FlashRNN: I/O Aware Optimization of Traditional RNNs on Modern Hardware

Korbinian Pöppel, Maximilian Beck, Sepp Hochreiter

Johannes Kepler University Linz, NXAI Lab, NXAI GmbH

JOHANNES KEPLER
UNIVERSITY LINZ

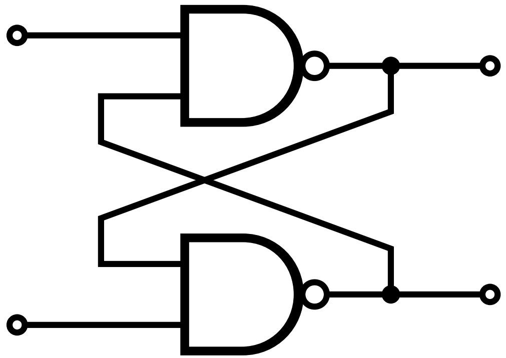
Why Traditional RNNs?

State-Tracking Tasks cannot be solved by Transformers[2] and current SSMs (Mamba[3]) [1]

Example: Parity Task

- Given a sequence of zeros and ones, add a zero or one, such that the total number of ones is even.
- Solved by a simple flip-flop, 1-bit finite-state automaton
- Cannot be solved by Transformer or Mamba in extrapolation
- Parity is only the simplest form of a State-Tracking Task [1]

0110101110
1010100111



BUT: Can be solved by Traditional RNNs

JOHANNES KEPLER
UNIVERSITY LINZ

Traditional RNNs

$$\text{gates}: \quad \left(\mathbf{g}_t^{(j)}\right)_{j\in\{1..N_g\}} \qquad \text{states}: \quad \left(\mathbf{s}_t^{(i)}\right)_{i\in\{1..N_s\}}$$

$$\text{gate input}: \quad \left(\mathbf{x}_t^{(j)}\right)_{j\in\{1..N_g\}} \qquad \text{recurrent matrix}: \quad \left(\mathbf{R}^{(j)}\right)_{j\in\{1..N_g\}}$$

$$\text{gate bias}: \quad \left(\mathbf{b}^{(j)}\right)_{j\in\{1..N_g\}} \qquad \text{element-wise nonlinearity}: \quad \mathcal{P}^{(i)}\left(\cdot,\cdot\right)$$

MatMul

$$\mathbf{g}_t^{(j)} = \mathbf{x}_t^{(j)} + \mathbf{R}^{(j)}\mathbf{s}_{t-1}^{(0)} + \mathbf{b}^{(j)} \tag{1}$$

Elem-Wise

$$\mathbf{s}_t^{(i)} = \mathcal{P}^{(i)}\left(\left(\mathbf{s}_{t-1}^{(i')}\right)_{i'\in\{1..N_s\}}, \left(\mathbf{g}_t^{(j)}\right)_{j\in\{1..N_g\}}\right) \tag{2}$$

repeated for $t \in \{1..T\}$ time steps

Traditional RNNs

- LSTM[5] – 4 gates, 2 states

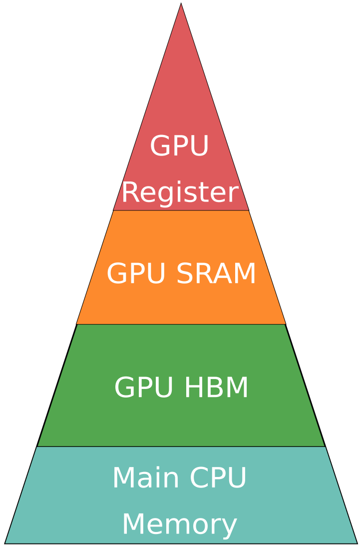
$$\text{states:} \quad \mathbf{y}_t, \mathbf{c}_t \qquad \text{gates:} \quad \mathbf{i}_t, \mathbf{f}_t, \mathbf{z}_t, \mathbf{o}_t$$

