ECE1512 Project B

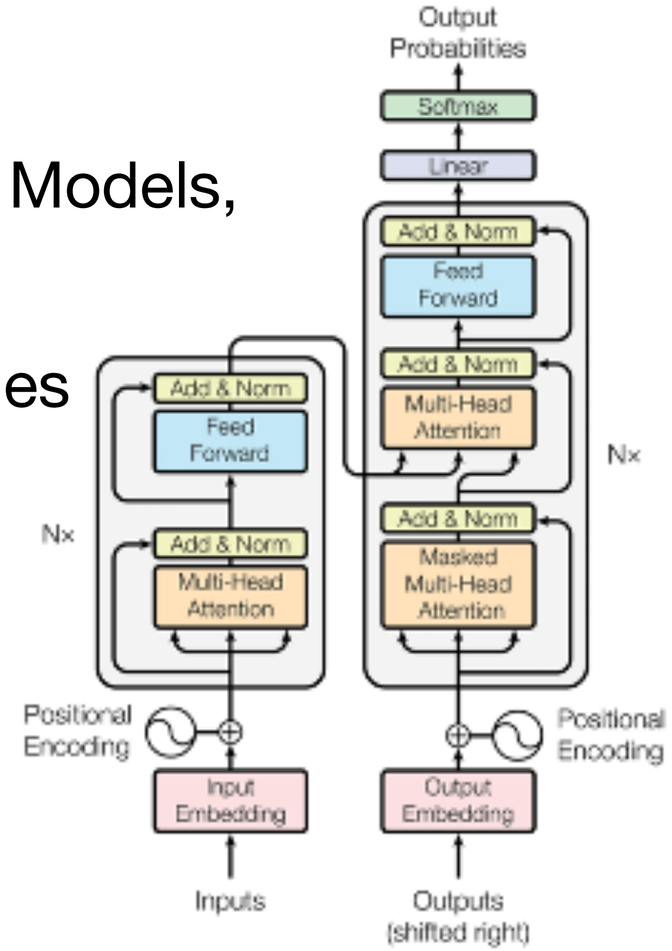
Introduction to State Space Models and Vision Language Models

Samir Khaki, November 20, 2024

Fundamentals of Transformers

Key Idea: Global Knowledge in Attention + Learnable Layers = Strong Sequence Model

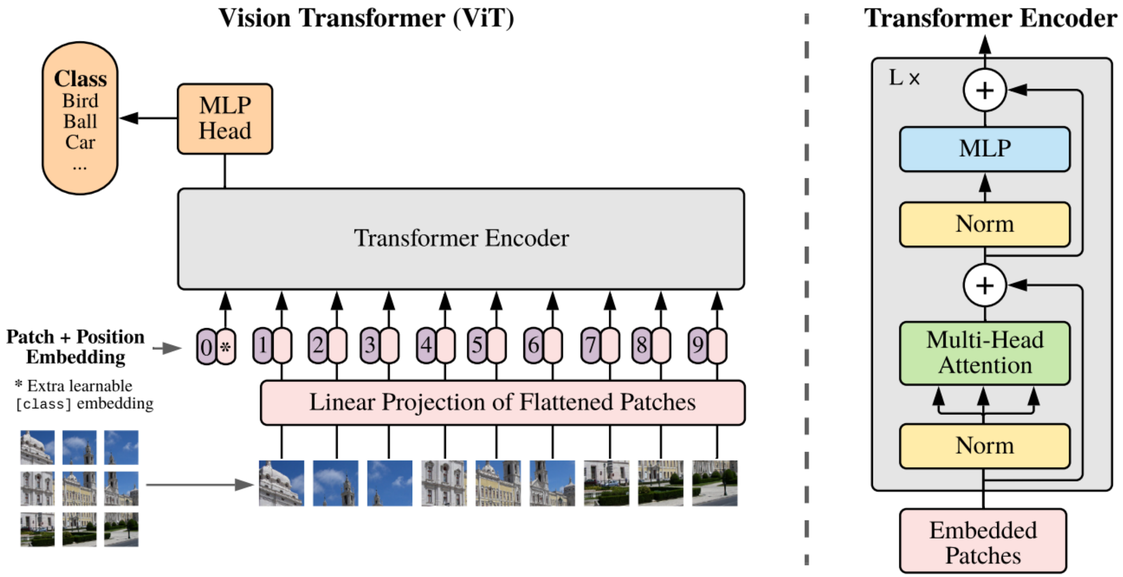
- Vaswani et al. 2017 —> “transformer”
  - *De-facto* Architecture for (LLM, Vision-LM, Foundation Models, etc..)
  - Features Attention to densely route **global** dependencies
  - **Quadratic** run time (n**2) in sequence length (n)



