CAUI 8기 논문 발표

Titans: Learning to Memorize at Test Time

2025.03.18

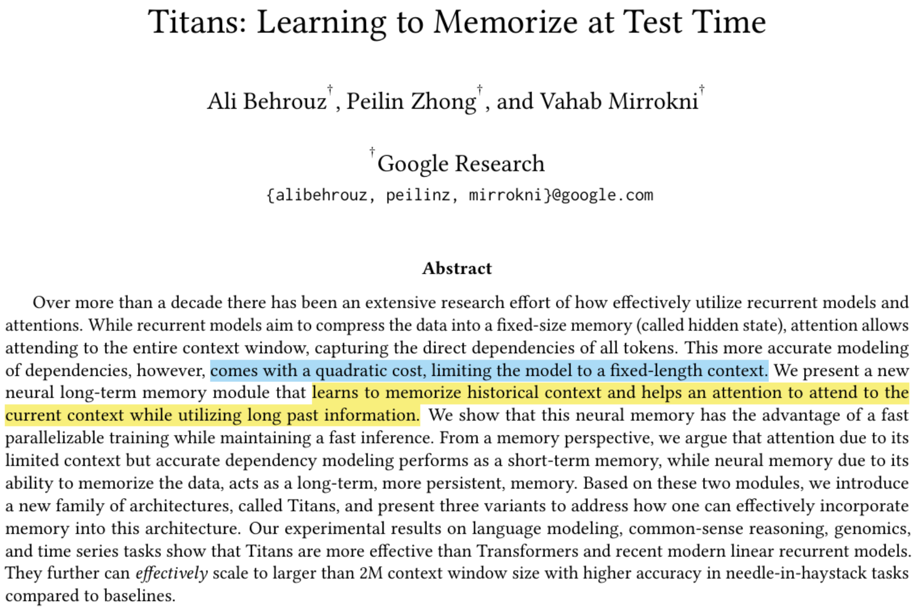
발표자 : 이동하

목차

1. 주제 선정 및 개요

2. 논문 리뷰

주제 선정 및 개요



Titans: Learning to Memorize at Test Time

Ali Behrouz[†], Peilin Zhong[†], and Vahab Mirrokni[†]

[†]Google Research
{alibehrouz, peilinz, mirrokni}@google.com

Abstract

Over more than a decade there has been an extensive research effort of how effectively utilize recurrent models and attentions. While recurrent models aim to compress the data into a fixed-size memory (called hidden state), attention allows attending to the entire context window, capturing the direct dependencies of all tokens. This more accurate modeling of dependencies, however, comes with a quadratic cost, limiting the model to a fixed-length context. We present a new neural long-term memory module that learns to memorize historical context and helps an attention to attend to the current context while utilizing long past information. We show that this neural memory has the advantage of a fast parallelizable training while maintaining a fast inference. From a memory perspective, we argue that attention due to its limited context but accurate dependency modeling performs as a short-term memory, while neural memory due to its ability to memorize the data, acts as a long-term, more persistent, memory. Based on these two modules, we introduce a new family of architectures, called Titans, and present three variants to address how one can effectively incorporate memory into this architecture. Our experimental results on language modeling, common-sense reasoning, genomics, and time series tasks show that Titans are more effective than Transformers and recent modern linear recurrent models. They further can *effectively* scale to larger than 2M context window size with higher accuracy in needle-in-haystack tasks compared to baselines.

