算術タスクを用いた文脈内学習による外挿能力の分析

進藤 稜真 [1] 竹下 昌志 [1] ジェプカ・ラファウ [2] 伊藤 敏彦 [2]
[1] 北海道大学大学院 情報科学院 [2] 北海道大学大学院 情報科学研究院
{shinto.ryoma, takeshita.masashi.68}@gmail.com
{rzepka, t-itoh}@ist.hokudai.ac.jp

概要

大規模言語モデルは文脈内学習 (ICL) が可能であることが知られているが, その内部メカニズムについては未だ統一的な見解が存在しない. 争点の一つに, ICL によってパラメータを更新することなく未知のタスクに適応可能か, すなわち外挿可能かどうかという点がある. そこで本研究では, 二変数一次関数 $z = ax + by$ に基づく算術データセットを構築し, ICL による内挿・外挿能力を定量的に評価した. その結果, ICL によって部分的に未学習のタスクでも解決可能であること, 文脈内の例を増加させることで, 内挿時と外挿時の内部表現が類似する傾向にあること, 学習データ内のタスクの多様性が ICL 能力の創発に重要であることが示唆された.

1 はじめに

大規模言語モデル（Large Language Model, 以下 LLM）は, 文脈内学習（In-Context Learning, 以下 ICL）[1, 2] が可能であることが知られている. ICL とは, プロンプト内にタスクの例を提示することで, パラメータを更新することなく推論性能を向上させる手法であり, 学習データの準備や追加の計算資源が必要ないため, 効率的で柔軟な応用が可能である [3, 4].

ICL のメカニズムに関しては, 文脈内の例からタスクの特性を認識し, 事前学習済みのタスクを選択・適用しているという仮説 [5] や, 選択だけでなく, 複数の学習済みタスクを組み合わせて推論が可能であるという仮説 [6] が提起されている. さらに, ICL はタスクの学習方法そのものを学ぶことが可能であり, 文脈内の例を基に未学習のタスクに適応できるという仮説もある [7]. しかし, これらの仮説は必ずしも実験結果と整合せず [8], ICL のメカニズムに関する統一的な理解は未だ得られていない.

特に, ICL によって未学習タスクに適応可能であるかどうかは, これらの仮説に説得的な根拠, あるいは反例を与えるため重要な争点となっている [9, 10]. しかし, 大規模な言語データを学習する場合, 学習済みのタスク (内挿) と完全に未学習のタスク (外挿) の境界を明確に定義することは現実的でなく, ICL による外挿能力を厳密に評価することは困難である.

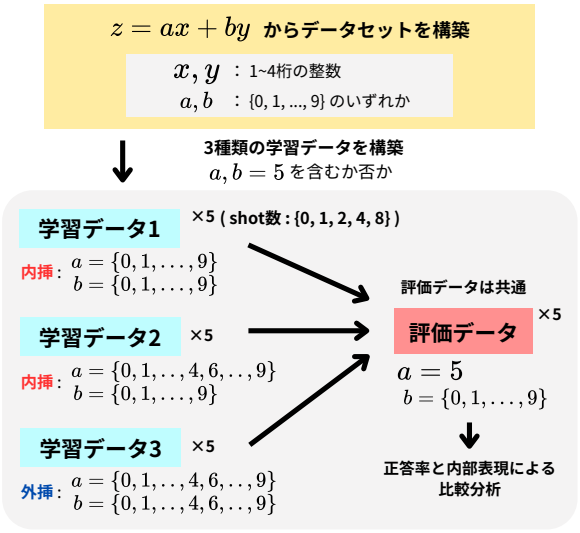

図 1 ICL の外挿能力分析のためのデータセットの構築方法と評価の流れ. データセットは $z = ax + by$ に基づいて構築する. 学習データは係数 $a, b = 5$ を含むか否かによって 3 種類に分類される. 評価データは $a, b = 5$ を含むため, 学習データに含まれる a, b の範囲によって, 内挿と外挿が定義される. 学習データ 1〜3 はさらに shot 数によって 5 種類に分類され, 学習データは合計 15 種類となる.

そこで本研究では, 算術タスクを用いて ICL の外挿能力を分析することで, ICL のメカニズムに関して重要な知見を与えることを目的とする. 算術タスクは, 数字の桁数や変数の範囲を制御することで, 内挿と外挿の領域を明確に分離できるという利点がある. さらに, 本タスクの推論には文脈内の例が不可欠であるように設計されており, タスクが解決された場合には ICL が行われたことを保証できる.