Fundamentals of Vision Transformers

Key Idea: Patchify the image + vanilla transformer = Strong Visual Encoder

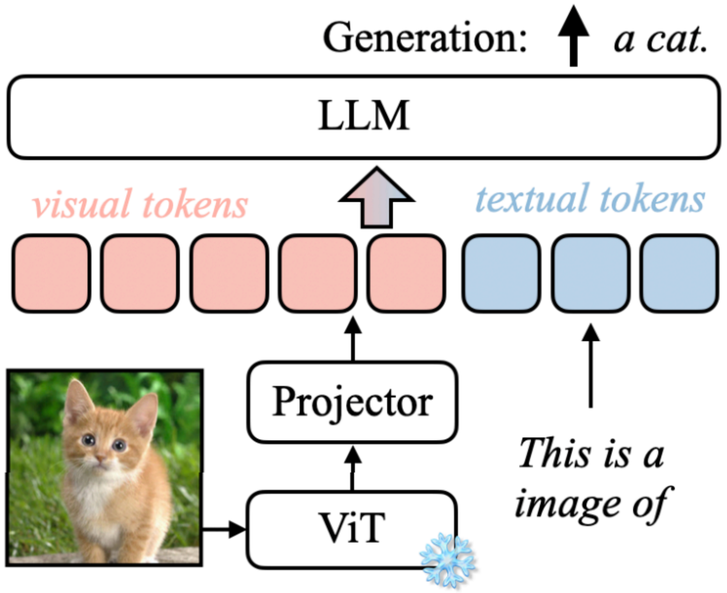
- Dosovitskiy et al. 2021 —> “visual transformer”
  - Not too different from conventional language transformers + **PATCHIFY**
  - **Patchification:** Convert “continous visual signals” (Images/Videos/etc) into “discrete spatial patches”



Next Era: Vision Language Models

Key Idea: Leverage LLMs + Visual Encoder

- Currently 3 main VLMs:
  - Qwen2-VL, VILA, LLaVA
- Three components:
  - Visual Encoder (CLIP)
  - Projector (FFN/DB)
  - LLM (Llama, Vicuna, etc.)



Fundamental of SSMs

Key Idea: Efficient Inference + Very Long Context

- *Introducing S4 -- A classical state space model for deep learning.*
- **Stability** Derived from State Space Control
- **Convolutional** model for training—an **RNN** for inference.
- **Long** convolutional models (filter == input size)
- **Fast**. Asymptotically more efficient than transformers

Input: (discrete) signal of dimension d at N time steps.

Output: (discrete) signal of dimension d at N time steps.

What does it provide?

Key Idea: LTI-based Convolution + constrained learnable parameters

- Continuous Time ODE

$$\frac{d}{dt}x(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

u(t) is the **input**
y(t) is the **output**
x(t) is the **hidden state**

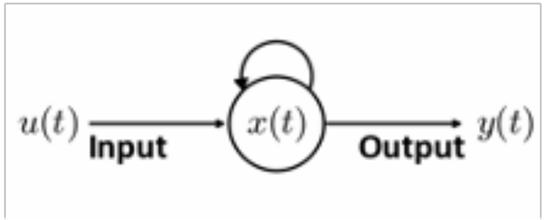
$$y(t), u(t) \in R^1,$$
$$x(t) \in R^d,$$
$$A \in R^{d\times d}, B \in R^{d\times 1}, C \in R^{1\times d}$$

LTI to start with **Convolutional Closed-form**

$$x(s) = \int_0^s e^{A(s-t)} Bu(t)\, dt$$

Routh Criterion for **OLHP Stability**

In AI/ML we can learn A & B (learnable parameters), hence we enforce the constraint (negative real-part eigenvalues of A) !



