大規模言語モデルの数値系列解釈メカニズムの解明に向けて：プロンプトの有無が順位情報表現に与える影響

新井深月 [1,2] 石垣達也 [2] 宮尾祐介 [3,2] 高村大也 [2] 小林一郎 [1,2]
[1] お茶の水女子大学 [2] 産業技術総合研究所 [3] 東京大学
{g2120503, koba}@is.ocha.ac.jp
{ishigaki.tatsuya, takamura.hiroya}@aist.go.jp
yusuke@is.s.u-tokyo.ac.jp

概要

大規模言語モデル（LLM）は，算術演算や数値的推論を含む多様な数学的タスクにおいて高い性能を示している．一方で，これらの能力がモデル内部のどのような表現や計算過程に基づいて実現されているのかについては明らかになっていない．特に，数値が系列として与えられる場合に，モデルが入力系列のどの位置に注目し，タスクに必要な情報を記憶しているのか調査されていない．本研究は数値系列を入力としたタスクを対象にプロンプト有無の違いがトランスフォーマーの隠れ状態ベクトルに与える影響を分析する．具体的には，各層における Key，Query，Value，および残差ストリームを抽出し，線形プロービングを用いて出力に必要な情報が読み出し可能となるか検証する．分析の結果，最大値の位置情報はプロンプトの有無に関わらず中間層以降で高精度に読み出し可能である一方，第 2・第 3 位に大きい値の情報はプロンプトを与えても抽出が困難であることが明らかになった．本研究の結果は，LLM における数値系列の処理が，モデルの事前学習によって形成された固有の内部表現に強く依存していることを示唆する．

1 はじめに

近年，大規模言語モデル（LLM）は自然言語生成にとどまらず，算術演算や数学的推論，数値時系列処理といった高度な数値解釈が必要なタスクにおいても高い性能を示している [1, 2]．このような能力の拡張は，LLM が数値を含む入力に対して，単なる表層的なパターン認識を超えた処理を内部で行っている可能性を示唆している．しかし，モデルが高い正解率を示す場合であっても，その回答がどの隠れ状態ベクトルや計算過程に基づいて生成されているのか，入出力のみから判断することは困難である．本研究では，数値系列を入力とするタスクを対象に，プロンプト有無という条件の違いがトランスフォーマーの隠れ状態ベクトルに与える影響を図 1 の流れで分析する．タスク指示を明示的に与えるか否かというプロンプト条件の違いは，モデルが注目すべき情報や計算の進め方に影響を与えると考える．トランスフォーマーの各層における Key，Query，Value，および残差ストリームを抽出し，線形プロービングを用いてタスクの回答に必要な情報がどの入力位置・どの層で読み出し可能となるかを隠れ状態ベクトルに与える影響分析を通じて，LLM の数値系列解釈メカニズム解明に向けた知見を提供する．

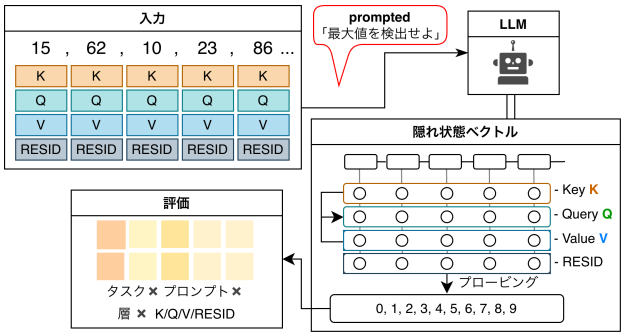

図 1 本研究の概要．数値系列解釈タスクにおける分析フローを示す．

2 関連研究

LLM の内部挙動を理解することを目的とした研究は，近年活発に行われており，特にモデル内部の計算過程を因果的に分析するメカニズム解釈可能性（Mechanistic Interpretability: MI）の分野において多くの成果が報告されている [3, 4]．これらの研究では，トランスフォーマーをアテンションヘッドや

残差ストリームからなる計算構造として捉え，特定のタスクに寄与する内部回路を同定する試みが行われている [5]．一方で既存研究の多くは，自然言語理解や記号的推論，あるいは算術演算を対象としており，時系列データを入力とした場合の隠れ状態ベクトルや計算過程に注目した分析は少ない．