주제 선정 및 개요

Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

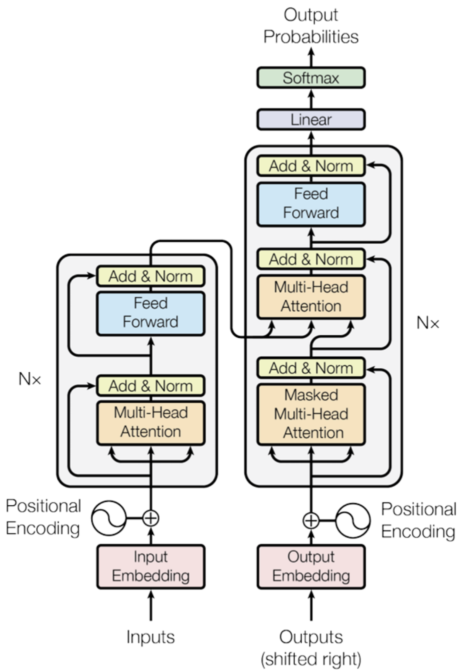
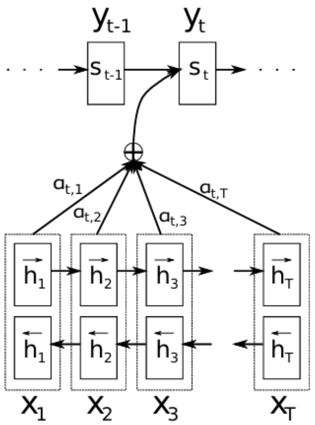


Figure 1: The Transformer - model architecture.

주제 선정 및 개요



y_{t-1}
y_t
s_{t-1}
s_t
$a_{t,1}$
$a_{t,2}$
$a_{t,3}$
$a_{t,T}$
$\overrightarrow{h}_1$ $\overrightarrow{h}_2$ $\overrightarrow{h}_3$ $\overrightarrow{h}_T$
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ $\overleftarrow{h}_3$ $\overleftarrow{h}_T$
X_1 X_2 X_3 X_T

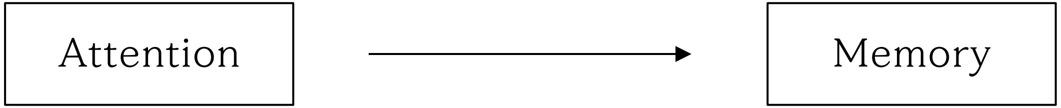
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align and Translate", arXiv, 2015.

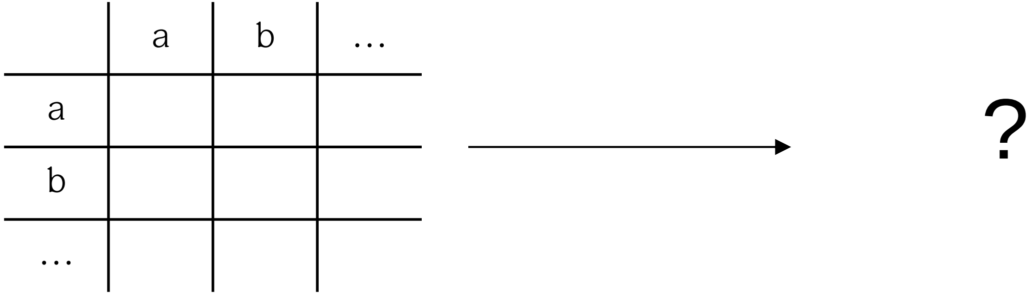
Attention mechanism

주제 선정 및 개요

Titans: Learning to Memorize at Test Time



How to memorize?



