# Advanced Time Series Analysis

## State Space Models

Anshul Thakur & Prof. David Clifton

Computational Health Informatics Lab, IBME
Department of Engineering Science
University of Oxford

# Table of Contents

# Table of Contents

# Structured State Space Model: A Linear RNN?

- SSMs use **linear state transitions** rather than non-linear hidden states
- **Time- and input-invariance:** rules governing state transitions and output generation do not change over the sequence

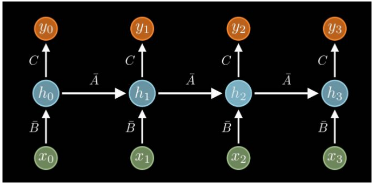


Figure: Discrete Linear SSM.

## Discrete Linear SSM

At time step *t* the system follows:

$$\boxed{\mathbf{h}_t = \mathbf{A}\,\mathbf{h}_{t-1} + \mathbf{B}\,\mathbf{x}_t} \qquad \text{(state update)}$$

$$\boxed{\mathbf{y}_t = \mathbf{C}\,\mathbf{h}_t + \mathbf{D}\,\mathbf{x}_t} \qquad \text{(output)}$$

**Notations:**

$\mathbf{x}_t \in \mathbb{R}^{d_{\text{in}}}$, $\mathbf{h}_t \in \mathbb{R}^{n}$, $\mathbf{y}_t \in \mathbb{R}^{d_{\text{out}}}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times d_{\text{in}}}$, $\mathbf{C} \in \mathbb{R}^{d_{\text{out}} \times n}$, $\mathbf{D} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$.

# From Recurrence to Convolution

## Discrete SSM:

$$\mathbf{h}_t = \mathbf{A}\mathbf{h}_{t-1} + \mathbf{B}\mathbf{x}_t, \qquad \mathbf{y}_t = \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{x}_t$$

## Unroll the recurrence:

$$\mathbf{h}_1 = \mathbf{B}\mathbf{x}_1,$$

$$\mathbf{h}_2 = \mathbf{A}\mathbf{h}_1 + \mathbf{B}\mathbf{x}_2 = \mathbf{A}\mathbf{B}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_2,$$

$$\mathbf{h}_3 = \mathbf{A}\mathbf{h}_2 + \mathbf{B}\mathbf{x}_3 = \mathbf{A}^2\mathbf{B}\mathbf{x}_1 + \mathbf{A}\mathbf{B}\mathbf{x}_2 + \mathbf{B}\mathbf{x}_3,$$

$$\mathbf{h}_t = \sum_{k=0}^{t-1} \mathbf{A}^k\mathbf{B}\,\mathbf{x}_{t-k}.$$

## Substitute into output:

$$\mathbf{y}_t = \mathbf{C}\mathbf{h}_t + \mathbf{D}\mathbf{x}_t = \sum_{k=0}^{t-1} \mathbf{C}\mathbf{A}^k\mathbf{B}\,\mathbf{x}_{t-k} + \mathbf{D}\mathbf{x}_t.$$

## Convolution:

$$K[k] = \begin{cases} \mathbf{D} + \mathbf{C}\mathbf{B}, & k = 0, \\ \mathbf{C}\mathbf{A}^k\mathbf{B}, & k > 0, \end{cases}$$

$$\boxed{\mathbf{y}_t = \sum_{k=0}^{t} K[k]\,\mathbf{x}_{t-k}} \iff \boxed{\mathbf{y} = K * \mathbf{x}}$$

## Intuition:

- Each $\mathbf{A}^k$ describes how information propagates over $k$ steps.
- $K[k]$ is the system's *impulse response* (weight for input lag $k$).
- Convolution sums all lagged contributions in parallel.

# Practical Implications

- $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ are learned end-to-end. The kernel weights $K[k]$ are *generated* from these matrices — not learned independently as in CNNs.
- **Pre-computation:** Build $K[0{:}T-1]$ once (using powers of $\mathbf{A}$), then apply conv1d or FFT convolution: $\mathbf{y} = K * \mathbf{x}$. No Python-level time loop — full GPU parallelism.
- **Efficiency: RNNs:** $O(T)$ sequential steps. **SSMs:** $O(T \log T)$ parallel convolution — much faster on GPUs, enabling efficient modeling of very long sequences (audio, sensors, text).
- **Modeling benefit:** Combines **short-term** direct effects (via $\mathbf{D}$) and **long-range** memory (via $\mathbf{A}^k$).