隠れ状態ベクトルにどのような情報が符号化されているかを分析する手法として，線形プロービングは広く用いられている [6]．線形分類器を用いて特定の属性が予測可能かを検証することで，その属性が隠れ状態ベクトルから読み出し可能な形で存在するかを評価できる．実際に数や曜日の周期的な構造が読み出せることが報告されており [7, 8]，プロービングは隠れ状態の分析において有効な手法であることが示されている [9]．以上を踏まえ，本研究は，数値系列を入力とするタスクを対象に，プロンプトの有無というタスク条件の違いが隠れ状態ベクトルに与える影響を分析する．

3 プロービング

3.1 入力数値の設定

入力数値を 0 から 999 の整数に限定した．この範囲の数値は使用モデルにおいて単一トークンとして表現されるため，トークンと隠れ状態ベクトルが一対一対応になるように設定されている．

3.2 データセット

実験には，人工的に生成した数値系列を用いる．各数値系列は長さ 10 の整数列とし，各要素は 0 から 999 の範囲で一様分布からサンプリングする．生成した各数値系列に対して，1) **最大値（Max）**，2) **2 番目に大きい値（2nd Max）**，3) および **3 番目に大きい値（3rd Max）**の位置を正解ラベルとして付与する[1)]．位置は数値系列中のインデックス（0 から 9）として定義し，10 クラス分類問題として扱う．

トークン位置 t	prefix	正解ラベル
0	$\langle 3 \rangle$	0
1	$\langle 3, 1 \rangle$	0
2	$\langle 3, 1, 4 \rangle$	2
3	$\langle 3, 1, 4, 2 \rangle$	2

表 1 [3, 1, 4, 2] を入力した場合の Max タスクにおけるトークン位置ごとの正解ラベルの例．

1) 正解ラベルの一意性を保つため，生成された数値系列内に同一の数値が含まれる場合は，当該系列をデータセットから除外した．

3.3 プロンプト条件

入力条件として以下の 2 種類を設定する：

- raw：数値系列のみの入力
- prompted：対象タスクを明示した指示文を数値系列の前に付与した入力

これらの比較によりプロンプトの有無が隠れ状態ベクトルに与える影響を調査する．

3.4 タスク設計

数値系列を入力としたタスクとして，最大値の位置を予測する Max，2 番目に大きい値の位置を予測する 2nd Max，および 3 番目に大きい値の位置を予測する 3rd Max の 3 種類を設定する．本研究では，モデルが数値系列を逐次的に処理しながら内部表現を更新していく過程を分析するため，数値系列を読み進めた時点で得られる情報のみに基づいて最大値の位置を予測する設定としている（正解ラベルの定義例を表 2 に示す）．予備実験において，Max は，タスク指示を与えない条件（raw）においても高い精度を示し，プロンプトの有無による差がほとんど観測されなかった．このため，より特殊なタスクとして 2nd Max および 3rd Max を追加した．各タスクにおける具体的なプロンプトを表 2 に示す．なお，最大値の位置を予測する設定を採用した背景として，最大値そのものの値を回帰するタスクについても予備実験を行った．しかし，最大値の値を直接回帰させる設定では，プロービングモデルの学習が十分に進まず，安定した性能を得ることができなかった．この結果からモデルは数値系列の最大値をその数値として保持しているというよりもどの位置に最大値が存在するかという情報を中心に符号化している可能性が示唆された．よって本研究では，最大値の「位置」を予測対象とするタスクを主として分析する．

3.5 使用モデル

実験には，Llama-3.1-8B-Instruct を用いる．推論には Hugging Face トランスフォーマーライブラリを使用し，モデルの重みおよびトークナイザは公開されている設定をそのまま使用した．

表 2　プロンプト条件ごとの入力例

条件	入力例
raw	`Sequence: [123, 45, 678, 9, 234, 56, 890, 12, 345, 67]`
prompted (Max)	`Which is the value that is the Maximum?` `Sequence: [123, 45, 678, 9, 234, 56, 890, 12, 345, 67]`
prompted (2nd Max)	`Which is the value that is the second largest?` `Sequence: [123, 45, 678, 9, 234, 56, 890, 12, 345, 67]`
prompted (3rd Max)	`Which is the value that is the third largest?` `Sequence: [123, 45, 678, 9, 234, 56, 890, 12, 345, 67]`

3.6　隠れ状態ベクトルの抽出

モデルの前向き計算中に，各トランスフォーマー層における Key ベクトル，Query ベクトル，Value ベクトルおよび残差ベクトルを抽出する．分析対象は入力数値系列中の数値トークンに対応する位置のみとし，プロンプト文や記号に対するトークン位置は分析から除外する．

