https://arxiv.org/pdf/2312.00752.pdf

Background

Attention layer

- Can't model outside of finite window
- Quadratic scaling with respect to window length

Many subquadratic- time architectures such as linear attention, gated convolution, recurrent models, structured state space models to address quadratic attention (transformer) but have not performed as well on important modalities

- Inability to perform content- based reasoning

SSM: combination of RNNs and CNNs with inspiration from classical state space models

- Can be computed with linear or near linear scaling in sequence length
- Principled mechanisms for modeling long range dependencies
- Successful in domains with continuous signal data such as audio and vision
- Less effective at modeling discrete and information dense data such as text

Idea

Improvements (MAMBA)

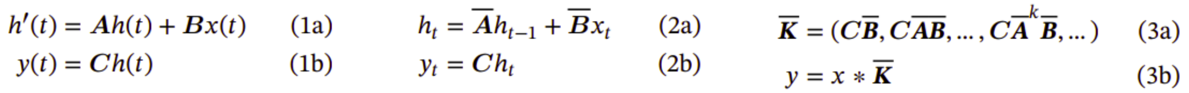
- Selection mechanism: Letting SSM parameters be functions of input address weakness with discrete modalities: model can selectively propagate or forget information along sequence length dimension based on current token
  - Lets model efficiently select data in an input dependent manner (focus on or ignore parts of input)
- Hardware aware parallel algorithms in recurrent mode (linear with sequence length)
  - Scan instead of convolution but doesn't materialize expanded state to avoid IO between levels of GPU memory hierarchy
- Architecture: combine design of prior SSM with MLP block of transformer into single block
  - Fully recurrent
  - High quality: selectivity for strong performance on dense modalities such as language and genomics
  - Fast training and inference: computation and memory scale linearly in sequence length, unrolling model autoregressively during inference requires constant time since it does not need cache of previous elements
  - Long context

Fast inference (5x higher than transformer), linear scaling in sequence length, performance improves on real data up to million length sequences

SSM

- Discretization: transforms “continuous parameters” to “discrete parameters”

  - Connections to resolution invariance (continuous time systems), proper normalization, gating mechanism of RNNs
- Computation
  - Convolution mode for efficient parallelizable training (whole input sequence is seen ahead of time)
  - Recurrent mode for efficient autoregressive inference (inputs are seen one timestep at a time)
- Linear Time Invariance: model's dynamics are constant through time
  - Fundamental limitations in modeling certain types of data (because of efficiency constraints), so this paper removes this constraint while overcoming efficiency bottleneck
- Structure and Dimensions
  - Impose diagonal structure on A matrix
- General State Space Models (s4 in this case)



$$h'(t) = \boldsymbol{A}h(t) + \boldsymbol{B}x(t) \quad (1a)$$
$$y(t) = \boldsymbol{C}h(t) \quad (1b)$$
$$h_t = \overline{\boldsymbol{A}}h_{t-1} + \overline{\boldsymbol{B}}x_t \quad (2a)$$
$$y_t = \boldsymbol{C}h_t \quad (2b)$$
$$\overline{\boldsymbol{K}} = (\boldsymbol{C}\overline{\boldsymbol{B}}, \boldsymbol{C}\overline{\boldsymbol{A}\boldsymbol{B}}, \dots, \boldsymbol{C}\overline{\boldsymbol{A}}^{k}\overline{\boldsymbol{B}}, \dots) \quad (3a)$$
$$y = x * \overline{\boldsymbol{K}} \quad (3b)$$

- SSM Architectures: previous works as primary baselines
  - Linear attention: approximation of self attention using recurrence (degenerate linear SSM)
  - H3: generalized above recurrence, SSM sandwiched by two gated connections + standard local convolution
  - Hyena: same architecture as H3 but replace S4 layer with MLP global convolution
  - RetNet: additional gate to architecture and simpler SSM, multi head attention instead of convolutions (alternative parallelizable computation path)
  - RWKV: recent RNN using linear attention approximation (LTI recurrence: ratio of two SSMs)

Fundamental problem of sequence modeling is compressing context into a smaller state

- Attention
  - Doesn't compress context at all, requires explicitly storing entire context (KV cache) -> linear time inference and quadratic time training
- Recurrent
  - Finite state -> Constant time inference and linear time training
  - Limited by how well state compressed context

