Blending Complementary Memory Systems in Hybrid Quadratic-Linear Transformers

Kazuki Irie[1] Morris Yau[2] Samuel J. Gershman[1,3]

[1]Department of Psychology & Center for Brain Science, Harvard University, USA [2]MIT CSAIL, USA [3]Kempner Institute for the Study of Natural and Artificial Intelligence, USA

Background

We have **two types** of transformers ("memory mechanisms") with **complementary strengths**:

- **softmax/quadratic attention** = key-value memory
- **linear attention / fast weight programming** = fast weight memory

Property	Key-value memory	Fast weight memory
Complexity	quadratic	**linear**
Context length	bounded	**unbounded**
Retrieval precision	**high**	low
Expressivity	low	**high**

How to get the best of both worlds?

Ideal solution (dream!): devise a clean/principled single memory mechanism with all these nice properties!

Engineer's solution (this work): Combine existing solutions! = build a hybrid system.

Remark: somewhat reminiscent of a **neuroscience hypothesis** (**Complementary Learning Systems**; McClelland et al. 1995): the brain achieves two memory functions that are **normally incompatible** (remembering specifics vs. extracting generalities) through a **division of labor** between separate episodic and semantic memory systems.

There have been many recent works on hybrid sequence models combining transformer and RNN-like models: layer-wise (e.g., xLSTM), or intra-layer (e.g., BASED, Infini-attention) hybrid approaches.

→ Our focus is **intra-layer hybrids of sliding window attention and DeltaNet**. Since our hybrid combines two types of "transformers", Query/Key/Value projections/variables can be **nicely shared between the two sub-systems.**

Quick reminder about DeltaNet (ICML 2021)

$$[\boldsymbol{q}_t, \boldsymbol{k}_t, \boldsymbol{v}_t, \beta_t] = \boldsymbol{W}_{\text{slow}}\boldsymbol{x}_t$$
$$\boldsymbol{W}_t = \boldsymbol{W}_{t-1} + \sigma(\beta_t)(\boldsymbol{v}_t - \boldsymbol{W}_{t-1}\phi(\boldsymbol{k}_t)) \otimes \phi(\boldsymbol{k}_t)$$
$$\boldsymbol{y}_t = \boldsymbol{W}_t\phi(\boldsymbol{q}_t)$$

- More **expressive** than quadratic attention (Grazzi+ 2024)
- while still supporting **parallel training** (Yang+ 2024)

Summary

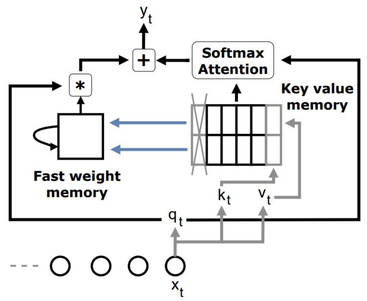
How to build intra-layer hybrid quadratic-linear attention models while leveraging complementary strengths of their components?

- We examine design choices for passing information between the two sub-systems, including the timing of information flow and the method used to combine their outputs (i.e.., gating).
- We advocate for expressivity as an important, yet often overlooked, aspect of hybrid designs. Consequently, the "time-synchronous" hybrid approach performs best.

HQLT Model Designs

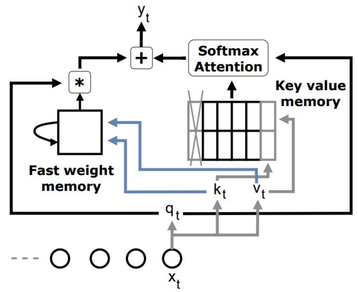
Different division of labor between the two sub-systems

Delayed-Stream HQLT



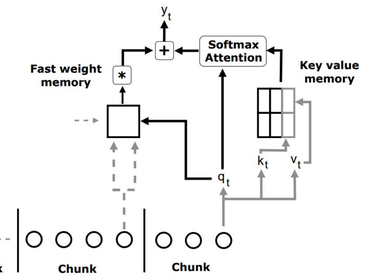
Old key/value vectors falling outside of the sliding window are fed to FW-memory (*clean division of labors*)

Synchronous HQLT



Key/value vectors are synchronously fed to both KV-memory and FW-memory (*as we'll see this is crucial to leverage the full advantage of FW-memory*)

Delayed-Chunk HQLT



Delayed variant with a "chunk size": KV-memory contents are flashed at the beginning of each chunk (*natural from the chunk-wise training view; just adding softmax within chunk*)

General language modelling tasks

Sanity checks!

