Mechanistic Interpretability

2025 동계세미나

김동준

Natural Language Processing
& Artificial Intelligence

Background

What is Mechanistic Interpretability?

- Neural network를 인간이 이해할 수 있는 알고리즘으로 reverse engineer 하려는 접근법
- LLM의 기본적인 작동 방식을 이해하는 것을 목표로 함
- Black Box System => Transparent System

Key Components

- Neurons: FFN의 기본 계산 단위
  - $neuron(x) = \sigma(wTx + b)$
  - 따라서 num(neuron) = num(parameters)
- Features: NN representation의 기본 단위
  - e.g. coding ability, golden gate bridge
- Circuits: 특정 feature를 수행하기 위해 상호작용하는 neuron 그룹
  - Subgraph of neurons within a neural network that works together to perform a specific computational task or algorithm
  - e.g. Sequence continuation circuit, Edge detection circuit, coding circuit, golden gate bridge circuit

Background

Why Mechanistic Interpretability Matters

Core Benefits

- 상관관계 이해가 아닌 Causal Understanding 가능
- 정확한 model editing 가능
- 더 깊은 모델 이해를 통한 AI safety 향상
- 더 통제되고 trustworthy 한 LLM 개발 가능

Current Applications

- Training phenomena 이해 (grokking, memorization)
- Transformer architecture 내부 분석
- 더 나은 model debugging tool 개발

- Causal Tracing
- Sparse AutoEncoders

Arithmetic

Interpreting Arithmetic Mechanism in Large Language Models through Comparative Neuron Analysis

Zeping Yu **Sophia Ananiadou**
Department of Computer Science, The University of Manchester
{zeping.yu@postgrad. sophia.ananiadou@}manchester.ac.uk

Interpreting and Improving Large Language Models in Arithmetic Calculation

Wei Zhang [*1 2] **Chaoqun Wan** [2] **Yonggang Zhang** [†3] **Yiu-ming Cheung** [3] **Xinmei Tian** [1 4] **Xu Shen** [†2]
Jieping Ye [2]

Instruction Following

From Language Modeling to Instruction Following: Understanding the Behavior Shift in LLMs after Instruction Tuning

Xuansheng Wu[♣*†], **Wenlin Yao**[♡‡], **Jianshu Chen**[♡],
Xiaoman Pan[♡], **Xiaoyang Wang**[♡], **Ninghao Liu**[♣], **Dong Yu**[♡]
[♣]University of Georgia [♡]Tencent AI Lab, Bellevue

JAILBREAK INSTRUCTION-TUNED LLMS VIA END-OF-SENTENCE MLP RE-WEIGHTING

Yifan Luo & Meitan Wang
School of Mathemetical Science
Peking University
luoyf@pku.edu.cn

Zhennan Zhou
School of Science
Westlake University
zhouzhennan@westlake.edu.cn

Bin Dong
Beijing International Center for Mathematical Research
Center for Machine Learning Research
Peking University
dongbin@math.pku.edu.cn

Interpreting Arithmetic Mechanism in Large Language Models through Comparative Neuron Analysis

Zeping Yu Sophia Ananiadou
Department of Computer Science, The University of Manchester
{zeping.yu@postgrad. sophia.ananiadou@}manchester.ac.uk

EMNLP 2024 Main

1. Comparative Neuron Analysis 방법론 제안
   - Final model prediction까지의 internal logic chain을 찾아내는 방법
   - Arithmetic ability를 가진 FFN neuron을 바로 찾는 것이 아니라, 이를 trigger하는 attention head 부터 찾는 식
2. Arithmetic Ability는 Attention Head 몇개만이 담당함
   - 각 Attention Head가 담당하는 Operation이 서로 다름

Interpreting Arithmetic Mechanism in Large Language Models through Comparative Neuron Analysis

Arithmetic Heads in LLMs

Attention Head Intervention 실험을 통해 Arithmetic Head의 존재 증명

- Llama-7B (32 layers * 32 attention head for each layer = 1,024 attention heads in total)
- Arithmetic evaluation dataset 제작 (4가지 subset (+ - * /) 총1,600개)
- Attention Head들의 파라미터를 한번에 하나씩 0으로 damage 시킨 후 accuracy 측정 (1,024 * 1,600 번)

