HRM & TRM

Multimodal Orchestration of Artificial Intelligence (MOAI) Lab

발표자 : 이수찬

Overview

- 최근에 삼성에서 TRM(Tiny Recursive Model) 논문을 publish해서 주목 받음
- 한 줄 요약하자면 HRM(Hierarchical Reasoning Model) 논문의 문제점을 지적한 논문
- HRM의 문제점을 지적하고 개선한 최신 looped transformer (recursive transformer) 논문

Hierarchical Reasoning Model

Guan Wang[1,†], Jin Li[1], Yuhao Sun[1], Xing Chen[1], Changling Liu[1],
Yue Wu[1], Meng Lu[1,†], Sen Song[2,†], Yasin Abbasi Yadkori[1,†]

[1]Sapient Intelligence, Singapore

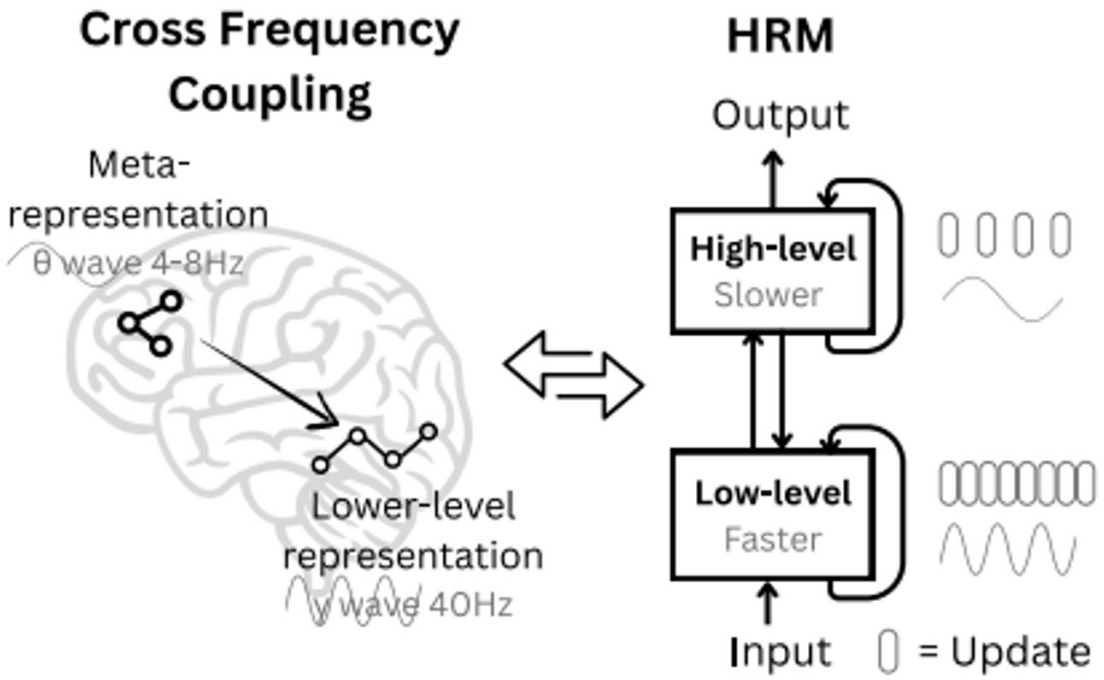
Publication date : 2025.10.06

Motivation

- 현재 대부분의 LLM들은 CoT 기법을 통해 reasoning 성능을 높이는 방법을 사용하고 있음
- CoT는 prompt level에서 reasoning 성능을 향상시키고자 하였는데, 저자들은 언어는 인간의 의사소통을 위한 도구이지, 사고의 기질 그 자체가 아니라고 주장