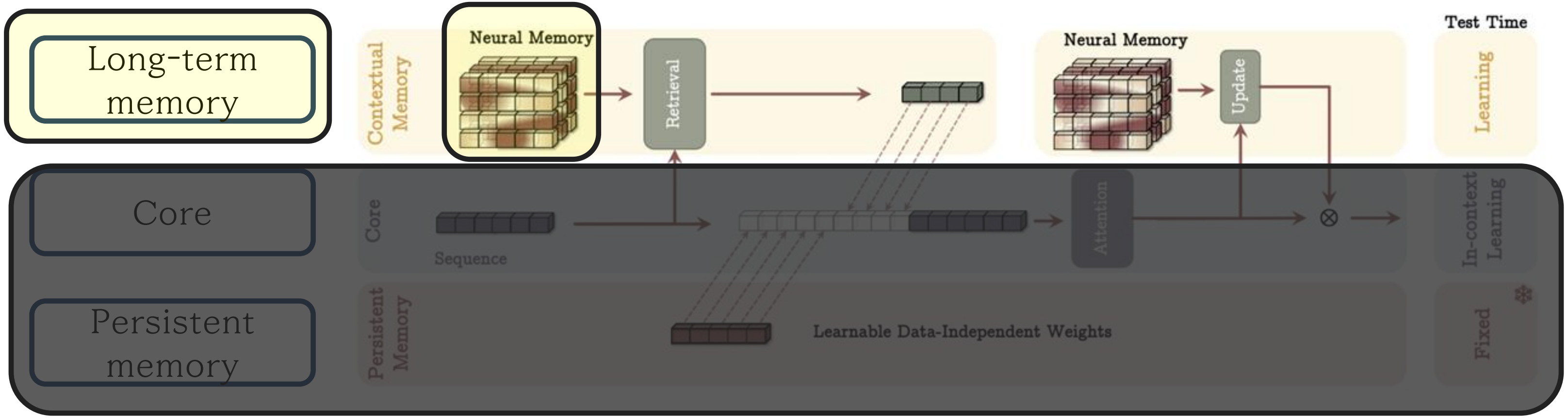
	a	b	…
a			
b			
…			

논문 리뷰

‘기억하는 법’에 대해 성능을 향상 시킨 모델을 만들어보자.

논문 리뷰

Titans: Learning to Memorize at Test Time



Long-term memory

Core

Persistent memory

논문 리뷰

$$\mathcal{M}_t$$

(정보를 입력 받았을 때)놀라는 정도 = 인상 깊은 정도 = 우리의 기억에 남는 정도

논문 리뷰

$$\mathcal{M}_t = \mathcal{M}_{\{t-1\}} - \theta_t \nabla\, l\left(\mathcal{M}_{\{t-1\}};\, x_t\right)$$

⇓

‘놀람’이 누적될 경우 뒤에 오는 정보의 가치가 희석됨.

Past surprise

$$S_t = \eta_t S_{\{t-1\}} - \theta_t \nabla\, l\left(M_{\{t-1\}};\, x_t\right)$$
$$\mathcal{M}_t = \mathcal{M}_{\{t-1\}} + \boxed{S_t,}$$

⇓

얼마나 많은 정보를 잊을까?

$$S_t = \eta_t S_{\{t-1\}} - \theta_t \nabla\, l\left(\mathcal{M}_{\{t-1\}};\, x_t\right)$$
$$\mathcal{M}_t = \boxed{(1 - \alpha_t)}\mathcal{M}_{\{t-1\}} + S_t$$

논문 리뷰

Objective function

$$k_t = x_t W_K, \qquad v_t = x_t W_V$$

$$(where\ W_K, W_V \in \mathbb{R}^{\{d_{\{in\}} \times d_{\{in\}}\}})$$

$$l(\mathcal{M}_{\{t-1\}};\, x_t) = \boxed{\left\| \mathcal{M}_{\{t-1\}}(k_t) - v_t \right\|_2^2}$$

메모리가 key와 value간의
관계를 기억하도록 학습.

Retrieving a memory

$$q\_t = x\_t\, W\_Q$$

메모리에서 기억을 끄집어낼 때

논문 리뷰

$$\mathcal{M}_t = (1 - \alpha_t)\mathcal{M}_{\{t-1\}} - \theta_t \nabla l\left(\mathcal{M}_{\{t-1\}};\ x_t\right) = \beta_t \mathcal{M}_0 - \sum_{i=1} \theta_i \left(\frac{\beta_t}{\beta_i}\right) \nabla l\left(\boxed{\mathcal{M}_{\{t'\}}};\ x_i\right)$$