결과

- 5개의 attention head이외의 나머지 모든 head를 없앴을 때는 0.01-2% accuracy 감소
- 5개의 attention head를 없앴을 때는10% 정도에서 21.4%까지 accuracy가 감소
- 또, 이 attention head들이 영향을 주는 연산이 서로 다름
- 1,2,3 Digit에서 각 연산에 영향을 가장 많이 주는 attention head는 같음
  - 곱셈은 futu

	ori	17^{22}	15^{9}	14^{19}	15^{23}	16^{1}
all	74.8	53.4	62.1	62.7	68.1	68.7
2D+	96.8	**42.9**	83.2	92.5	89.7	91.6
2D-	94.4	**72.3**	84.6	93.2	86.5	**79.1**
2D*	56.6	50.5	50.9	51.3	52.3	56.9
2D/	51.4	48.2	**29.5**	**13.8**	43.8	47.1

Table 1: Accuracy (%) when intervening different heads. "ori": original model. 17^{22}: $22th$ head in $17th$ layer.

	17^{22}(+)	17^{22}(-)	20^{18}(*)	14^{19}(/)
1D	46.5	62.2	6.8	54.9
2D	58.4	52.6	11.2	71.8
3D	52.5	56.9	8.1	53.2

Table 2: Accuracy decrease (%) in 1D, 2D and 3D.

Comparative Neuron Analysis

- '17^{22} head가 덧셈/뺄셈에 영향을 많이 준다'에서 끝내지 않고, 17^{22} head를 없애는 것이 모델에 정확히 어떠한 영향을 주는지 분석
- Attention Head는 arithmetic ability에 영향을 주는 FFN neuron을 activate시키는 역할이라고 주장
  - 이러한 FFN neuron을 찾는 방법이 CNA
  - 이러한 주장을 증명하기 위해 뒤에 실험들 진행

CNA

- 다른 관련 연구들과 동일한 Importance Score 방법
- Importance Score를 구하는 방법을 다르게 함
  - PT를 하며 파라미터를 손상시키고 그 파라미터의 loss 값을 보는 것이 아닌
  - 파라미터를 손상시키고 Inference 할 때, 모델 답변의 log probability 변화량을 importance score라고 정의
- Attention Head 손상이 없는 original model과 Attention Head을 손상시킨

Feature Predicting via Arithmetic Heads

- Attention Head는 arithmetic ability에 영향을 주는 FFN neuron을 activate시키는 역할이라고 주장

  ⇒ 17^{22} head를 손상시킨 모델을 arithmetic query로 inference 시켰을 때의 neuron들 중 가장 Importance Score가 큰 neuron이 실제 덧셈/뺄셈 능력을 가지고 있는 neuron이다

  ⇒ 17^{22} head를 손상시킨 모델을 "3 + 5 ="로 inference 시켰을 때의 neuron들 중 가장 Importance Score가 큰 neuron은 28_{3696}이였고 이 neuron이 실제 덧셈/뺄셈 능력을 가지고 있는지 확인하기 위해

  1. Unembedding matrix와 곱하여 top 10 token을 확인
     - 전부 8과 관련된 token
  2. Original model (attention head 손상 x)에서 28_{3696} neuron을 없애고 accuracy 측정
     - mask top 99는 가장 중요한 99개의 neuron 손상 => accuracy 100% 감소
     - keep top 99는 가장 중요한 99개 이외의 모든 neuron 손상 => accuracy 감소 적음

     => top 99개의 FFN neuron이 대부분의 덧셈/뺄셈 담당함을 증명

FFNv	mdl	imp	coef	top10 tokens
28_{3696}	ori	0.82	6.21	**[8, eight, VIII,**
28_{3696}	inv	0.13	0.95	huit, acht, otto]
25_{7164}	ori	0.31	8.44	[six, **eight**, acht,
25_{7164}	inv	0.07	2.08	Four, twelve, six, four, vier]
19_{5769}	ori	0.20	3.79	**[eight, VIII, 8,**
19_{5769}	inv	0.06	1.28	III, huit, acht]

Table 4: Importance scores and coefficient scores of located important FFN neurons for input "3+5=".

	top99	top50	top30	top20	top10
mask	100.0	96.0	89.5	86.8	68.4
keep	3.9	7.8	13.2	18.4	38.2
coef	49.1	60.4	67.2	72.7	77.1

Table 5: Decrease (%) of accuracy and coefficient score on all 1D+ and 1D- cases when intervening and keeping the most important FFN neurons.