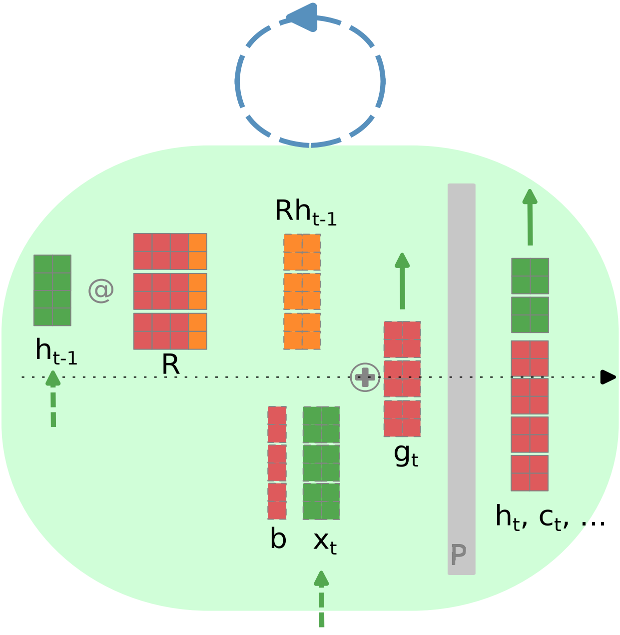
$$\mathbf{c}_t = \sigma\left(\mathbf{f_t}\right) \cdot \mathbf{c}_{t-1} + \sigma\left(\mathbf{i}_t\right) \cdot \tanh\left(\mathbf{z}_t\right)$$

$$\mathbf{y}_t = \sigma\left(\mathbf{o_t}\right) \cdot \tanh\left(\mathbf{c}_t\right)$$

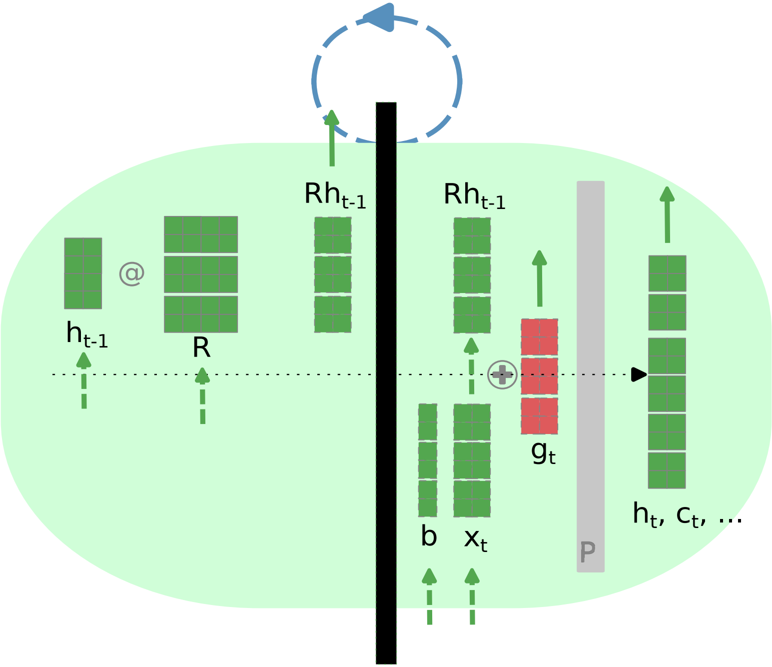
- Elman Networks[4] – 1 gate, 1 state
- GRU[6] – 3 gates, 2 states
- sLSTM[7] – 4 gates, 4 states
- ...

