State-Space Models vs Transformers for Ultra Low-Power Edge AI

Tony Lewis

Chief Technology Officer

Jon Tapson

Chief Development Officer

BrainChip, Inc.

brainchip™

About BrainChip – Founded 2013

Design & license machine learning accelerators for ultra low-power AI

Business Model: IP Licensing

15+ years of AI architecture research & technologies

65+ data science, hardware & software engineers

Publicly traded Australian Stock Exchange (BRN:ASX)

Customers

onsor, MegaChips, Mercedes-Benz, NASA, FRONTGRADE, esa European Space Agency, RENESAS, Chelpis, Valeo, AFRL

Goals of This Presentation

- Analysis of computation and bandwidth in state-space models and transformers
- Establish energy and costs savings measures that are available ONLY to SSM
  - Efficient off-line processing of context information
  - Use of read only memory, e.g., flash to dramatically reduce power
- Conclusion

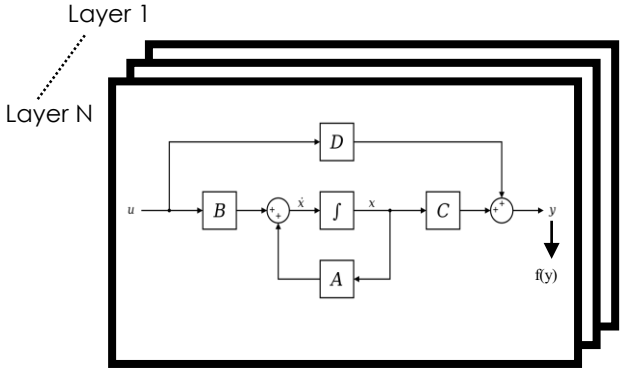
Problem Statement

- Goal:
  - Achieve < 0.5 W system power
  - Sub-100 ms latency for edge AI such as RAG
  - Low SRAM < 1 MB and SoTA performance
- Why? Unlocks new cost and power sensitive markets
- Key challenges:
  - Transformer based LLMs dominate today
  - Transformers KV cache expands blows up chip cache
  - RAG uses long context length (> 1024 tokens)
- Opportunity:
  - State-Space Models address power, size issues

State-Space Model Overview (1/2)

- State-Space refers to time-domain model of coupled linear difference equations used to model physical systems.
  - **x** is the state of the system
  - **u** is the input. For LLM, a real vector 1k-4k long
  - **A** is a diagonal matrix. It is stable, acts as a low pass filter, with oscillations. **x**=A**x** are decoupled filters
  - **Bu** drives this filter. B is a mixing term
  - **C** reads out the state
  - This part is a generic State-Space Model

State-Space Model



$$\mathbf{x}_{k+1}^{(\ell)} = \mathbf{A}^{(\ell)}\mathbf{x}_{k}^{(\ell)} + \mathbf{B}^{(\ell)}\mathbf{u}_{k}^{(\ell)}$$

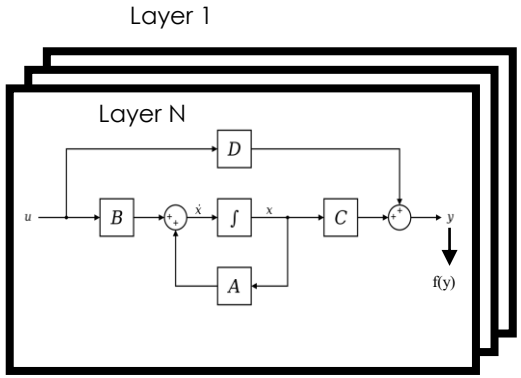
$$\mathbf{y}_{k+1}^{(\ell)} = \mathbf{C}^{(\ell)}\mathbf{x}_{k+1}^{(\ell)} + \mathbf{D}^{(\ell)}\mathbf{u}_{k}^{(\ell)}$$

$$\mathbf{u}_{k+1}^{(\ell+1)} = F\left(\mathbf{y}_{k+1}^{(\ell)}\right)$$

State-Space Model Overview (2/2)