Interpreting and Improving Large Language Models in Arithmetic Calculation

Wei Zhang [*1 2] Chaoqun Wan [2] Yonggang Zhang [†3] Yiu-ming Cheung [3] Xinmei Tian [1 4] Xu Shen [†2]
Jieping Ye [2]

ICML 2024 Oral

- 앞 논문과 동일하게 'Arithmetic Calculation을 가능하게 하는 MLP들이 있고, 이는 특정 Attention Head들로 (<5%)인하여 나타난다'를 찾아 냈다고 주장
  - 여기서 MLP는 파라미터/뉴런 단위가 아니라 전체 MLP 레이어를 뜻함
- 추가로, 이러한 MLP/Attention Head들은 다른 dataset로의 transferbility가 나타나고 심지어 다른 태스크로도 transfer되는 경우가 있다라고 주장
- Arithmetic Calculation을 가능하게 하는 MLP/Attention Head만을 fine tuning 시키면, 다른 태스크의 성능은 유지하면서 Arithmetic Calculation의 성능을 높이는 것이 가능함을 보여줌

Methodology

앞 논문과 동일하게 자체 Arithmetic Evaluation Benchmark 데이터셋 제작

Arithmetic Computation을 담당하는 AH와 MLP를 찾기 위해 *path patching* 사용

- (Sender -> Reciever) 관계를 파악하는데 사용되는 Causal Intervention Technique
- Reference/Counterfactual data의 AH activation 값 Ar, Ac 수집
  - Reference data Xr: 3 + 5 =
  - Counterfactual data Xc: 3 < 5 =
- for n in AH: replace Ar(n) with Ac(n)
- Output logit의 변화 측정하여 변화량이 가장 큰 n을 Key Head라고
- For logits(z), Log Probability $\log P(y_u) = Z_i - \log(\sum_j e^{z_j})$

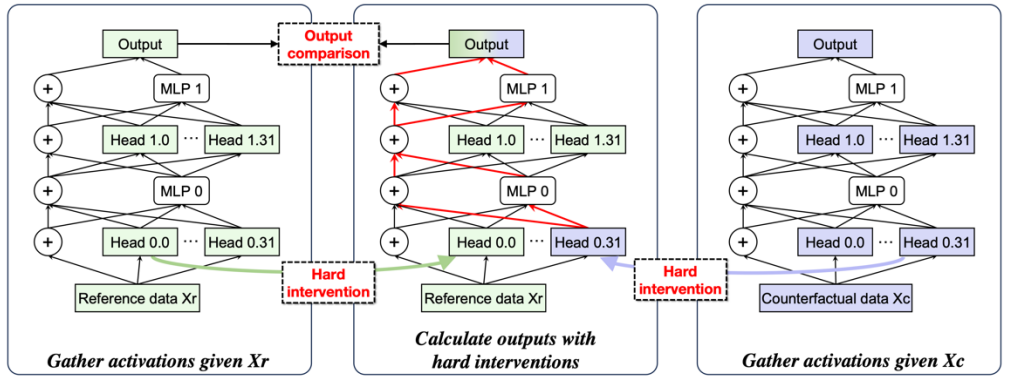

Figure 11: A case illustration of the method "path patching". It measures the importance of forward paths (*i.e.*, the red lines that originate from Head 0.31 to Output) for the two-layer transformer in completing the task on reference data.