- Performance is mostly comparable between different HQLT variants (slight advantage for Delayed-Stream for the 1.3B model)
- Enlarging the window size also has a limited impact for these general LM tasks
- Use of vanilla linear attention instead of DeltaNet yields significant performance degradation

Remark: Our code is based on "**flame**" for training and "**fla**" for model implementation.

github.com/kazuki-irie/hybrid-memory

Model	Wiki. ppl ↓	LMB. ppl ↓	LMB. acc ↑	PIQA acc ↑	Hella. acc_n ↑	Wino. acc ↑	ARC-e acc ↑	ARC-c acc_n ↑	Avg.
340M params									
Transformer++	26.5	34.9	**33.9**	67.6	41.0	53.7	60.2	29.0	47.6
DeltaNet	27.6	35.0	32.8	67.1	40.8	52.6	58.5	28.8	46.8
HQLT									
Delayed-Stream	26.4	33.1	33.6	**67.9**	42.1	51.8	59.4	29.5	47.4
Delayed-Chunk	26.7	29.9	33.5	66.8	42.3	50.9	61.1	**30.6**	47.5
Synchronous	**26.3**	**29.4**	33.3	66.2	**42.7**	**53.8**	**61.5**	29.4	**47.8**
1.3B params									
Transformer++	19.8	17.9	42.6	71.0	50.3	55.8	65.2	33.2	53.0
DeltaNet	20.6	19.9	39.3	70.1	49.5	52.5	68.5	34.2	52.3
HQLT									
Delayed-Stream	20.0	16.5	**43.5**	70.7	**51.6**	56.0	**69.3**	**36.0**	**54.5**
Delayed-Chunk	20.2	16.3	41.3	71.8	50.9	55.0	67.9	35.2	53.7
Synchronous	**19.8**	**15.9**	42.8	**72.0**	51.5	**56.1**	68.1	33.1	53.9
Ablations with *340M params*									
HQLT Synchronous									
sum mixing	27.7	34.8	32.5	67.0	41.2	52.4	60.9	28.9	47.2
dynamic vector mixing (25M)	26.3	29.4	33.3	66.2	42.7	53.8	61.5	29.4	47.8
w. Linear Attn. (no DeltaNet)	33.3	114.2	21.0	63.2	36.4	51.7	53.8	26.8	42.2
window ×2 = 128	26.6	27.8	35.6	68.1	41.7	51.2	61.4	30.1	48.0
window ×4 = 256	26.7	27.7	35.4	67.0	42.2	52.7	59.8	29.0	47.7
window ×8 = 512	27.0	28.0	35.9	66.3	41.3	53.2	60.1	29.0	47.6

Synthetic algorithmic tasks (expressivity evaluation)

Expectably, only the "Synchronous" variant can leverage the expressivity advantage of DeltaNet

Model	Parity acc ↑	Mod Arith acc ↑
Transformer [7]	2.2	3.1
Mamba [7]	100.0	24.1
DeltaNet [7]	100.0	97.1
HQLT		
Delayed-Stream	3.3	27.8
Delayed-Chunk	2.8	1.4
Synchronous	**100.0**	**97.0**
w. Linear Attn.	2.5	44.5

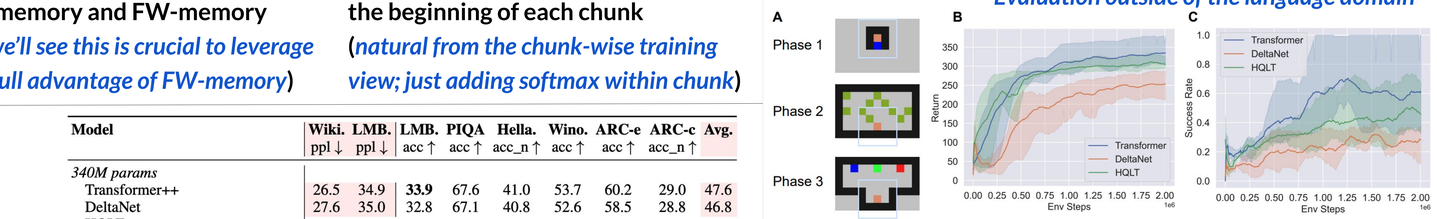
Retrieval tasks

- Large window size is needed for these retrieval intensive tasks
- Naive "sum mixing" is not enough for HQLT (context-dependent scalar or vector gating helps)

	Window size	SWDE acc ↑	SQuAD acc ↑	FDA acc ↑	Avg.
340M params					
Transformer++	2048	**44.9**	36.9	**52.3**	**44.7**
Transformer++	1024	30.4	25.5	31.2	29.0
DeltaNet	-	18.5	25.2	8.6	17.4
HQLT Synchronous					
sum mixer	64	13.3	26.2	12.6	17.4
dynamic scalar	64	21.1	27.7	11.4	20.1
dynamic vector	64	20.0	28.4	10.9	19.8
window ×2	128	16.9	30.5	17.9	21.8
window ×4	256	18.6	35.6	15.1	23.1
window ×8	512	22.9	35.7	17.3	25.3
window ×16	1024	34.0	**37.1**	50.1	40.4

Reinforcement learning in partially observable environment

Evaluation outside of the language domain



Task: there are three phases: (1) observe a color (here blue), (2) collect green apples (distraction task!), (3) reach the right color (if so, counted as success). Partial observation (light blue box) is limited around the agent (brown). Total sequence len = 750

Result: small window size of 64 improves HQLT over DeltaNet

Discussions

- Overall promising results but retrieval performance is not satisfactory (still requires a large sliding window size)
- Designing a more sophisticated communication between the two sub-systems may address this limitation