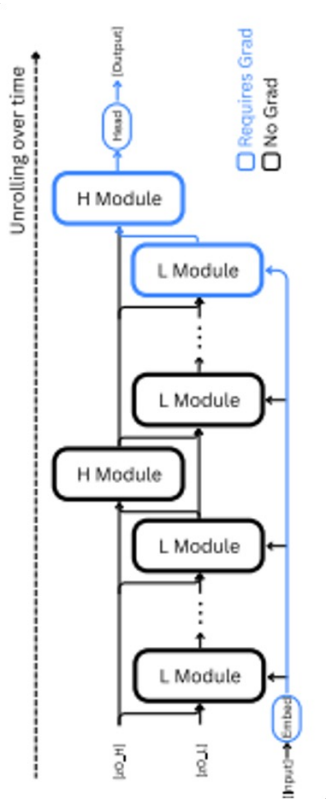
its internal hidden state space[15,16]. This aligns with the understanding that language is a tool for human communication, not the substrate of thought itself[17]; the brain sustains lengthy, coherent chains of reasoning with remarkable efficiency in a latent space, without constant translation back to language. However, the power of latent reasoning is still fundamentally constrained by a model's

Motivation



- 뇌는 서로 다른 시간 척도에서 작동하는 피질 영역 전반에 걸쳐 연산을 계층적으로 조직하여, 깊이 있는 다단계 추론을 가능하게 함
- 인간의 뇌의 생물학적 모델링에 영감을 받아 2개의 recursive module을 결합시켜 HRM을 제안

HRM



$$i = 1, \ldots, N \times T.$$

$$\tilde{x} = f_I(x; \theta_I) \ .$$

$$z_L^i = f_L\left(z_L^{i-1}, z_H^{i-1}, \tilde{x}; \theta_L\right) \ ,$$

$$z_H^i = \begin{cases} f_H\left(z_H^{i-1}, z_L^{i-1}; \theta_H\right) & \text{if } i \equiv 0 \ (\text{mod } T) \ , \\ z_H^{i-1} & \text{otherwise} \ . \end{cases}$$

$$\hat{y} = f_O(z_H^{NT}; \theta_O) \ .$$

- 느리게 작동하는 High-level module이 guide하고, 빠르게 작동하는 Low-level module이 execute하는 하위 처리를 가능하게 하면서도 전역적인 일관성을 유지한다고 주장

Hierarchical convergence

- 일반적인 RNN 구조는 forward pass를 반복하다 보면 fixed point에 조기 수렴하게 되어 effective depth가 제한됨
- HRM은 H-module, L-module을 구분
- L-module이 한 cycle동안 fixed point에 수렴하더라도 z_H라는 dependence를 줌으로써 H-module이 update될 경우 새로운 수렴 단계를 시작할 수 있도록 해줌 (Hierarchical convergence)

Approximate gradient

- 기존의 RNN 구조는 gradient update를 위해선 T 번의 타임스텝이 있다면 O(T)의 메모리를 요구
- 앞서 언급한 fixed point로의 조기 수렴 문제를 이용해 L-module의 전체 타임스텝의 gradient를 활용하는게 아니라, 마지막 타임스텝의 gradient만 활용해서 update 함으로써 O(1)의 메모리로 절감

$$z_L^i = f_L\left(z_L^{i-1}, z_H^{i-1}, \tilde{x}; \theta_L\right) ,$$
$$z_H^i = \begin{cases} f_H\left(z_H^{i-1}, z_L^{i-1}; \theta_H\right) & \text{if } i \equiv 0 \pmod{T} , \\ z_H^{i-1} & \text{otherwise} . \end{cases}$$

$$z_L^\star = f_L(z_L^\star, z_H^{k-1}, \tilde{x}; \theta_L) .$$
$$z_H^k = f_H(z_H^{k-1}, z_L^\star; \theta_H) .$$
$$z_H^\star = \mathcal{F}(z_H^\star; \tilde{x}, \theta). \quad ?$$

$$\frac{\partial z_H^\star}{\partial \theta} = \left(I - J_{\mathcal{F}}\big|_{z_H^\star}\right)^{-1} \frac{\partial \mathcal{F}}{\partial \theta}\bigg|_{z_H^\star} .$$