⇓

$$\nabla l\,(W_0;\ x_t) = (W_0 x_t - x_t)x_t^T$$

Mini batch 단위로 matrix 연산이 가능하게 하고, 식 변환을 통해 chunk 단위로 update

$$\sum_{i=1} \theta_i \left(\frac{\beta_t}{\beta_i}\right) \nabla l\,(W_0;\ x_i) = \boxed{\Theta_b B_b (W_0 X - X) X^T}$$

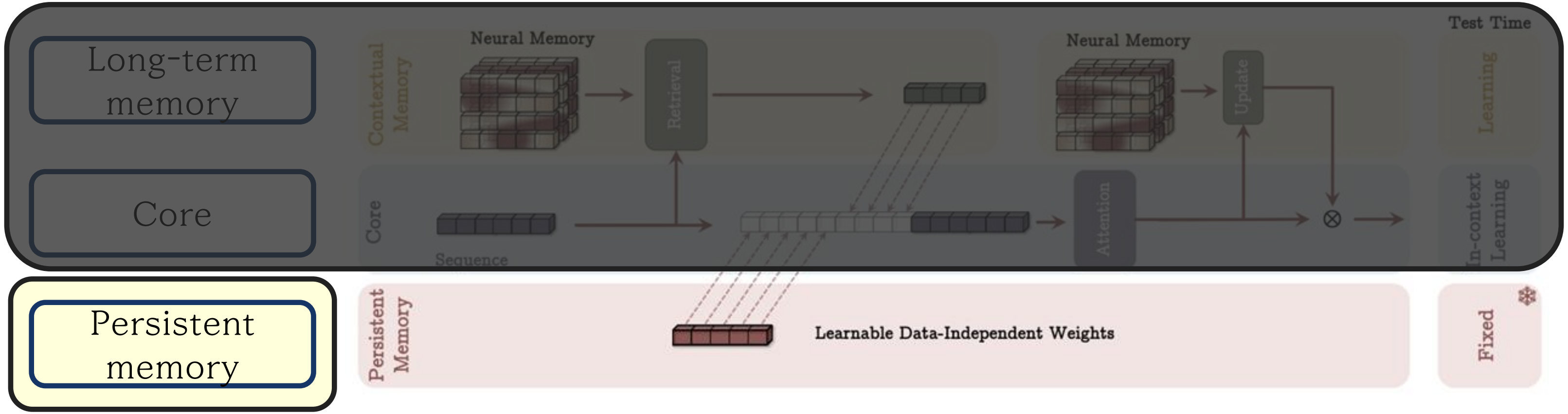
논문 리뷰

지금까지 우리는

‘Long term Memory’를 설계

논문 리뷰

Titans: Learning to Memorize at Test Time



Long-term memory

Core

Persistent memory

변하지 않는 메모리를 설계해보자.

논문 리뷰

Persistent memory

Persistent memory가 왜 따로 필요할까?

1. Input-independent

2. Attention-like weights

3. Initial bias

$$\boxed{x_{\{new\}} = \boxed{\begin{bmatrix} p_1 & p_2 & \dots & p_{\{N_p\}} \end{bmatrix}} || \, x}$$

