Deep Learning Additional Homework - Mamba Discretization

110550088 李杰穎

April 2, 2025

1 Introduction

Mamba [1] is a ground-breaking state space model (SSM) published in 2023. Developed by researchers Albert Gu and Tri Dao, the model introduces a novel approach to sequence modeling that addresses key limitations in existing transformer and state space architectures. Unlike traditional transformer models that struggle with long-range dependencies and computational efficiency, Mamba offers a linear-time sequence modeling technique with selective state spaces. The model demonstrates remarkable performance across various tasks, challenging the dominance of attention-based architectures by providing a more efficient alternative for processing sequential data. By selectively processing information and maintaining a compact state representation, Mamba represents a significant advancement in the field of machine learning and neural network design.

Compared with Structured State Space sequence model (S4) [2], Mamba introduces a selective state space mechanism with a linear-time complexity, enabling more efficient and dynamic processing of sequential data. While S4 uses a uniform continuous-time state space model with quadratic computational complexity, Mamba employs a selective scanning algorithm that can dynamically focus on relevant parts of the input sequence. This approach allows Mamba to learn which information is most important, providing a more adaptable and computationally efficient alternative to both S4 and traditional transformer architectures. The model achieves this through a novel hardware-friendly design with fused kernels, making it particularly effective for long-sequence modeling tasks and demonstrating significant improvements in both computational efficiency and model performance.

In this homework, we aim to derive the discretization process in the Structured State Space model, including Mamaba [1] and S4 [2].

2 Discretization

The blog post, titled "Structured State Spaces: Combining Continuous-Time, Recurrent, and Convolutional Models" [3], provides a comprehensive overview of the structure state space model. In this blog post, it mentions that why discretization is a crucial step for the state space model. As in real world applications, we can only obtain discrete data, and the discretization process is a key step to convert continuous-time state space models into discrete-time models.

As described in the original Mamba [1] paper, the discretization process turn continuous parameters, A, B and Δ into discrete-time parameters, $\bar{A}$ and $\bar{B}$. Given the state space model, the continuous-time state space model is defined as follows:

$$\frac{\mathrm{d}h(t)}{\mathrm{d}t} = Ah(t) + Bx(t) \tag{1}$$

$$y(t) = Ch(t) \tag{2}$$

The discretization process for zero-order hold is defined as follows:

$$\bar{A} = \exp(\Delta A) \tag{3}$$

$$\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \tag{4}$$

After the discretization process, we can obtain the discrete-time state space model:

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t\,, \tag{5}$$

$$y_t = Ch_t\,, \tag{6}$$

where h_t is the hidden state at time step t, x_t is the input at time step t, and y_t is the output at time step t. The parameters $\bar{A}$ and $\bar{B}$ are the discretized versions of the continuous-time parameters A and B, respectively. The parameter Δ is the time interval between two consecutive time steps.

Now, we prove the above discretization process. The discretization process is based on the zero-order hold (ZOH) method, which is a common technique used to convert continuous-time systems into discrete-time systems. The ZOH method assumes that the input signal is held constant over each sampling interval, which allows us to derive the discrete-time state space model from the continuous-time state space model. The ZOH method can be derived from the following equations:

$$h(t) = \exp(At)h(0) + \int_0^t \exp(A(t-\tau))Bx(\tau)\,\mathrm{d}\tau \tag{7}$$

$$y(t) = Ch(t) \tag{8}$$

where $h(0)$ is the initial state at time $t = 0$, and $x(\tau)$ is the input signal at time τ. The first term in the equation represents the state evolution due to the initial state, while the second term represents the contribution of the input signal to the state evolution.

To derive the discrete-time state space model[1], we can sample the continuous-time state space model at discrete time steps. Let $t_k = k\Delta$ be the discrete time steps, where k is an integer and Δ is the sampling interval. We can rewrite the continuous-time state space model as follows:

$$h'(t) = Ah(t) + Bx(t). \tag{9}$$

We know that:

$$\frac{\mathrm{d}}{\mathrm{d}t}e^{At} = Ae^{At} = e^{At}A. \tag{10}$$

We can rewrite Equation 9 as:

$$e^{-At}h'(t) = e^{-At}Ah(t) + e^{-At}Bx(t) \tag{11}$$

$$e^{-At}h'(t) - Ae^{-At}h(t) = e^{-At}Bx(t) \tag{12}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(e^{-At}h(t)) = e^{-At}Bx(t) \tag{13}$$

and by integrating both sides, we have:

$$e^{-At}h(t) = h(0) + \int_0^t e^{-A\tau}Bx(\tau)\,\mathrm{d}\tau \tag{14}$$

$$h(t) = e^{At}h(0) + \int_0^t e^{A(t-\tau)}Bx(\tau)\,\mathrm{d}\tau \tag{15}$$

Equation 15 is the analytical solution to the continuous model.