Path Patching으로 찾은 Key Head가 실제로 영향력이 크고 나머지 AH는 영향력이 적다는 것을 보여주기 위해 *mean ablation 사용*

- n의 activation 값을 Xc에 대한 activation 값들의 평균으로 바꿔줌

Identification & Verification of Key Heads

- Logit change가 큰 AH는 갯수가 굉장히 적음
- Key Head: logit change < -5%
- Llama2-7B 기준
- Key Head는 대부분 중간 레이어에 있음 (12-17)
- Key MLP는 17번째 레이어 이전까지는 영향력이 전혀 없다 17번째 이후부터 많은 영향을 끼침

- Key Head가 올바르게 찾아졌음을 verify하기 위해 top k head에 mean ablation 진행 결과 random head를 mean ablation 한 것에 비해 accuracy가 감소함

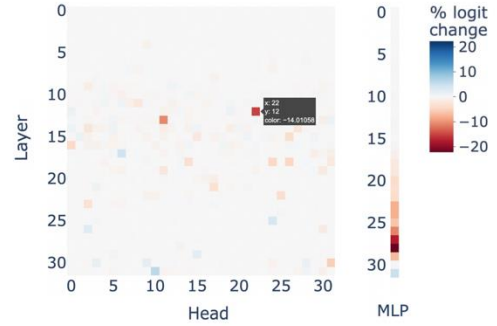

Figure 2: We conduct path patching experiments on LLaMA2-7B across four mathematical tasks, by searching for each head and MLP directly affecting the logit of the right answer. For each head/MLP, a darker color indicates a larger logit difference from the model before patching.

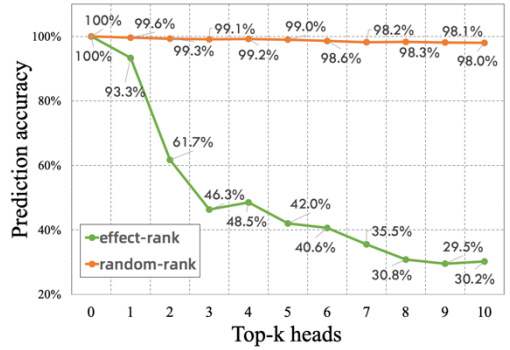

Figure 3: The influence on prediction accuracy after knocking out top-k attention heads that are sorted by the effect of each head on logits ("effect-rank"), and knocking out randomly-sorted top-k heads ("random-rank").

Identification & Verification of Key Heads

- 앞 논문과 같은 결과로 특정 key head는 특정 연산을 담당

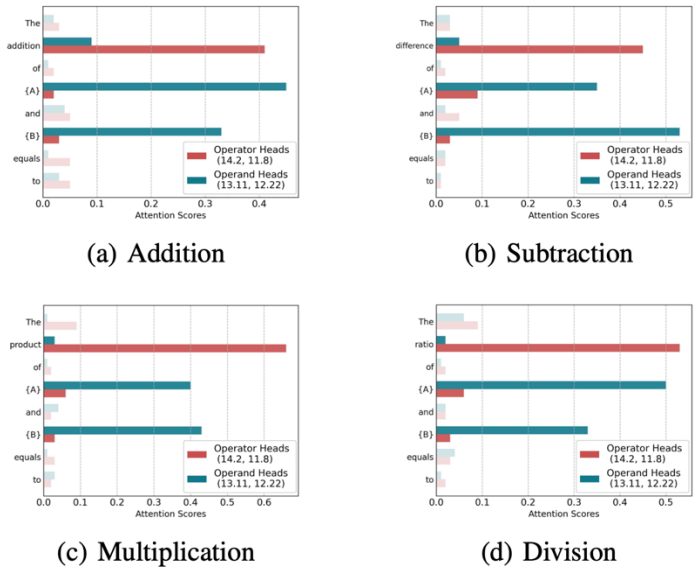

Figure 5: The attention score distribution of key heads across four calculation tasks. The key heads (*e.g.*, 13.11, 14.2) attend to number operands and calculation operators.