- Innovation in SSMs: **F** is a non-linearity, e.g., SiLU, ReLU
- Relation to neural networks
  - **B** matrix is a set of input weights to individual neurons
  - **A** gives the neurons dynamics, similar to RNNs
- Difference with RNNs
  - Because of the regular structure, this RNN can be converted to CNN for fast training on GPUs!
  - Recurrent inference, small, efficient

State-Space Model



$$\mathbf{x}_{k+1}^{(\ell)} = \mathbf{A}^{(\ell)}\mathbf{x}_{k}^{(\ell)} + \mathbf{B}^{(\ell)}\mathbf{u}_{k}^{(\ell)}$$

$$\mathbf{y}_{k+1}^{(\ell)} = \mathbf{C}^{(\ell)}\mathbf{x}_{k+1}^{(\ell)} + \mathbf{D}^{(\ell)}\mathbf{u}_{k}^{(\ell)}$$

$$\mathbf{u}_{k+1}^{(\ell+1)} = F\left(\mathbf{y}_{k+1}^{(\ell)}\right)$$

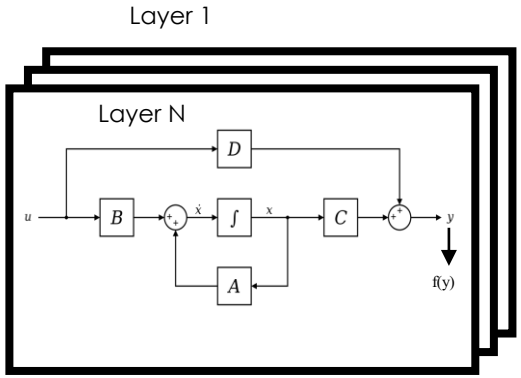
Training

- Can train SSM as convolutional networks exploiting parallelism in GPUs
- Distillation pathway from transformers to state space model
  - Distillation is a popular way of training smaller LLMs
  - Start with large LLM an use it as a teacher for the small SMM.
  - Cross-architecture distillations for transformers -> State-Space Models are being developed (Mohawk)

State-Space Model Cache Is Tiny

- Memory requirements, BrainChip TENNs LLM 1B parameters
- State size includes:
  - States per layer: 4K
  - Word size : 2 Bytes
  - N Layers : 24
- Calculation for 1B parameter model:
  - State size = 4K*2*24* = 393 KB

State-Space Model



$$\mathbf{x}_{k+1}^{(\ell)} = \mathbf{A}^{(\ell)}\mathbf{x}_{k}^{(\ell)} + \mathbf{B}^{(\ell)}\mathbf{u}_{k}^{(\ell)}$$

$$\mathbf{y}_{k+1}^{(\ell)} = \mathbf{C}^{(\ell)}\mathbf{x}_{k+1}^{(\ell)} + \mathbf{D}^{(\ell)}\mathbf{u}_{k}^{(\ell)}$$

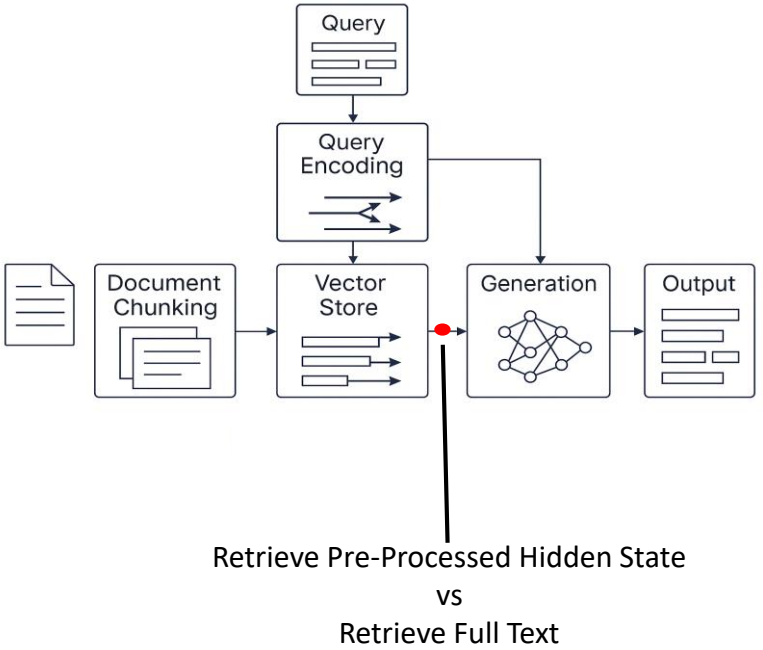
$$\mathbf{u}_{k+1}^{(\ell+1)} = F\left(\mathbf{y}_{k+1}^{(\ell)}\right)$$