Deep supervision

- L-module이 T번, H-module이 N번 forward되는 단위를 segment라고 하고, 이를 M 번 실행
- 하나의 segment가 forward pass될 때마다 gradient update

At each segment m, we apply a deep supervision step as follows:

1. Given the state z^{m-1} from the previous segment, compute the next state z^m and its associated output $\hat{y}^m$ through a forward pass in the HRM model:

$$(z^m, \hat{y}^m) \leftarrow \text{HRM}(z^{m-1}, x; \theta)$$

2. Compute the loss for the current segment:

$$L^m \leftarrow \text{LOSS}(\hat{y}^m, y)$$

3. Update parameters:

$$\theta \leftarrow \text{OPTIMIZERSTEP}(\theta, \nabla_\theta L^m)$$

Adaptive computational time (ACT)

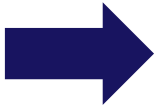
- Segment의 크기는 어떻게 정할까?
- Inference 시에는 segement의 크기를 정해놓지만 training을 할 때는 유연하게 computation하기 위해서 Adaptive computational time을 적용 (Q-learning)

$$\hat{Q}^m = \sigma(\theta_Q^\top z_H^{mNT})$$

$$\hat{Q}^m = (\hat{Q}^m_{\text{halt}}, \hat{Q}^m_{\text{continue}})$$

$$\hat{G}^m_{\text{halt}} = \mathbf{1}\{\hat{y}^m = y\}\,,$$

$$\hat{G}^m_{\text{continue}} = \begin{cases} \hat{Q}^{m+1}_{\text{halt}}\,, & \text{if } m \geq N_{\max}\,, \\ \max(\hat{Q}^{m+1}_{\text{halt}}, \hat{Q}^{m+1}_{\text{continue}})\,, & \text{otherwise}\,. \end{cases}$$



$$L^m_{\text{ACT}} = \text{Loss}(\hat{y}^m, y) + \text{BinaryCrossEntropy}(\hat{Q}^m, \hat{G}^m)\,.$$

HRM

T=2, N=1, M=2인 HRM 예시

$$
\begin{aligned}
x &\leftarrow f_I(\tilde{x}) \\
z_L &\leftarrow f_L\left(z_L + z_H + x\right) && \text{\# without gradients} \\
z_L &\leftarrow f_L\left(z_L + z_H + x\right) && \text{\# without gradients} \\
z_H &\leftarrow f_H\left(z_L + z_H\right) && \text{\# without gradients} \\
z_L &\leftarrow f_L\left(z_L + z_H + x\right) && \text{\# without gradients} \\
z_L &\leftarrow z_L.detach() \\
z_H &\leftarrow z_H.detach() \\
z_L &\leftarrow f_L\left(z_L + z_H + x\right) && \text{\# with gradients} \\
z_H &\leftarrow f_H\left(z_L + z_H\right) && \text{\# with gradients} \\
\hat{y} &\leftarrow \text{argmax}\left(f_O\left(z_H\right)\right)
\end{aligned}
$$

Less is More: Recursive Reasoning with Tiny Networks

Alexia Jolicoeur-Martineau
Samsung SAIL Montréal
alexia.j@samsung.com

Publication date : 2025.10.06

Motivation

- 최적의 HRM 구조를 찾고자 엔지니어링 요소가 좀 더 가미됨
- **Question 1) Low-level module이 실제로 equilibrium point에 도달하는가?**
- **Question 2) 2개의 module을 왜 사용해야 하나?**
- **Question 3) ACT를 위한 Q-learning이 computational cost를 생각했을때 꼭 필요한가?**

Question 1

- HRM 논문에서 1-step gradient update를 위해서 L-module에서의 cycle 이후에 fixed point를 가정
- HRM 논문에서 fixed point 주장을 위해서 사용한 T, N을 실제 실험에선 사용하지 않음

1. the residual for z_H is clearly well above 0 at every step

   앞서 IFT 사용을 위한 H-module의 fixed point 가정도 틀린 가정이었음

2. the residual for z_L only becomes closer to 0 after many cycles, but it remains significantly above 0