Next, we want to discretize the continuous-time state space model. We can sample the continuous-time state space model at discrete time steps. We denote $h_k = h(k\Delta)$, $x_k = x(k\Delta)$, where k is an integer and Δ is the sample interval. We can rewrite Equation 15 as:

$$h_k = h(k\Delta) = e^{A(k\Delta)}h(0) + \int_0^{k\Delta} e^{A(k\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau \tag{16}$$

And also we can rewrite h_{k+1} as:

$$h_{k+1} = h((k+1)\Delta) = e^{A((k+1)\Delta)}h(0) + \int_0^{(k+1)\Delta} e^{A((k+1)\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau \tag{17}$$

$$= e^{Ak\Delta}e^{A\Delta}h(0) + \int_0^{k\Delta} e^{A((k+1)\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau + \int_{k\Delta}^{(k+1)\Delta} e^{A((k+1)\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau \tag{18}$$

$$= e^{Ak\Delta}e^{A\Delta}h(0) + e^{A\Delta}\int_0^{k\Delta} e^{A(k\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau + \int_{k\Delta}^{(k+1)\Delta} e^{A((k+1)\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau \tag{19}$$

$$= e^{A\Delta}\left[e^{Ak\Delta}h(0) + \int_0^{k\Delta} e^{A(k\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau\right] + \int_{k\Delta}^{(k+1)\Delta} e^{A((k+1)\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau \tag{20}$$

$$= e^{A\Delta}h_k + \int_{k\Delta}^{(k+1)\Delta} e^{A((k+1)\Delta-\tau)}Bx(\tau)\,\mathrm{d}\tau \tag{21}$$

[1]The derivation is referenced from [4]

Now, let's focus on the integral part. By the assumption of zero-order hold, we can assume that $x(\tau)$ is constant over the interval $[k\Delta, (k+1)\Delta]$[2]. Therefore, we can rewrite the integral as:

$$\int_{k\Delta}^{(k+1)\Delta} e^{A((k+1)\Delta-\tau)} Bx(\tau)\,\mathrm{d}\tau = \int_{k\Delta}^{(k+1)\Delta} e^{A((k+1)\Delta-\tau)} Bx_{k+1}\,\mathrm{d}\tau \tag{22}$$

$$= Bx_{k+1} \int_{k\Delta}^{(k+1)\Delta} e^{A((k\Delta+\Delta-\tau)}\,\mathrm{d}\tau \tag{23}$$

Let $v(\tau) = k\Delta + \Delta - \tau$, we know that $\mathrm{d}v = -\,\mathrm{d}\tau$. Therefore, we have:

$$Bx_{k+1} \int_{k\Delta}^{(k+1)\Delta} e^{A((k\Delta+\Delta-\tau)}\,\mathrm{d}\tau = -Bx_{k+1} \int_{\Delta}^{0} e^{Av(\tau)}\,\mathrm{d}v(\tau) \tag{24}$$

$$= Bx_{k+1} \int_{0}^{\Delta} e^{Av(\tau)}\,\mathrm{d}v(\tau) \tag{25}$$

$$= A^{-1}\left[e^{Av}\right]_0^{\Delta} Bx_{k+1} \tag{26}$$

$$= A^{-1}\left[e^{A\Delta} - I\right] Bx_{k+1} \tag{27}$$

$$= \frac{A^{-1}}{\Delta}\left[e^{A\Delta} - I\right] Bx_{k+1} \cdot \Delta \tag{28}$$

$$= (A\Delta)^{-1}\left[\exp(A\Delta) - I\right](B\Delta)x_{k+1}. \tag{29}$$

By rewriting Equation 21 with the above integral, we have:

$$h_{k+1} = \exp(A\Delta)h_k + (A\Delta)^{-1}\left[\exp(A\Delta) - I\right](B\Delta)x_{k+1}. \tag{30}$$

Thus, we prove that the discretization process is:

$$\bar{A} = \exp(\Delta A) \tag{31}$$

$$\bar{B} = (\Delta A)^{-1}\left[\exp(\Delta A) - I\right] \cdot \Delta B \tag{32}$$

where $\bar{A}$ and $\bar{B}$ are the discretized versions of the continuous-time parameters A and B, respectively. The parameter Δ is the time interval between two consecutive time steps.

References

[1] Albert Gu and Tri Dao. "Mamba: Linear-time sequence modeling with selective state spaces". In: *arXiv preprint arXiv:2312.00752* (2023).

[2] Albert Gu, Karan Goel, and Christopher Ré. "Efficiently modeling long sequences with structured state spaces". In: *arXiv preprint arXiv:2111.00396* (2021).

[3] Albert Gu et al. *Structured State Spaces: Combining Continuous-Time, Recurrent, and Convolutional Models.* Hazy Research Blog. Accessed on 2025-04-02. Jan. 2022. URL: `https://hazyresearch.stanford.edu/blog/2022-01-14-s4-3` (visited on 04/02/2025).

[4] Wikipedia contributors. *Discretization.* `https://en.wikipedia.org/wiki/Discretization`. Accessed: 2025-04-02. 2025. URL: `https://en.wikipedia.org/wiki/Discretization?oldformat=true`.

[2] We take the right value x_{k+1} in order to match the original derivation.