How do we discretize?

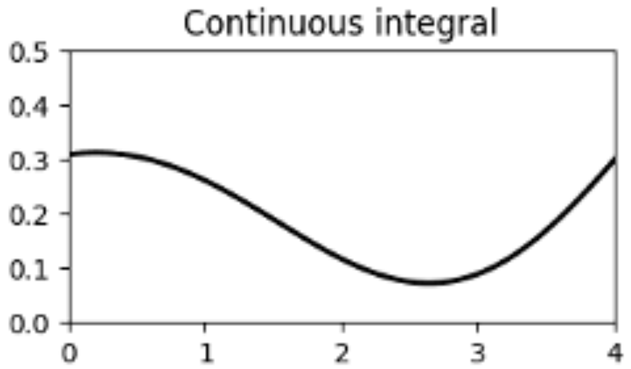
Informational (has not been heavily explored)

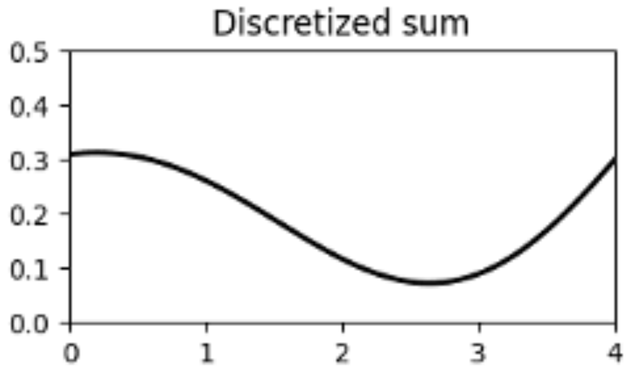
Exact discretization? – Intractable for **Heavy Matrix (exp/integrals)**

How should we discretize the system?

- **Euler**
- **Trapezoidal**
- **ZOH**

Each method will exhibit a different degree of information loss – but Neural Networks can be Robust!