3.7　プロービング手法と評価

各層および各隠れ状態ベクトルタイプごとに線形分類器を学習し，正解ラベル（0 から 9）を予測する．線形分類器を用い，評価指標には正解率を使用する．分類器の学習設定は付録 A に示す．

4　実験結果

線形プロービングによる実験結果を，1) タスク種別，2) プロンプト条件，3) 層別の挙動，4) 隠れ状態ベクトルの種類の観点から整理して報告する．

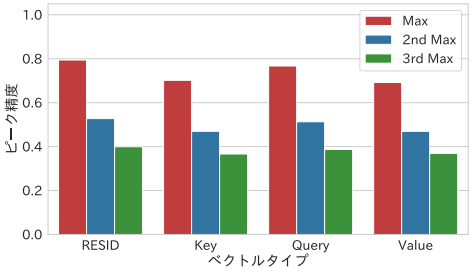

図 2　隠れ状態ベクトルタイプ別のピーク精度の比較．各タスク（Max, 2nd Max, 3rd Max）で得られた，層ごとの最大正解率を示す．

4.1　タスク別のプロービング精度

タスク種別ごとのプロービング精度を比較した結果，Max タスクにおいて 0.80 程度の高い正解率が得られた（図 2）．特に残差ベクトルからは，最大値位置を高精度で予測可能であった．一方，2nd Max および 3rd Max タスクでは，Max タスクと比較して全体的に正解率が低下した．順位が下がるにつれて難易度が高くなる傾向が見られ，3rd Max タスクでは正解率が 0.40 以下まで精度が低下した．2nd Max および 3rd Max タスクの層別・ベクトルタイプごとの詳細な結果は付録 B に示す．さらに，利用可能な情報量が異なるためトークン位置によって分類難易度が異なると考えられる．よって，位置別に精度を算出した結果を付録 C に示す．

4.2　プロンプトの影響

raw 条件と prompted 条件を比較した結果，すべてのタスクで両条件間で顕著な精度差は観測されなかった（図 4）．raw 条件においても prompted 条件と同程度の正解率が得られている箇所が多く存在した．さらに明示的な指示を与えない raw 条件の方が，prompted 条件より精度が高い傾向が見られた．

4.3　層別の挙動

層ごとのプロービング精度を分析した結果，多くのタスクにおいて中間層で最も高い正解率が得られる傾向が確認された（図 9）．初期層では正解率が低く，層が進むにつれて精度が上昇し，中間層付近でピークを迎える場合が多かった．この傾向は raw 条件，prompted 条件に共通して観測された．

4.4　隠れ状態ベクトル間の比較

全ての設定において，残差ベクトルから最も高いプロービング精度が得られた．Key，Query，Value ベクトルからのプロービング精度は残差ベクトルと比較して低い傾向にあり，全タスクおよび両入力条件において共通して観測された．