- 같은 task의 unseen dataset (SVAMP)를 봤을 때에도, Key Head가 손상된 모델의 accuracy가 떨어짐

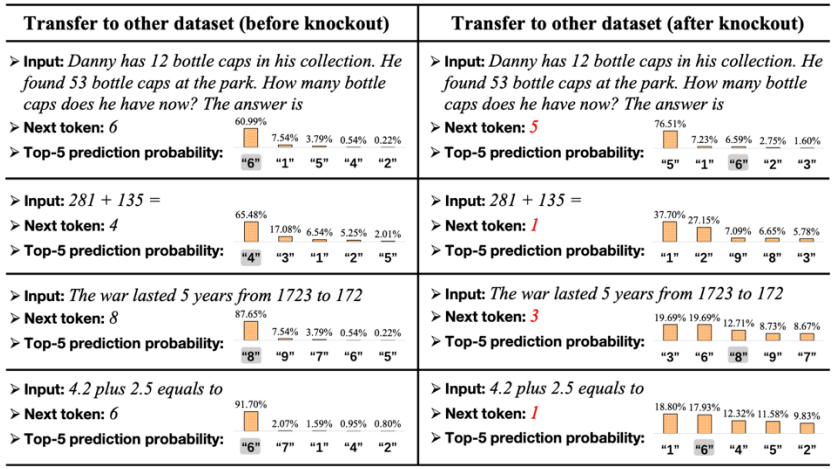

Figure 4: After knocking out the key heads, LLaMA2-7B predicts incorrectly on the cases of SVAMP dataset and other data formats of multi-digit integers, rational numbers.

Precise SFT

- Llama2-7B, Llama2-13B top 32 key head에 대해서만 수학 데이터셋으로 SFT

Table 1: Overall performance. We evaluate the capabilities of LLaMA2-7B and LLaMA2-13B, transitioning from generic tasks (*e.g.*, MMLU and CSQA) to mathematical tasks (*e.g.*, GSM8K, AddSub, SingleEq, and SVAMP). Supervised fine-tuning across the entire parameter set (denoted as Full SFT) leads to enhanced performance in math-related tasks, albeit at the expense of its capabilities in generic tasks. In contrast, selectively tuning only the parameters of 32 critical attention heads (denoted as Precise SFT) yields comparable improvements while preserving the model's proficiency in generic tasks, with faster training speed (samples processed per second) and less tuned parameters.

			Mathematical Tasks								Generic Tasks			
			GSM8K		AddSub		SingleEq		SVAMP		MMLU		CSQA	
Models	**Train Speed**	**Tuned Params.**	**Acc.**	**Δ**	**Acc.**	**Δ**	**Acc.**	**Δ**	**Acc.**	**Δ**	**Acc.**	**Δ**	**Acc.**	**Δ**
LLaMA2-7B	-	-	14.6	-	30.5	-	65.4	-	34.7	-	46.0	-	59.8	-
+ Full SFT	15sam./sec.	6.7B	24.6	+10.0	53.7	+23.2	68.2	+2.8	50.3	+15.6	40.5	-5.5	54.0	-5.8
+ Precise SFT	50sam./sec.	0.07B	27.4	+12.8	50.6	+20.1	69.7	+4.3	55.8	+21.1	46.4	+0.4	59.6	-0.2
LLaMA2-13B	-	-	28.7	-	33.7	-	76.6	-	45.7	-	54.8	-	67.3	-
+ Full SFT	8sam./sec.	13.0B	44.6	+15.9	62.2	+28.5	79.8	+3.2	62.8	+17.1	50.2	-4.6	62.0	-5.3
+ Precise SFT	34sam./sec.	0.08B	46.3	+17.6	61.1	+27.4	82.2	+5.6	66.6	+20.9	55.0	+0.2	67.2	-0.1

Conclusion

- 수학 능력을 담당하는 Attention Head들과 이 AH들이 활성화 시키는 MLP Layer의 뉴런들은 확실히 존재함
- 각 AH가 담당하는 연산은 다름
- 첫 논문에서 증명되었듯이, 이 MLP Layer의 뉴런들이 실질적으로 수학 능력을 담당하는 뉴런

- CNA와 Path Patching 둘다 다른 방법이지만 수학 능력을 담당하는 뉴런을 찾을 수 있는 방법

- 두 논문 다 2024 논문인데 왜 Llama3와 같은 더 최근 모델은 안 했을까
  - 첫 논문은 Llama-7B/GPT-J 두번째 논문은 Llama2-7B/Mistral-7B로 실험함

Thank you