルゴリズムで高速化可能。

Mamba-2：Structured State-Space Duality

つなごう。驚きを。幸せを。 docomo Business

Mamba-2はSSMベースのモデルであるが、（masked）attentionと捉えることもできる。

実際、Mamba-2のSSMの出力は $y = Mu$ と表せる。（Mは$M_{ij} = C_i^T A_i \cdots A_{j+1} B_j$ を成分とする下三角行列）

A 関連の要素をマスク行列Lにまとめると

$$y = Mu = L \circ C \; B^T \cdot u,$$

mask query key value

となり、これはLをマスク行列、C をQuery、B をKey、uをValueとする**masked attention**の式となる。

[4]では上のようにSSMとattentionを同一視する理論的枠組である**SSD (structured state-space duality)**が提唱された。

※ “SSD”はこのような同一視が可能なモデルのことを指すこともある。

まとめ

- **S4** [1]
  - HiPPO [5]の理論に基づいたDPLR行列を利用し、SSMにおける長期記憶性を向上させた

- **Mamba** [3]
  - SSMのパラメータB, C, Δを入力に依存させることで選択性を持たせた
  - SSMの行列Aは対角行列に制限
  - S5 [2]でも用いられていたParallel scan [6]による並列化

- **Mamba-2** [4]
  - SSMの行列Aはスカラー行列に制限
  - SSDアルゴリズムによる高速化
  - AttentionとSSMの双対性（SSD）の提唱

参考文献

[1] Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.

[2] Smith, J. T., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling. In *The International Conference on Learning Representations (ICLR)*, 2023.

[3] Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[3] Gu, A., & Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Language Modeling (COLM),* 2024.

[4] Dao, T. and Gu, A. Transformers are SSMs: Generalized models and efficient algorithms with structured state space duality. In *International Conference on Machine Learning (ICML)*, 2024.

[5] Gu, A., Dao, T., Ermon, S., Rudra, A., and Ré, C. HIPPO: Recurrent memory with optimal polynomial projections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] Blelloch, G. E. Prefix sums and their applications. 1990.