実験では, 計 15 種類のデータセットを独自に構築し, それぞれを学習したモデルの評価データ正解率と, 埋め込み表現の比較分析を通して ICL による外

挿能力の分析を行った. その結果, 次のような知見が得られた. (i) ICL によって, 学習済みタスクを組み合わせて新たなタスクを解決可能であること. (ii) shot 数を増加させることで, 内挿時と外挿時の内部表現が類似する傾向にあること. (iii) 学習データ内のタスクの多様性が ICL 能力の創発に重要であること. 本研究で得られた知見は, ICL のメカニズム理解へ大きく貢献するものである.

2 実験設計

2.1 データセットの構築

データセットは, 二変数一次関数 $z = ax + by$ に基づいて構築した (図 1 参照). x および y は 1 桁から 4 桁の整数であり, a および b は 0 から 9 の 1 桁の整数である. 各データ内の 1 つの計算式は, (x, y, z) の形でモデルへ入力され, 係数 a, b は明示されない.

$a = 2, b = 1$ で 2shot の場合の例

```
input:(132,5532,5796),(355,22,732),(4412,3356,
output:12180
```

input 内の各計算式では a , b が共通であり, x , y はランダムに生成される. したがって, x , y の値のみからは z の値が一意に定まらず, モデルは ICL によって例に共通する a, b を特定した上で z を推論する必要がある. この設計により, 本データセットを適切に学習したモデルは, 本タスクの推論時に ICL を行っていることが保証されるため, ICL による外挿能力について分析が可能となる.

2.1.1 データセットの構成

学習データは計 200,000 件 (訓練：検証 = 8:2), 評価データは計 1,000 件である. 各データの a, b は事前に定義された範囲（表 1 参照）からランダムに決定される. 本実験では, $a, b = 5$ を内挿と外挿を分離する基準とし, 学習データセットは $a, b = 5$ を含むか否かによって 3 種類に分類される. さらに, これらは input に含まれる計算例の数 (shot 数) によって, 0,1,2,4,8-shot の 5 通りに分けられ, 合計 15 種類の学習データセットが存在する.

2.2 モデルと評価

本実験では, ByT5 base [11] を使用して学習を行った. ByT5 は数字を必ず一文字単位でトークン化するため, 二桁以上の数字も分割されずに一意にトークン化できる. この一意性により, a, b の範囲に基づいて構築した学習データにおいて, トークナイザに関わらず内挿と外挿の領域が厳密に区別可能となる. 評価では検証データの損失が最も低いモデルを用いて評価データの正解率を算出した. また, 学習中[1)]は 1,000 ステップごとに検証データと評価データ 1,000 問の正解率推移を記録した.

表 1 各データセットの係数 a, b の範囲

データ	a の範囲	b の範囲
学習データ 1	$a \in \{0, .., 5, .., 9\}$	$b \in \{0, .., 5, .., 9\}$
学習データ 2	$a \in \{0, .., 4, 6, .., 9\}$	$b \in \{0, .., 5, .., 9\}$
学習データ 3	$a \in \{0, .., 4, 6, .., 9\}$	$b \in \{0, .., 4, 6, .., 9\}$
評価データ	$a = 5$	$b \in \{0, .., 5, .., 9\}$

2.3 モデル間の内部表現の類似度比較

正解率のみでは評価できないモデル内部を解析するため, 内挿・外挿時の各モデル間の内部表現を比較分析する. そこで, 以下の 3 種類の分析データ 1,000 件 ($\{d_i\}_{i=1}^{N}, (N = 1000)$) を新たに作成した. 分析データ 1 は全モデルにとって内挿, 分析データ 2 は学習データ 1, 2 を, 分析データ 3 は学習データ 1 を学習したモデルにとってのみ内挿となる.

- 分析データ 1: $a \neq 5, b \neq 5$
- 分析データ 2: $a \neq 5, b = 5$
- 分析データ 3: $a = 5, b = 5$

まず, 各データ推論時におけるエンコーダ最終層の出力ベクトル行列（データ件数×ベクトル次元数）を取得する. モデルが 3 種類あるため, 計 9 種類の出力ベクトル行列が得られる. 次に, それぞれの出力ベクトル行列において, 各データペア (d_i, d_j) に対応する出力ベクトル $\mathbf{h}_i, \mathbf{h}_j$ のコサイン類似度を

$$\text{cos-sim}(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \|\mathbf{h}_j\|}$$

により算出し,データ件数 × データ件数 の類似度行列 $\mathbf{S} \in \mathbb{R}^{N \times N}$ を得る. この行列は, 分析データセット内の各データ推論時に, そのモデルがどの程度類似した内部表現を持つかを示すものである.