3. z_L is very far from converged after one f_L evaluation at T cycles, which is when the fixed-point is assumed to be reached and the 1-step gradient approximation is used

Solution 1

- 1-step gradient를 사용하기 위한 가정자체가 성립하지 않음
- 기존에는 segment마다 1-step gradient update를 하던 방식
- 마지막 segment에서 full-step gradient update

Table 1. Ablation of TRM on Sudoku-Extreme comparing % Test accuracy, effective depth per supervision step ($T(n+1)n_{layers}$), number of Forward Passes (NFP) per optimization step, and number of parameters

Method	Acc (%)	Depth	NFP	# Params
HRM	55.0	24	2	27M
TRM ($T=3, n=6$)	87.4	42	1	5M
w/ ACT	86.1	42	2	5M
w/ separate f_H, f_L	82.4	42	1	10M
no EMA	79.9	42	1	5M
w/ 4-layers, $n=3$	79.5	48	1	10M
w/ self-attention	74.7	42	1	7M
w/ $T=2, n=2$	73.7	12	1	5M
w/ 1-step gradient	56.5	42	1	5M

Question 2

- HRM의 핵심 motivation 중 하나 - *뇌는 서로 다른 시간 척도에서 작동하는 피질 영역 전반에 걸쳐 연산을 계층적으로 조직하여, 깊이 있는 다단계 추론을 가능하게 함*
- 그렇다면 꼭 2개의 module을 사용해야 할 필요가 없어 해당 직관은 다소 비약이 있음
- 질문(x), 추론 임베딩(z_L), 해답 임베딩(z_H) 컨셉이라고 해석 가능하다고 주장
- Recursive path에서 추론 임베딩을 누락 시킬 경우 논리적 과정을 잊게 되고, 해답 임베딩을 누락시킬 경우 추론 과정에서 이전의 해답을 잊게 됨

Table 2. TRM on Sudoku-Extreme comparing % Test accuracy when using more or less latent features

Method	# of features	Acc (%)
TRM y, z (Ours)	2	87.4
TRM multi-scale z	$n+1=7$	77.6
TRM single z	1	71.9

Solution 2

- 결국 hierarchical한 module로의 분리가 중요한게 아니라 추론 임베딩과 해답 임베딩을 recursive reasoning에 활용하는게 중요하다는게 저자들의 직관
- 따라서 single layer로 입력에 따라서 execution을 다르게 하는 식으로 parameter를 절반으로 감소 시킴

Table 1. Ablation of TRM on Sudoku-Extreme comparing % Test accuracy, effective depth per supervision step ($T(n+1)n_{layers}$), number of Forward Passes (NFP) per optimization step, and number of parameters

Method	Acc (%)	Depth	NFP	# Params
HRM	55.0	24	2	27M
TRM ($T=3, n=6$)	87.4	42	1	5M
w/ ACT	86.1	42	2	5M
w/ separate f_H, f_L	82.4	42	1	10M
no EMA	79.9	42	1	5M
w/ 4-layers, $n=3$	79.5	48	1	10M
w/ self-attention	74.7	42	1	7M
w/ $T=2, n=2$	73.7	12	1	5M
w/ 1-step gradient	56.5	42	1	5M

Question 3

- ACT를 이용하면 training level에서 computational cost를 줄일 수 있음
- 하지만 미래 Q-value 계산을 위해선 computation을 한번 더 해야되는 비효율적인 문제가 생기는데, HRM 저자들은 이에 대한 논의를 누락
- ACT를 이용한 training cost 절감 vs. 미래 Q-value 계산을 위한 computation cost