5　考察

実験前に立てた仮説を整理した上で，得られた結果がその仮説とどのように異なっていたかを述べ，結果に基づく分析を行う．

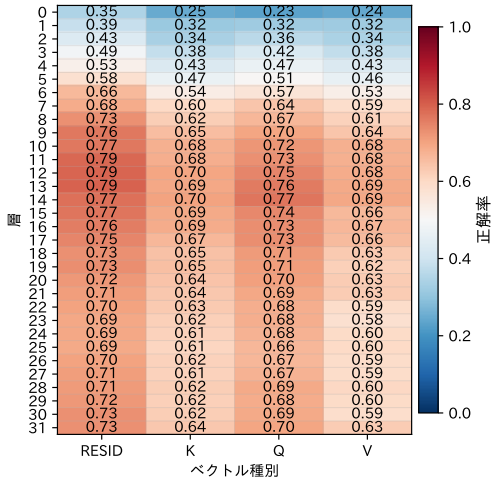

図 3　Max タスクにおける線形プロービング精度のヒートマップ（raw 条件）．

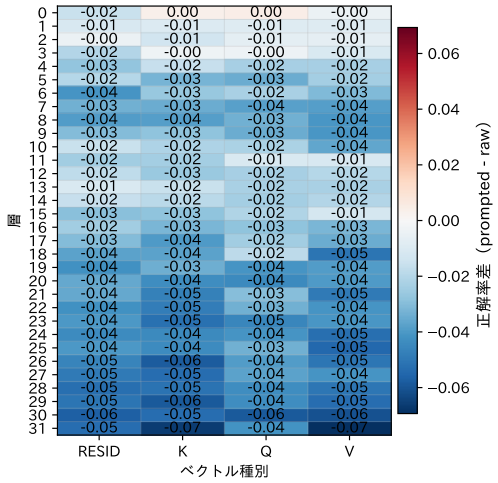

図 4　Max タスクにおける条件間の精度差のヒートマップ（prompted – raw）．正の値は prompted 条件の方が精度が高いことを示す．

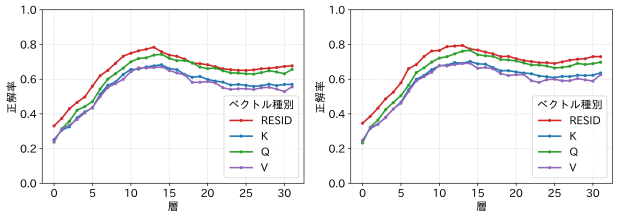

図 5　Max タスクにおける層ごとのプロービング精度の推移．上に raw 条件，下に prompted 条件の結果を示す．

5.1　事前仮説

本研究では，タスクの特殊性に応じてプロンプト効果が異なると仮定していた．具体的には，Max タスクについては raw 条件においても高い精度が得られる一方で，2nd Max や 3rd Max のようなあまり与えられることのないタスクでは，指示を明示する prompted 条件において raw 条件よりも高いプロービング精度が得られると予想していた．この仮説は「最大値の情報はモデル内部に自然に符号化されている一方で，2 番目以降の順位情報は，明示的な指示が与えられた場合にのみ内部表現として強化される」という考えに基づくものであった．

5.2　仮説との不一致

実験結果は仮説を支持しなかった．Max タスクにおいては予想通り，raw 条件と prompted 条件の間に一貫した差が見られなかったが，2nd Max および 3rd Max タスクにおいても，prompted 条件による精度向上は確認されなかった．つまり，プロンプトによってタスク内容を明示したにも関わらず，順位が下がるにつれて低下する精度の傾向は改善されず，raw 条件との差は限定的であった．この結果は，「プロンプトによって内部表現がタスク特化的に変化する」という想定が，少なくとも本設定においては成り立たないことを示している．

5.3　結果に基づく解釈

プロンプトがモデルの出力挙動に影響を与えうる一方で，指示に対応する情報が数値トークンの隠れ状態ベクトルにどのように符号化されているかという観点では，必ずしも表現の可読性を高める方向には作用しない可能性を示唆している．また，raw 条件でも prompted 条件と同程度の精度を示したことに関しては，明示的な指示なしでも最大値に対応する情報が安定して符号化されていることを示している．

6　おわりに

数値系列を入力としたタスクを対象に，プロンプトの有無が LLM 内部の隠れ状態ベクトルに与える影響を線形プロービングを用いて分析した．実験の結果，最大値位置に関する情報は raw 条件下の方が prompted 条件下より高精度で読み出し可能であることが確認された．本研究の結果はプロンプトがモデルの出力を制御する役割を果たす一方で，数値系列に対応する隠れ状態ベクトルの情報構造を必ずしもタスク特化的に再構成するとは限らないことを示している．すなわち数値系列に関する内部理解は明示的な指示に依存せず，暗黙的に形成されている可能性がある．この知見は数値系列処理における LLM の内部機構を理解する上で重要な示唆を与える．

謝辞

この成果は，産総研政策予算プロジェクト「フィジカル領域の生成 AI 基盤モデルに関する研究開発」の結果得られたものである．

参考文献

[1] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In **Advances in Neural Information Processing Systems**, Vol. 35, pp. 24824–24837, 2022.

[2] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In **Advances in Neural Information Processing Systems**, Vol. 36, 2024.

[3] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. **Transformer Circuits Thread**, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[4] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In **The Eleventh International Conference on Learning Representations**, 2023.

[5] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. **Transformer Circuits Thread**, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[6] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018.

[7] Tianyi Lan, Neel Nanda, et al. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. In **Proceedings of EMNLP**, 2024.