# Table of Contents

# Structured vs Selective Models

DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

**Algorithm 1** SSM (S4)

**Input:** $x : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
**Output:** $y : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
1: $\boldsymbol{A} : (\mathtt{D}, \mathtt{N}) \leftarrow \mathsf{Parameter}$
$\triangleright$ Represents structured $N \times N$ matrix
2: $\boldsymbol{B} : (\mathtt{D}, \mathtt{N}) \leftarrow \mathsf{Parameter}$
3: $\boldsymbol{C} : (\mathtt{D}, \mathtt{N}) \leftarrow \mathsf{Parameter}$
4: $\Delta : (\mathtt{D}) \leftarrow \tau_\Delta(\mathsf{Parameter})$
5: $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathtt{D}, \mathtt{N}) \leftarrow \mathsf{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
6: $y \leftarrow \mathsf{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
$\triangleright$ Time-invariant: recurrence or convolution
7: **return** $y$

$$\bar{\mathbf{A}} = e^{\Delta \mathbf{A}} \qquad \bar{\mathbf{B}} = (e^{\Delta \mathbf{A}} - \mathbf{I})\, \mathbf{A}^{-1}\mathbf{B}$$

**Algorithm 2** SSM + Selection (S6)

**Input:** $x : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
**Output:** $y : (\mathtt{B}, \mathtt{L}, \mathtt{D})$
1: $\boldsymbol{A} : (\mathtt{D}, \mathtt{N}) \leftarrow \mathsf{Parameter}$
$\triangleright$ Represents structured $N \times N$ matrix
2: $\boldsymbol{B} : (\mathtt{B}, \mathtt{L}, \mathtt{N}) \leftarrow s_B(x)$
3: $\boldsymbol{C} : (\mathtt{B}, \mathtt{L}, \mathtt{N}) \leftarrow s_C(x)$
4: $\Delta : (\mathtt{B}, \mathtt{L}, \mathtt{D}) \leftarrow \tau_\Delta(\mathsf{Parameter} + s_\Delta(x))$
5: $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}} : (\mathtt{B}, \mathtt{L}, \mathtt{D}, \mathtt{N}) \leftarrow \mathsf{discretize}(\Delta, \boldsymbol{A}, \boldsymbol{B})$
6: $y \leftarrow \mathsf{SSM}(\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \boldsymbol{C})(x)$
$\triangleright$ Time-varying: recurrence (*scan*) only
7: **return** $y$

$$s_B(\mathbf{x}) = \mathrm{Linear}_N(\mathbf{x}) \quad s_\Delta(x) = \mathrm{Broadcast}_D(\mathrm{Linear}_1(x)),$$

$$\bar{\mathbf{A}}_{b,l,d,:,:} = e^{\Delta_{b,l,d}\,\mathbf{A}} \quad \in \mathbb{R}^{N \times N}$$

$$\bar{\mathbf{B}}_{b,l,d,:} = (e^{\Delta_{b,l,d}\,\mathbf{A}} - \mathbf{I})\,\mathbf{A}^{-1}\mathbf{B}_{b,l,:} \quad \in \mathbb{R}^{N}$$

# Scanning over Recurrence

| | | | | | | |
|---|---|---|---|---|---|---|
| Initial array | 9 | 6 | 7 | 10 | 8 | 7 |
| Prefix-Sum | 9 | 15 | 22 | 32 | 40 | 47 |

$$h_t = \overline{A}h_{t-1} + \overline{B}x_t$$

The recurrent formula of the SSM model can also be thought of as a scan operation, in which each state is the sum of the previous state and the current input.

| | | | | | | |
|---|---|---|---|---|---|---|
| Model input | $x_0$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ |
| Scan output | $h_0$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $h_5$ |

- Scanning can be parallelized.
- Model selective space models such as Mamba use effective parallel scans.

# Table of Contents

# Further Reading

- **Mamba & Mamba V2:** State Space Models enabling input-dependent dynamics
- **S4 / S5 Families:** Structured State Space Models introducing stable, parallelizable sequence representations.
- **Hybrid Models:** Combining SSMs with Transformers or Convolutional layers for multimodal and hierarchical tasks.