State-Space Machine Are Markov

- Given an SSM, the current state is conditionally independent of the past:

$$p(x_t \mid x_{0:t-1}) \;=\; p(x_t \mid x_{t-1}).$$

- Implications for Retrieval Augmented Generation
- “Chunks” of text are retrieved for processing.
  - With SSMs, preprocess the entire chunk and store in hidden state
  - Can then “seed” state-machine.
  - Computation cost is ~0 for any context length size!



State-Space Models: Hardware Benefit

- Memory transfers can be Read Only
  - DDR is not needed. Minimum DDR confirmation are ~2 watts and above
  - Flash brings us below < 0.5 watts active. Plus no leakage
- Compute is constant
  - At 20 tokens/sec, for a 1B model requires ~ 20 GMACs. Using 1 pJ/MAC, energy for LLM is ~20 milliwatts.
  - Bandwidth < 5-10 GB/Sec,
  - Time to first token < 100 ms for RAG do to caching

How Do Transformers Compare?

- Memory
  - In a 1B model, i.e., Llama 3.2 1B must cache Key and Value terms for all layers.
  - 1K tokens requires an overwhelming 50 MB of cache; 50 MB > 1 MB
    Compute: Attention head grows in compute and size a N^2.
  - Memory bandwidth: Must cache KV on DRAM. IO bandwidth begins to be dominated by KV read/writes for long context.
  - DDR means higher minimum floor; > 2 watts
- Compute of 1K input context requires trillions of macs.
  - High compute means large energy costs.

Transformers versus SSM

Aspect	Transformers	SSM
Research Activity	Intense optimization efforts	Growing interest. Fewer optimization techniques
Lossy Compression	Full context retention	Hidden state acts as lossy bottleneck
Computational Complexity	O(N^2) (very poor)	O(N)
Inference Speed	Slower	Much faster
Die Area (cost)	Very high	Very low
Flash Compatible vs. DRAM	No	Yes
Precompute Offline	No	Yes

BrainChip TENNs 1B versus Transformer 1B

	TENNs 1B	Llama 3.2 1B	Comment
Perplexity, lower is better	6.3	13.7 (base)	SMM shows strong possibilities for RAG applications
Teraflops 1024 context tokens (RAG application)	0	2.5	Offline compute is great benefit
Teraflops additional 100 query tokens	0.1	0.25	
MMLU	40	49	Transformers excel for certain tasks
Write bandwidth KV cache	0	156 MB	➔ Large on-chip memory or external DRAM
Read bandwidth KV cache	0	95 GB	Latency reduced w/ slow mem

Summary

- State-Space Models are a viable alternative for LLM at the extreme edge
  - SMM requires small cache, read-only memory at low bandwidth and low compute intensity.
  - Total power for a 1B design comes in under 0.5 watts for both Flash access and compute
- Transformers cannot meet ultra-low power requirements today, due to the following:
  - Transformers require large cache, read-write memory and off board DDR
  - Transformers require high compute with many TOPS with many mac units, driving up cost, power, and heat generation

What Are the Drawbacks of State-Space Models?

- Our models have better performance on metrics like perplexity versus public domain Transformer models.
- Yet, some tasks like copying, and in context learning remain difficult for SSM.
- Transformers are the subject of intense research, with new efficiencies every day.
- The main strength and weakness of SSM is their Markov property.

Resources

- State-Space Models: https://en.wikipedia.org/wiki/State-space_representation
- Mohawk: https://goombalab.github.io/blog/2024/distillation-part1-mohawk/
- Transformer Compute Requirements: Kaplan et al., https://arxiv.org/pdf/2001.08361

Thank You

See our demonstration in booth # 716