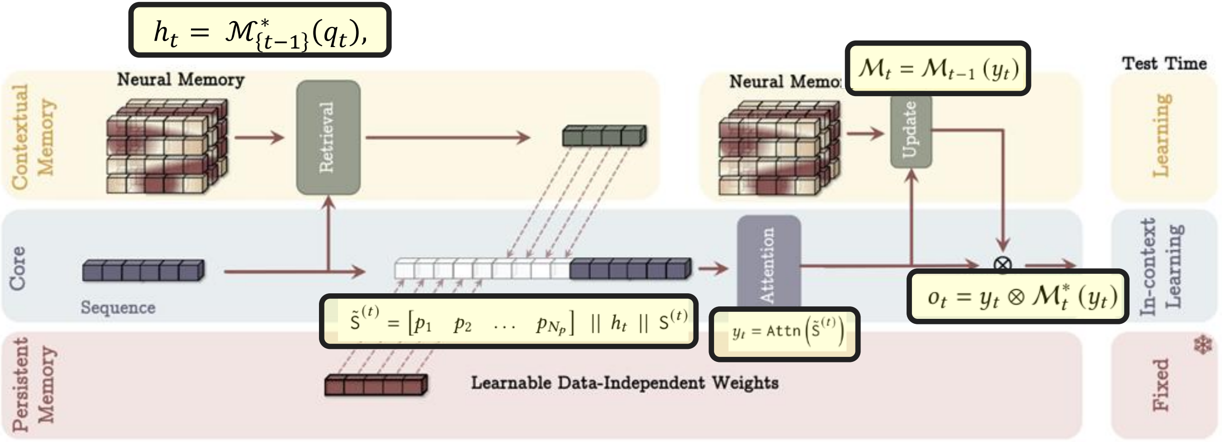
논문 리뷰

1. Memory as a Context

Long-term memory

Core

Persistent memory



$h_t = \mathcal{M}^*_{\{t-1\}}(q_t),$

$\mathcal{M}_t = \mathcal{M}_{t-1}(y_t)$

$\tilde{S}^{(t)} = [p_1 \quad p_2 \quad \dots \quad p_{N_p}] \,||\, h_t \,||\, S^{(t)}$