Selective Copying task (varies position of tokens to memorize)

- Content aware reasoning to memorize relevant tokens and filter out irrelevant ones
- Convolution models aren't good at this (spacing between inputs and outputs is varying and can't be modeled by static convolution kernels)

Induction heads

- Context aware reasoning to know when to produce correct output in appropriate context

- Recurrent models aren't good at this (can't affect hidden state passed along sequence in input- dependent way)

Efficiency vs effectiveness tradeoff characterized by how well they compress their state

- Efficient models have small state
- Effective models have state that contains all necessary information from context

Fundamental principle for building sequence models is selectivity or context- aware ability to focus on or filter out inputs into sequential state

- Selection mechanism controls how information propagates or interacts along sequence dimension

See Algorithm 2 for more details

- Made parameters functions of input
- Length dimension L -> model has changed from time invariant to time varying
- Loses equivalence to convolutions with implications to efficiency

Core limitation in usage of SSMs is their computational efficiency which is why S4 and all derivatives used LTI (non selective) models (global convolutions)

We want to maximize hidden state dimension without paying speed and memory costs

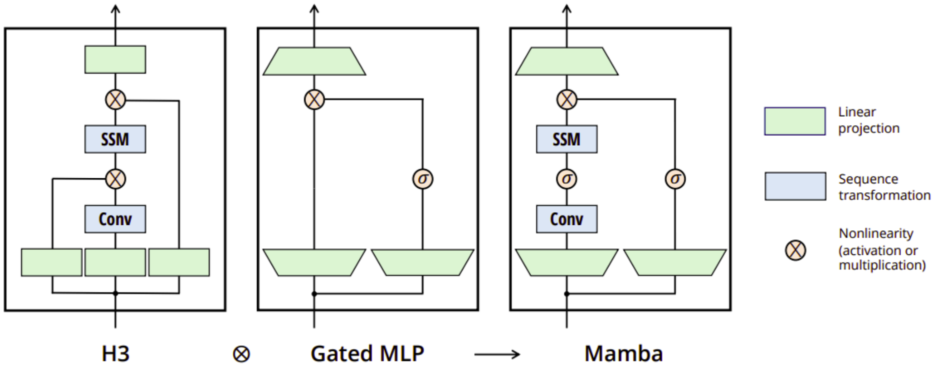
- Recurrent models such as SSMs balance tradeoff between expressivity and speed (models with larger hidden state are more effective but slower)
- Recurrent is more flexible than convolution but would require more computation so more efficient convolution mode sas introduced
- Prior LTI SSMs leverage dual recurrent- convolutional forms to increase effective state dimension

Selection mechanism overcomes limitations of LTI models

- Kernel fusion, parallel scan, recomputation

Fused selective scan layer has same memory requirements as optimized transformer implementation with FlashAttention

- They do a lot of HBM and SRAM stuff to speed up process
- Naive recurrent computation uses less than convolutional computation
    - For long sequences and not- too- large state dimension N, recurrent mode uses fewer FLOPs
- Sequential nature of recurrence and large memory usage (attempt to not materialize the full state h)



Classical gating mechanism of RNNs is an instance of our selection mechanism for SSMs

Interpretation

- Variable spacing through selectivity can filter out irrelevant noise tokens between inputs of interest (remove filler words like “um”)
- Filtering context: recurrent models can’t ignore irrelevant context while selective models can simply reset their state at any time
- Boundary resetting: selective SSMs and Transformers can create boundaries between multiple independent sequences
    - LTI will bleed information between sequences
- Interpretation of delta: delta controls how much to focus or ignore current input
    - Large delta resets state and focuses on current input while small delta persists state and ignores current input
    - SSMs are continuous systems discretized by a timestep delta
        - As delta approaches infinity, systems focuses on current input for longer (selecting it and forgetting state) while opposite is a transient (ignored) input
- Interpretation of A: doesn’t need need to be selective because delta is already selective (only interacts with model through delta)
- Interpretation of B and C:
    - Finer grained control over whether to let an input x into state h (B) or state into output y (C)
    - Content (input) -> B
    - Context (hidden state) -> C

Tasks

Synthetics: copying and induction heads

Audio and Genomics: modeling audio waveforms and DNA sequences

Language modeling: first linear time sequence model that achieves transformer quality performance