[8] Amit Arnold Levy and Mor Geva. Language models encode numbers using digit representations in base 10. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, **Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)**, pp. 385–395, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[9] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

A　線形プロービングの学習設定

線形プロービング実験における分類器の学習設定の詳細を表 3 に示す．

表 3　線形プロービングにおける分類器および学習設定

項目	設定内容
分類器	線形分類器
エポック数	100
データ	訓練: テスト = 0.7 : 0.3
入力	各層・各隠れ状態ベクトル
出力	数値系列中の位置（10 クラス，0–9）

B　タスク別のプロービング精度

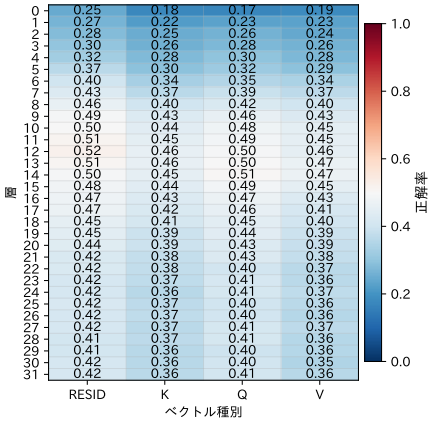

図 6　2nd Max タスクにおける線形プロービング精度のヒートマップ（prompted 条件）．

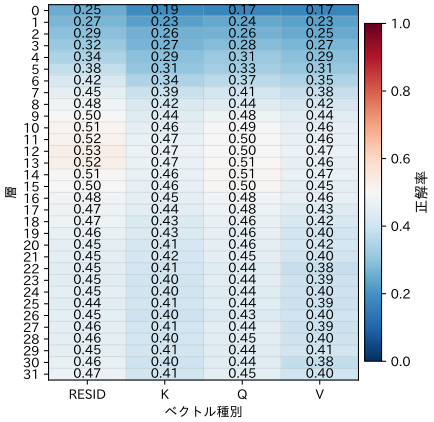

図 7　2nd Max タスクにおける線形プロービング精度のヒートマップ（raw 条件）．

C　トークン位置に着目した追加分析

Max タスクについて，トークン位置 t における内部表現は，系列の先頭から位置 t までの prefix のみに基づいて計算される．このため，トークン位置によって内部表現が利用できる情報量が異なり，分類難易度は変化すると考えられる．図 10 に，トークン位置ごとのプロービング精度の変化を示す．

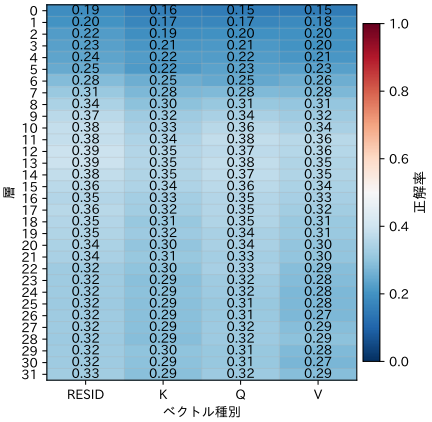

図 8　3rd Max タスクにおける線形プロービング精度のヒートマップ（prompted 条件）．

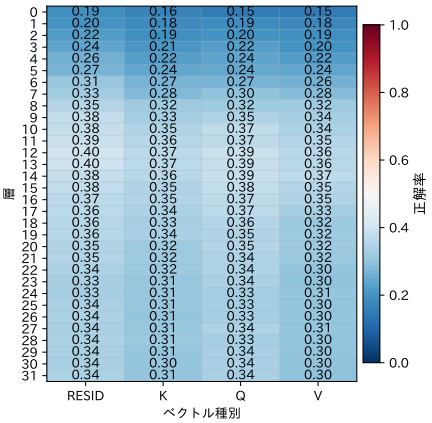

図 9　3rd Max タスクにおける線形プロービング精度のヒートマップ（raw 条件）．

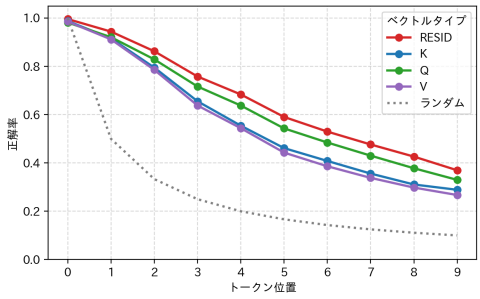

図 10　Max タスクにおける，ベクトルタイプ（RESID, K, Q, V）ごとのトークン位置別プロービング精度．