Solution 3

- 문제는 미래 Q-value 계산에서 오는 computation cost
- 미래 Q-value 계산을 제외한 binary cross entropy loss를 사용

$$\hat{Q}^m = \sigma(\theta_Q^\top z_H^{mNT})$$

$$\hat{Q}^m = (\hat{Q}^m_{\text{halt}}, \hat{Q}^m_{\text{continue}})$$

$$\hat{G}^m_{\text{halt}} = \mathbf{1}\{\hat{y}^m = y\}\,,$$

$$\hat{G}^m_{\text{continue}} = \begin{cases} \hat{Q}^{m+1}_{\text{halt}}\,, & \text{if } m \geq N_{\max}\,, \\ \max(\hat{Q}^{m+1}_{\text{halt}}, \hat{Q}^{m+1}_{\text{continue}})\,, & \text{otherwise}\,. \end{cases}$$

$$L^m_{\text{ACT}} = \text{Loss}(\hat{y}^m, y) + \text{BinaryCrossEntropy}(\hat{Q}^m, \hat{G}^m)\,.$$

Table 1. Ablation of TRM on Sudoku-Extreme comparing % Test accuracy, effective depth per supervision step $(T(n+1)n_{layers})$, number of Forward Passes (NFP) per optimization step, and number of parameters

Method	Acc (%)	Depth	NFP	# Params
HRM	55.0	24	2	27M
TRM ($T=3, n=6$)	87.4	42	1	5M
w/ ACT	86.1	42	2	5M
w/ separate f_H, f_L	82.4	42	1	10M
no EMA	79.9	42	1	5M
w/ 4-layers, $n=3$	79.5	48	1	10M
w/ self-attention	74.7	42	1	7M
w/ $T=2, n=2$	73.7	12	1	5M
w/ 1-step gradient	56.5	42	1	5M

Experiments & Results

Table 3. % Test accuracy on Sudoku-Extreme dataset. HRM versus TRM matched at a similar effective depth per supervision step $(T(n+1)n_{layers})$

		HRM $n = k$, 4 layers		TRM $n = 2k$, 2 layers	
k	T	Depth	Acc (%)	Depth	Acc (%)
1	1	9	46.4	7	63.2
2	2	24	55.0	20	81.9
3	3	48	61.6	42	87.4
4	4	80	59.5	72	84.2
6	3	84	62.3	78	OOM
3	6	96	58.8	84	85.8
6	6	168	57.5	156	OOM

Experiments & Results

Table 4. % Test accuracy on Puzzle Benchmarks (Sudoku-Extreme and Maze-Hard)

Method	# Params	Sudoku	Maze
Chain-of-thought, pretrained			
Deepseek R1	671B	0.0	0.0
Claude 3.7 8K	?	0.0	0.0
O3-mini-high	?	0.0	0.0
Direct prediction, small-sample training			
Direct pred	27M	0.0	0.0
HRM	27M	55.0	74.5
TRM-Att (Ours)	7M	74.7	85.3
TRM-MLP (Ours)	5M/19M[1]	87.4	0.0

Experiments & Results

Table 5. % Test accuracy on ARC-AGI Benchmarks (2 tries)

Method	# Params	ARC-1	ARC-2
Chain-of-thought, pretrained			
Deepseek R1	671B	15.8	1.3
Claude 3.7 16K	?	28.6	0.7
o3-mini-high	?	34.5	3.0
Gemini 2.5 Pro 32K	?	37.0	4.9
Grok-4-thinking	1.7T	66.7	16.0
Bespoke (Grok-4)	1.7T	79.6	29.4
Direct prediction, small-sample training			
Direct pred	27M	21.0	0.0
HRM	27M	40.3	5.0
TRM-Att (Ours)	7M	44.6	7.8
TRM-MLP (Ours)	19M	29.6	2.4

Conclusion

- HRM의 여러 문제점을 지적했지만 TRM도 읽으면서 의문이 들었던 부분이 있음 (마지막 segment에서 full-step gradient update를 해도 괜찮은가?)
- 개인적으론 MDLM과 AR의 하이브리드 형태인 semi AR 모델에 관심을 가졌던 입장에서 HRM에서 제안했던 high level module, low level module의 컨셉과 reasoning을 architecture level에서 접근하는 점은 매력적으로 다가왔음

EOD