$y_t = \mathrm{Attn}(\tilde{S}^{(t)})$

$o_t = y_t \otimes \mathcal{M}^*_t(y_t)$

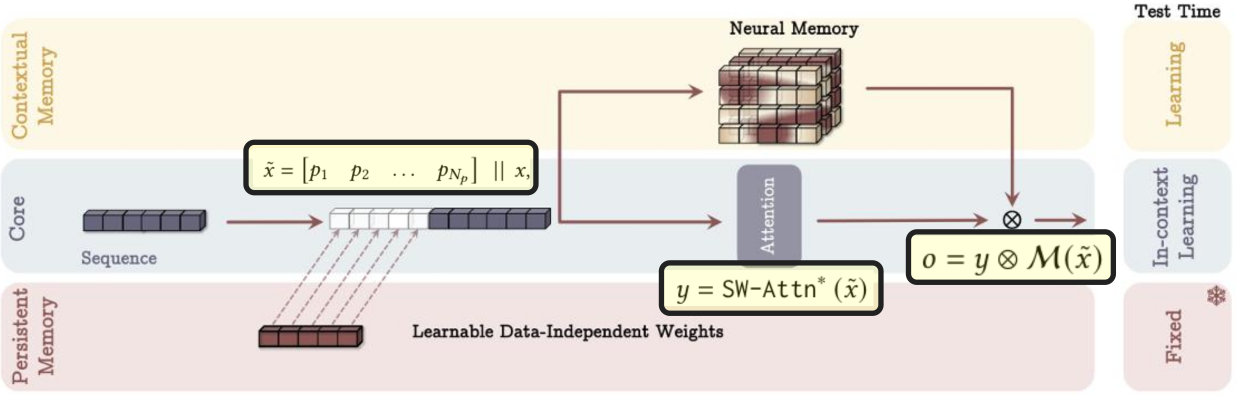
논문 리뷰

2. Gated Memory

Long-term memory

Core

Persistent memory



$\tilde{x} = [p_1 \quad p_2 \quad \dots \quad p_{N_p}] \;||\; x,$

$y = \texttt{SW-Attn}^{*}(\tilde{x})$

$o = y \otimes \mathcal{M}(\tilde{x})$

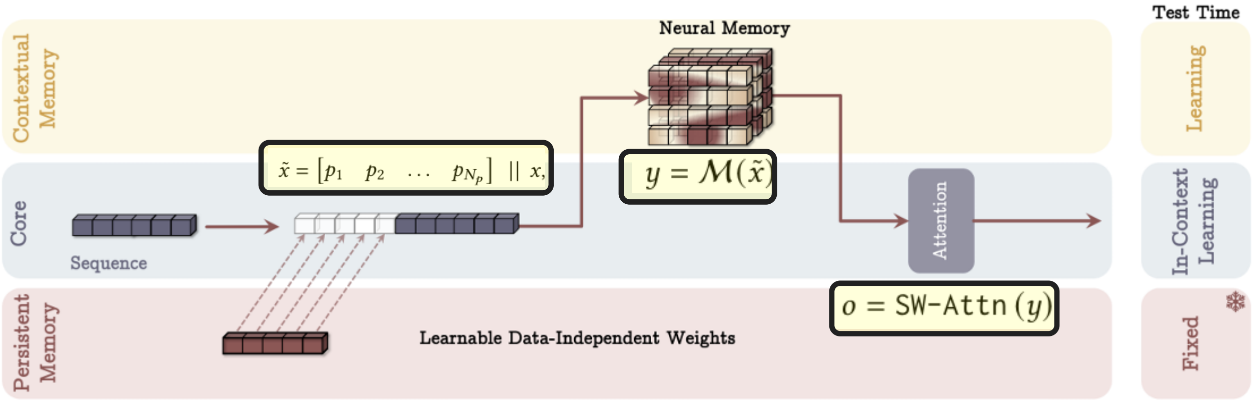
논문 리뷰

3. Memory as a Layer

Long-term memory

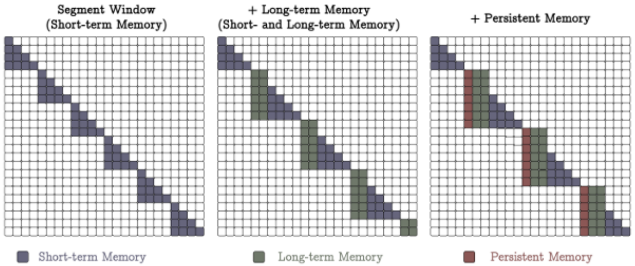
Core

Persistent memory

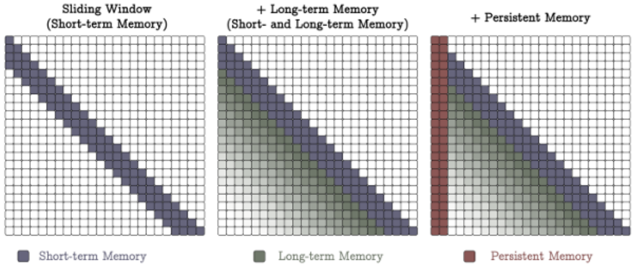

Test Time
Neural Memory
Contextual Memory
Learning
$\tilde{x} = [p_1 \quad p_2 \quad \dots \quad p_{N_p}] \mid\mid x,$
$y = \mathcal{M}(\tilde{x})$
Core
Sequence
Attention
In-Context Learning
$o = \texttt{SW-Attn}(y)$
Persistent Memory
Learnable Data-Independent Weights
Fixed