Speed Optimization





Fused



Alternating

Sequence Loop

Write to HBM

Read from HBM

Pointwise Non-Linearity

Matrix Multiplication

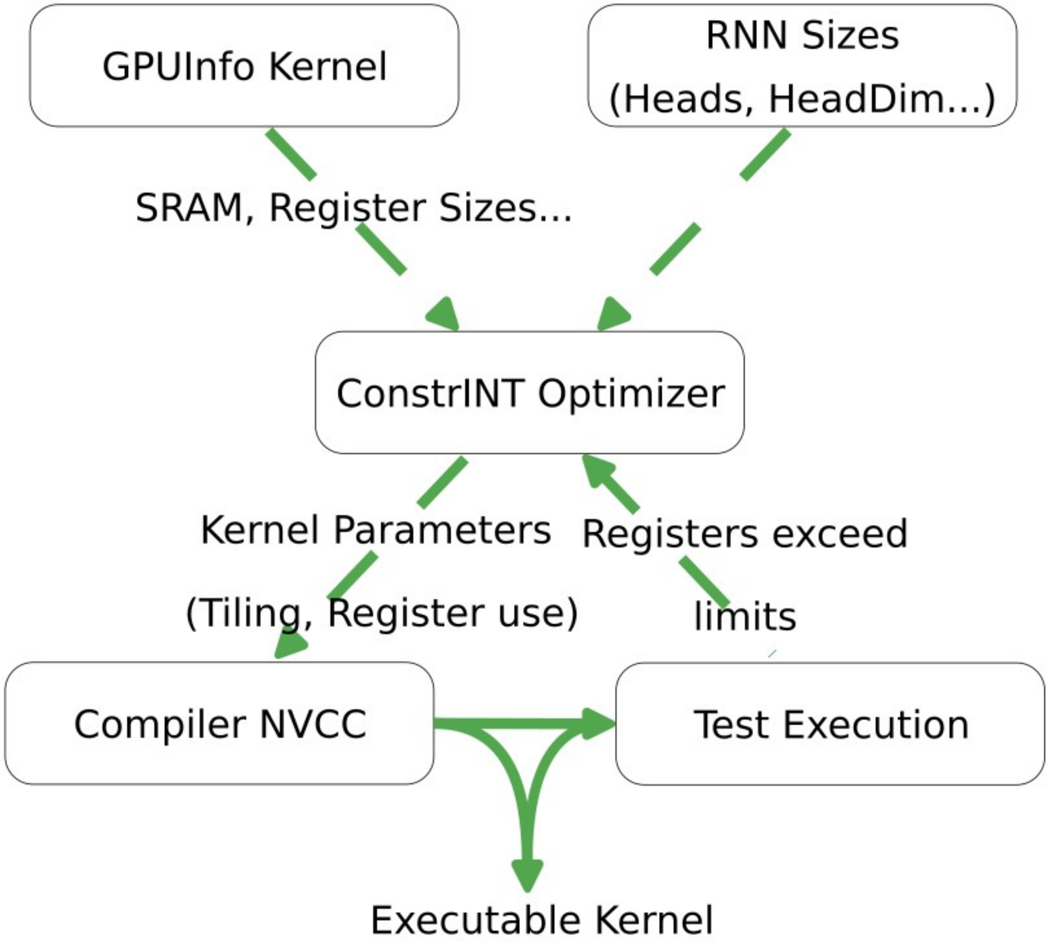
Pointwise Addition

Kernel Boundary

JOHANNES KEPLER
UNIVERSITY LINZ

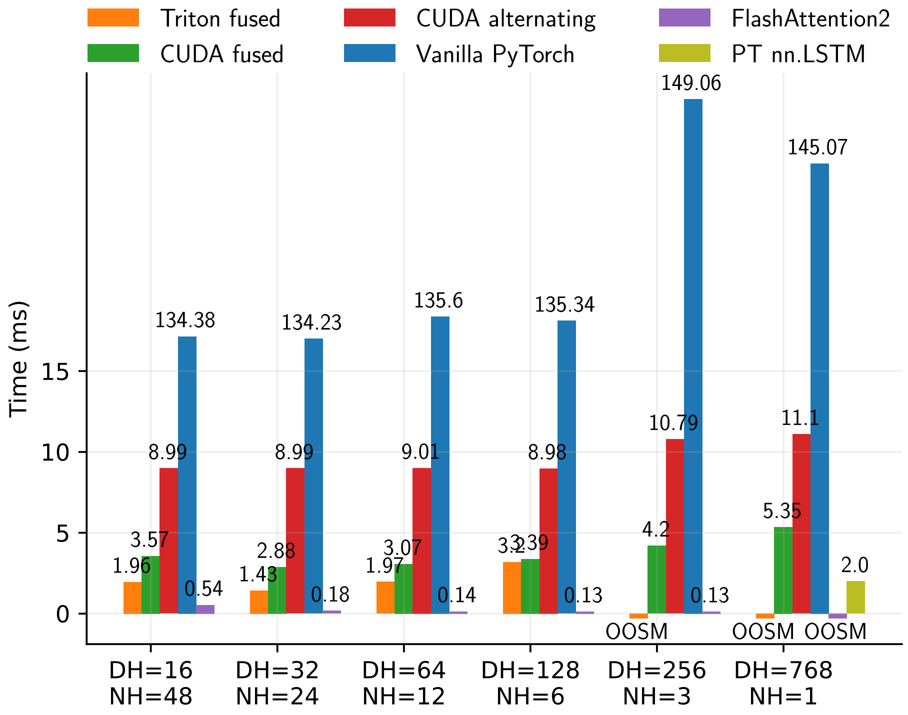
Hardware Adaption: SRAM... sizes

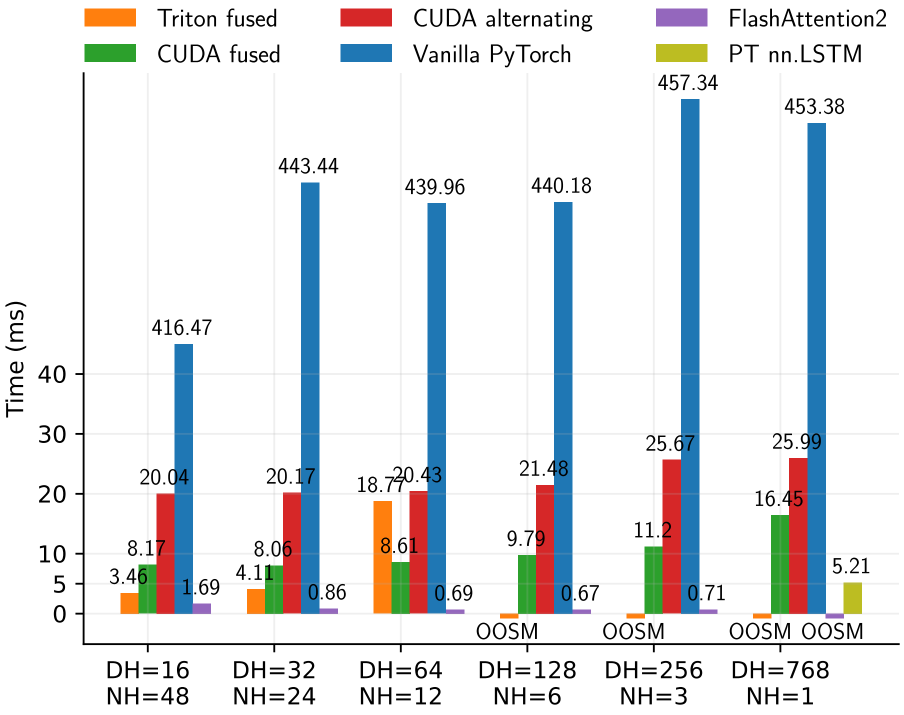
- Hardware Constraints as Integer CSP
  (12 variables – 15 constraints)
- Solve with ConstrINT



JOHANNES KEPLER
UNIVERSITY LINZ

Speed Results





JOHANNES KEPLER UNIVERSITY LINZ

30x – 140x speed up over vanilla PyTorch impl.
a bit slower than closed source cuDNN LSTM

Language Modeling Test and Parity

Language Modeling on SlimPajama[8], 160M parameters: FlashRNN-based RNN models with Transformer backbone are just 2x slower than Transformers

Model	Heads	Param. (M)	Train Time (h)	Median Step (s)	Val PPL (val)
LSTM CUDA fused	1	190	9.9	0.535	22.1
LSTM CUDA altern.	1	190	10.8	0.575	21.9
LSTM PT `nn.LSTM`	1	190	4.5	0.285	25.8
LSTM CUDA fused	12	164	5.9	0.325	22.2
LSTM CUDA altern.	12	164	9.6	0.511	22.1
sLSTM CUDA fused	1	190	10.1	0.543	21.3
sLSTM CUDA altern.	1	190	10.9	0.577	21.4
sLSTM CUDA fused	12	164	6.8	0.342	21.7
sLSTM CUDA altern.	12	164	9.7	0.509	21.8
Transformer	12	162	2.9	0.190	17.9

Parity Extrapolated Validation Results: RNNs can do state tracking

Model	Transformer	Mamba	mLSTM	Elman	GRU	LSTM	sLSTM
Acc (Ext.)	0.52	0.56	0.54	1.00	1.00	1.00	1.00

Conclusion

- Traditional RNNs can be largely accelerated on modern GPUs
- Not as fast in training as parallelizable Transformers
- Valuable for Tasks that need State-Tracking Capabilities