$$x(s) = \int_0^s e^{A(s-t)} Bu(t) dt$$

$$x[n+1] = T \sum_{k=0}^{n} e^{AT(n-k)} Bu[k]$$

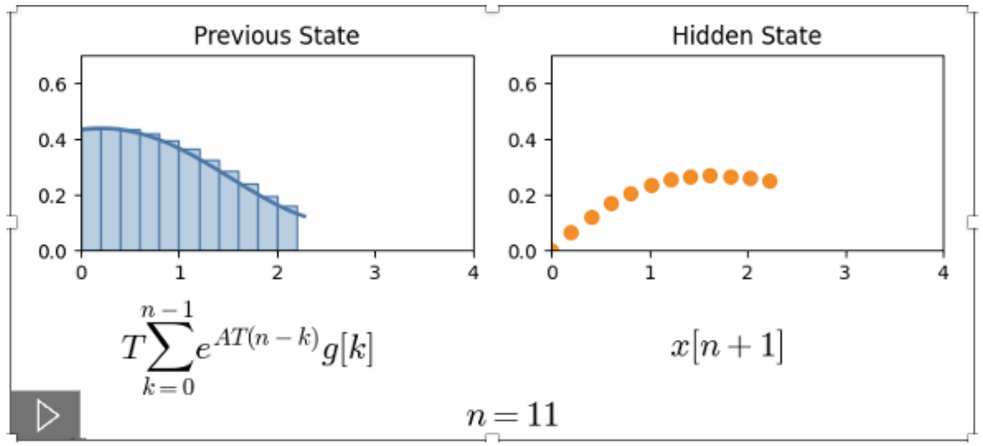
2 Interpretations of SSM

(Left) = Fast Inference — (Right) = Long Context Train

- Fast Recurrent Inferencing

$$x[n+1] = T\sum_{k=0}^{n} e^{AT(n-k)} g[k] \quad g[k] = Bu[k]$$

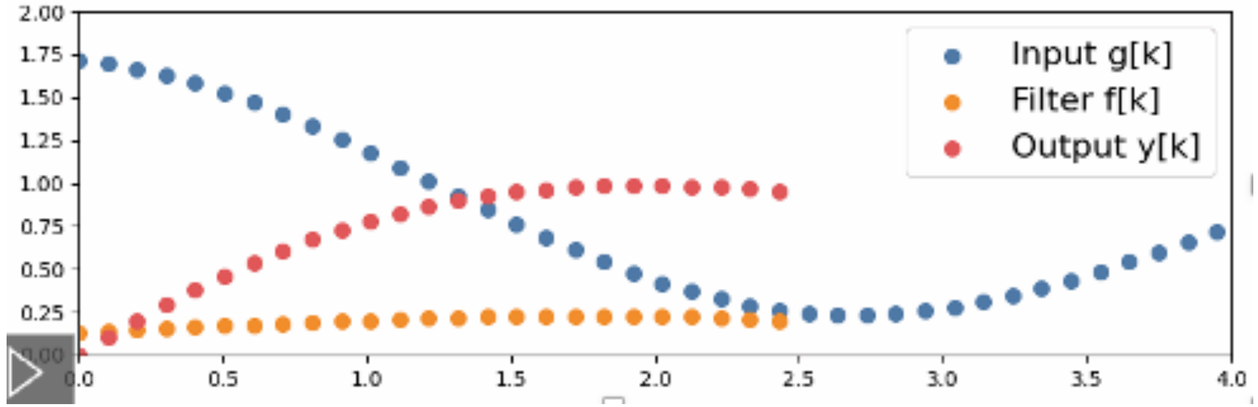
$$x[n+1] = Tg[n] + Te^{AT}\sum_{k=0}^{n-1} e^{AT(n-1-k)} g[k] = Tg[n] + e^{AT}x[n]$$



- *Long* Discrete Convolutions

$$f[k] = e^{ATk}$$

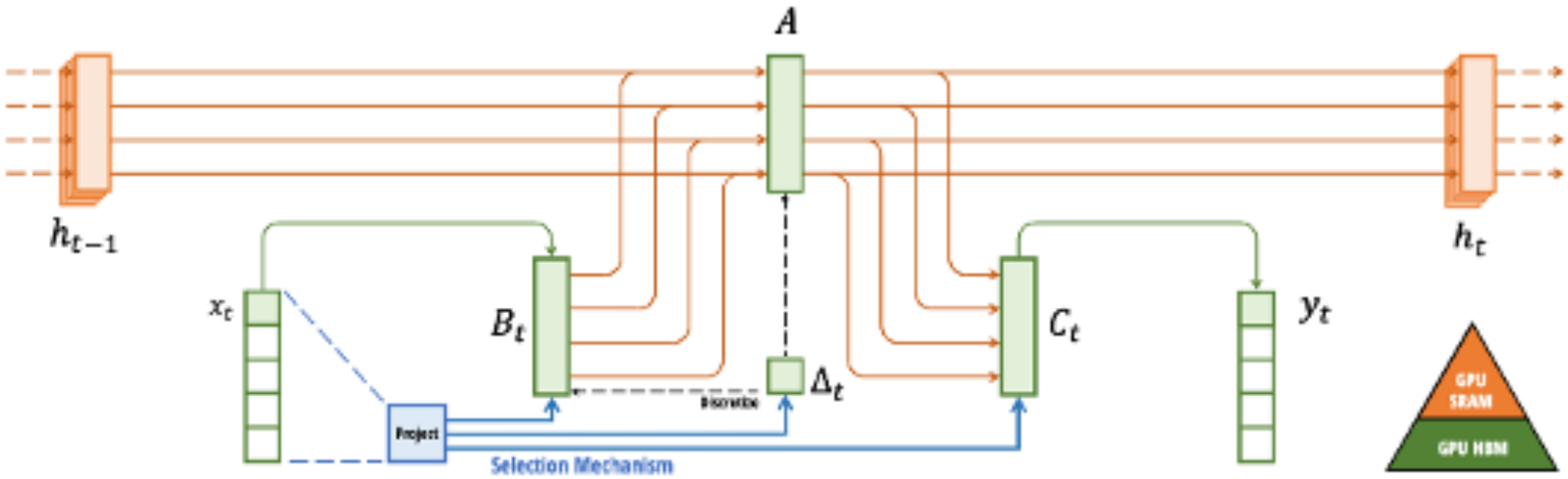
$$x[n+1] = T\sum_k f[n-k]g[k] = T(f * g)[n]$$