논문 리뷰

MAC vs MAG

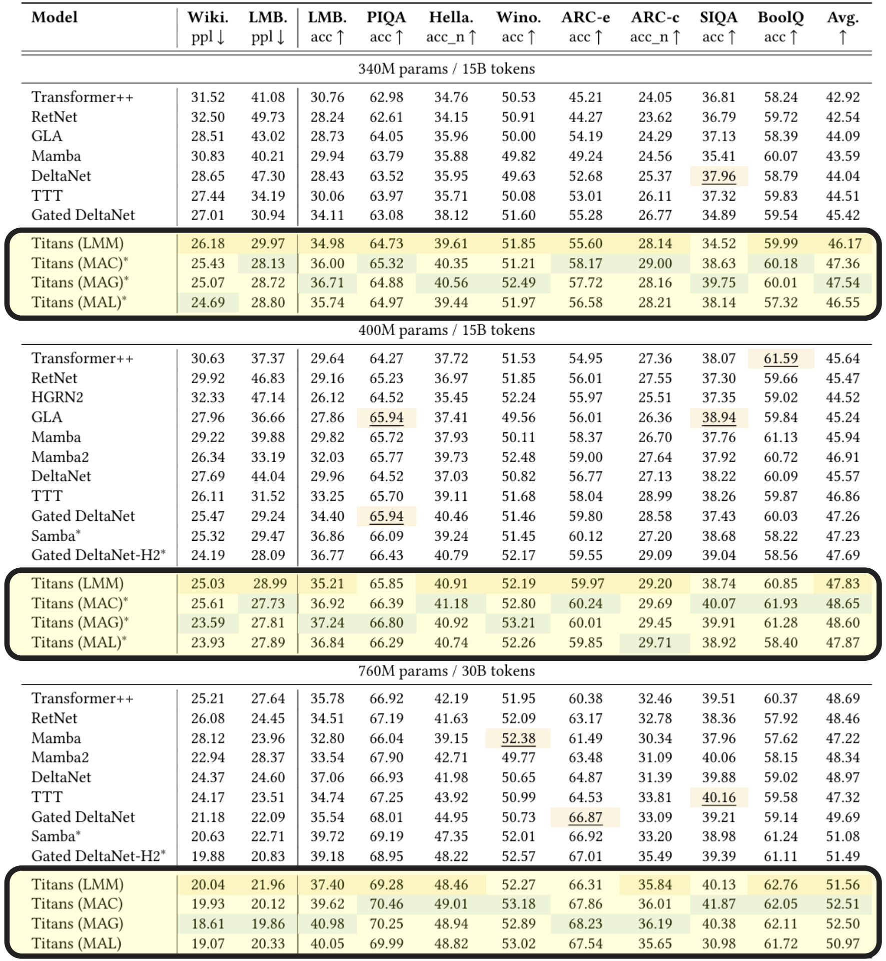


(a) **Memory as a Context (MAC).** We segment the sequence and use full causal attention in each window. Again, the first N_p tokens are persistent memory and the next N_l are long-term memory tokens



(b) **Memory as Gating (MAG).** We use sliding window attention (SWA) as a short-term memory and our neural memory module as a long-term memory, combining by a gating.