JOHANNES KEPLER
UNIVERSITY LINZ

Bibliography

[1] Merrill, William, Jackson Petty, and Ashish Sabharwal. “The Illusion of State in State-Space Models.” In Forty-First International Conference on Machine Learning, 2024. https://openreview.net/forum?id=QZgo9JZpLq.

[2] Gu, Albert, and Tri Dao. “Mamba: Linear-Time Sequence Modeling with Selective State Spaces.” arXiv, December 1, 2023. http://arxiv.org/abs/2312.00752.

[3] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need.” arXiv:1706.03762 [Cs], December 5, 2017. http://arxiv.org/abs/1706.03762.

[4] Elman, Jeffrey L. “Finding Structure in Time.” Cognitive Science 14, no. 2 (March 1990): 179–211. https://doi.org/10.1207/s15516709cog1402_1.

[5] Hochreiter, Sepp, and Jürgen Schmidhuber. “Long Short-Term Memory.” Neural Computation 9, no. 8 (November 1, 1997): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[6] Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation.” In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1724–34. Doha, Qatar: Association for Computational Linguistics, 2014. https://doi.org/10.3115/v1/D14-1179.

[7] Beck, Maximilian, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael K. Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. “xLSTM: Extended Long Short-Term Memory.” In The Thirty-Eighth Annual Conference on Neural Information Processing Systems, 2024. https://openreview.net/forum?id=ARAxPPIAhq.

[8] D. Soboleva, F. Al-Khateeb, R. Myers, J. R. Steeves, J. Hestness, and N. Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/ slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

JOHANNES KEPLER UNIVERSITY LINZ