Introducing Mamba

Key Idea: Theoertical Efficiency + Practical Implementation

- **Speed & Stability**
- For matrix exp A: use **Imposed Characteristics**: Simple structure, diagonal matrices
- For **long convolutions**: don't materialize the full quadratic hidden state, expressive monarch matrices and unrolling convs into FFTs
- Despite **improved** efficiency, there's **still a gap** to the baseline architectures – attention free networks were still underperforming – until Mamba (~~**true sub-quadratic model**~~)



Why Mamba?

Selectivity == Compression

- **Built on S4 (Convolutional / Recurrent Properties)**
  - + input - dependent properties —> (time-varying systems) based on a time-vs-sequence viewpoint
- **Selective Scanning (focus) —> as a means of compression**
  - + Adapting to the input sequence increases network capacity & complexity
  - + Selectively scanning (kernel fusion/recomputation/etc.) decrease costs
- **Contributions**
  - 1. Selective SSM (conditional processing based on input)
  - 2. Simplicity = Replace Transformer (MLP + Quadratic Attention) with SSM
  - 3. Hardware Aware Parallelism —> Recurrent execution + increased utilization over the memory hierarchy

Assignment Outline (2/4 Parts)

Part 1 & 2: SSM

- 1. State Space Models:
  - Read the Mamba paper and write up to a 10 page document (1 column format excluding figures/ references) summarizing **(60% of the grade)**:
    - Key Points, Techincal Contributions, Areas for improvement
    - + 2 ways it can be extended or made more efficient (each way must include) (3 if you do in a group):
      - How you would extend it
      - Diagrams of your framework
      - Tasks it would solve, etc
      - Think of this as a paper outline which includes the idea, related works, some resemblance of method, and experimental setup (however you do not need to do these experiments)
      - For **one** of these ways conduct some reasonable experiments and include results/analysis

Assignment Outline (2/4 Parts)

Part 3 & 4: VLM

- 1. Vision Language Models:
  - Pick one of the three main works (Qwen2VL, VILA, LLaVA) (I suggest VILA or LLaVA): **(40% of the grade)**:
    - Read and Summarize the main contributions (similar to the SSM part)
    - Discuss the efficiency bottlenecks in said architectures and propose a new approach:
      - This can be anything from kernel fusion to token slicing, etc..
      - For your selected approach, include a method, and some rudiementary experiments
      - Since VLM's require high GPU usage, we are relaxing the requirment for accuracy to be reported, rather just FLOPS or latency, which can be tested out of place, hence removing the requirment of sophisticated GPU setups.
      - For example, if you propose a new attention mechanism, you can profile this with a series of tensor matmul operators on Google Collab.

Project B Deliverables

SSM + VLM Deliverables

- PDF 1:
  - Up to 10 Pages: Details on Mamba + 2 ways to extend + experiments for one of those ways
- PDF 2:
  - No page limit, include proposed method and some efficiency experiments