논문 리뷰



Model	Wiki. ppl ↓	LMB. ppl ↓	LMB. acc ↑	PIQA acc ↑	Hella. acc_n ↑	Wino. acc ↑	ARC-e acc ↑	ARC-c acc_n ↑	SIQA acc ↑	BoolQ acc ↑	Avg. ↑
340M params / 15B tokens											
Transformer++	31.52	41.08	30.76	62.98	34.76	50.53	45.21	24.05	36.81	58.24	42.92
RetNet	32.50	49.73	28.24	62.61	34.15	50.91	44.27	23.62	36.79	59.72	42.54
GLA	28.51	43.02	28.73	64.05	35.96	50.00	54.19	24.29	37.13	58.39	44.09
Mamba	30.83	40.21	29.94	63.79	35.88	49.82	49.24	24.56	35.41	60.07	43.59
DeltaNet	28.65	47.30	28.43	63.52	35.95	49.63	52.68	25.37	37.96	58.79	44.04
TTT	27.44	34.19	30.06	63.97	35.71	50.08	53.01	26.11	37.32	59.83	44.51
Gated DeltaNet	27.01	30.94	34.11	63.08	38.12	51.60	55.28	26.77	34.89	59.54	45.42
Titans (LMM)	26.18	29.97	34.98	64.73	39.61	51.85	55.60	28.14	34.52	59.99	46.17
Titans (MAC)*	25.43	28.13	36.00	65.32	40.35	51.21	58.17	29.00	38.63	60.18	47.36
Titans (MAG)*	25.07	28.72	36.71	64.88	40.56	52.49	57.72	28.16	39.75	60.01	47.54
Titans (MAL)*	24.69	28.80	35.74	64.97	39.44	51.97	56.58	28.21	38.14	57.32	46.55
400M params / 15B tokens											
Transformer++	30.63	37.37	29.64	64.27	37.72	51.53	54.95	27.36	38.07	61.59	45.64
RetNet	29.92	46.83	29.16	65.23	36.97	51.85	56.01	27.55	37.30	59.66	45.47
HGRN2	32.33	47.14	26.12	64.52	35.45	52.24	55.97	25.51	37.35	59.02	44.52
GLA	27.96	36.66	27.86	65.94	37.41	49.56	56.01	26.36	38.94	59.84	45.24
Mamba	29.22	39.88	29.82	65.72	37.93	50.11	58.37	26.70	37.76	61.13	45.94
Mamba2	26.34	33.19	32.03	65.77	39.73	52.48	59.00	27.64	37.92	60.72	46.91
DeltaNet	27.69	44.04	29.96	64.52	37.03	50.82	56.77	27.13	38.22	60.09	45.57
TTT	26.11	31.52	33.25	65.70	39.11	51.68	58.04	28.99	38.26	59.87	46.86
Gated DeltaNet	25.47	29.24	34.40	65.94	40.46	51.46	59.80	28.58	37.43	60.03	47.26
Samba*	25.32	29.47	36.86	66.09	39.24	51.45	60.12	27.20	38.68	58.22	47.23
Gated DeltaNet-H2*	24.19	28.09	36.77	66.43	40.79	52.17	59.55	29.09	39.04	58.56	47.69
Titans (LMM)	25.03	28.99	35.21	65.85	40.91	52.19	59.97	29.20	38.74	60.85	47.83
Titans (MAC)*	25.61	27.73	36.92	66.39	41.18	52.80	60.24	29.69	40.07	61.93	48.65
Titans (MAG)*	23.59	27.81	37.24	66.80	40.92	53.21	60.01	29.45	39.91	61.28	48.60
Titans (MAL)*	23.93	27.89	36.84	66.29	40.74	52.26	59.85	29.71	38.92	58.40	47.87
760M params / 30B tokens											
Transformer++	25.21	27.64	35.78	66.92	42.19	51.95	60.38	32.46	39.51	60.37	48.69
RetNet	26.08	24.45	34.51	67.19	41.63	52.09	63.17	32.78	38.36	57.92	48.46
Mamba	28.12	23.96	32.80	66.04	39.15	52.38	61.49	30.34	37.96	57.62	47.22
Mamba2	22.94	28.37	33.54	67.90	42.71	49.77	63.48	31.09	40.06	58.15	48.34
DeltaNet	24.37	24.60	37.06	66.93	41.98	50.65	64.87	31.39	39.88	59.02	48.97
TTT	24.17	23.51	34.74	67.25	43.92	50.99	64.53	33.81	40.16	59.58	47.32
Gated DeltaNet	21.18	22.09	35.54	68.01	44.95	50.73	66.87	33.09	39.21	59.14	49.69
Samba*	20.63	22.71	39.72	69.19	47.35	52.01	66.92	33.20	38.98	61.24	51.08
Gated DeltaNet-H2*	19.88	20.83	39.18	68.95	48.22	52.57	67.01	35.49	39.39	61.11	51.49
Titans (LMM)	20.04	21.96	37.40	69.28	48.46	52.27	66.31	35.84	40.13	62.76	51.56
Titans (MAC)	19.93	20.12	39.62	70.46	49.01	53.18	67.86	36.01	41.87	62.05	52.51
Titans (MAG)	18.61	19.86	40.98	70.25	48.94	52.89	68.23	36.19	40.38	62.11	52.50
Titans (MAL)	19.07	20.33	40.05	69.99	48.82	53.02	67.54	35.65	30.98	61.72	50.97

논문 리뷰

정리하면…

어떻게 기억할까?에 대해서 ‘Memory’라는 새로운 구조를 도입해서 유기적으로 기억을 저장하고 끄집어낼 수 있는 아키텍처를 설계했다.

감사합니다.