最後に, 分析データ $\ell (\ell = 1, 2, 3)$ における各モデルの類似度行列を $\mathbf{S}_{\ell 1}, \mathbf{S}_{\ell 2}, \mathbf{S}_{\ell 3}$ とし, モデル $m, n (m, n = 1, 2, 3)$ 間のスピアマン相関係数 [12] を

$$r_{\ell,mn} = \text{SpearmanCorr}(\text{vec}(\mathbf{S}_{\ell m}), \text{vec}(\mathbf{S}_{\ell n}))$$

で計算する. この $r_{\ell,mn}$ は, 分析データ ℓ 推論時の, モデル m と n がどの程度類似した内部表現構造を持つ傾向にあるかを定量的に表す指標となる.

1) ハイパーパラメータの詳細は付録 A.1 を参照

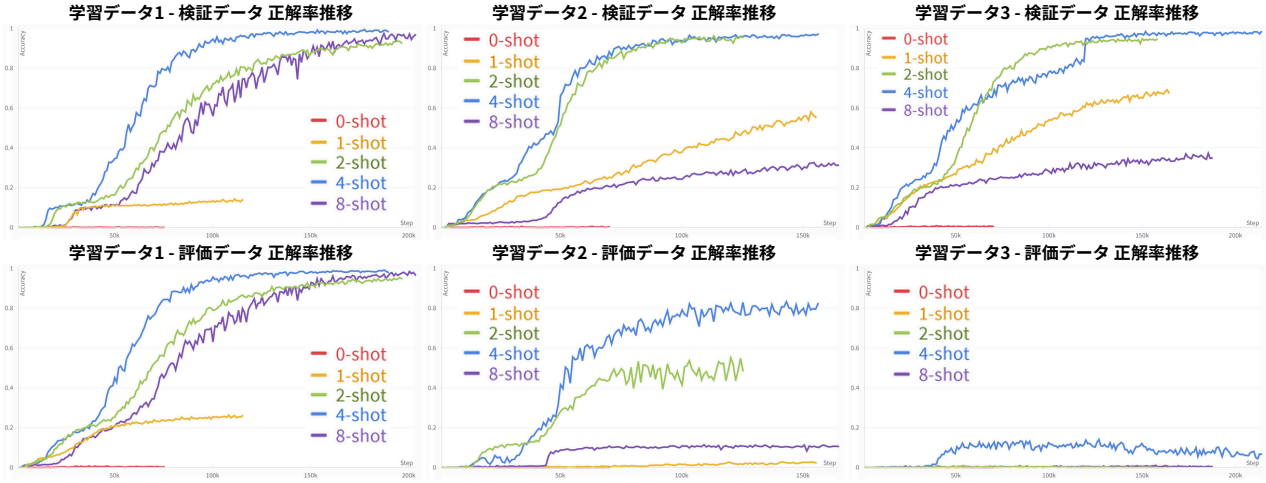

図 2　各データセットを学習したモデルの検証データ (上段) と評価データ (下段) 正解率推移.

3　実験結果

3.1　正解率の結果

図 2 に, 各データセットを学習したモデルの検証データ（上段）と評価データ（下段）の正解率推移を示す. また, 表 2 には, 最も検証データの損失が低い時点における評価データ正解率を示す.

検証データの結果　検証データの正解率推移を示した図 2 上段に共通する傾向として, 0-shot や 1-shot の場合は正解率が伸び悩む一方, 2-shot および 4-shot では正解率が 1 に近づく結果が見られた. 8-shot の場合, 学習データ 1 (左列) では検証データの正解率が 1 に収束したものの, 学習データ 2 (中央列) および 3 (右列) では正解率が向上しなかった. これらの結果は, shot 数が 2 または 4 の場合のみにおいて, ICL によって本タスクを解決できたことを示している.

評価データの結果 - 学習データ 1　図 2 左列下段のグラフからは, 2,4,8-shot の場合に, 評価データ正解率が 1 に収束する学習過程が確認できる. 一方, 0-shot の場合は正解率 0.002, 1-shot の場合は 0.116 にとどまった (表 2 参照).

評価データの結果 - 学習データ 2　図 2 中央下段のグラフからは, 2-shot では正解率 0.5 付近, 4-shot では 0.8 付近に収束している様子が確認できる. 一方, 0,1-shot では正解率はほとんど 0 であり, 8-shot においても 0.1 程度にとどまった (表 2 参照).

評価データの結果 - 学習データ 3　表 2 から, ほとんどの shot 数で正解率が 0 であることが確認できる. 唯一, 4-shot では学習初期段階で正解率が 0.1 程度に達し (図 2 の右列下段を参照), その後徐々に正解率が下がる様子が見られた. 最終的には正解率は 0.066 となった (表 2 参照).

表 2　各データで学習したモデルの評価データ正解率

データセット	0shot	1shot	2shots	4shots	8shots
学習データ 1	0.002	0.116	0.936	0.979	0.971
学習データ 2	0.000	0.015	0.473	0.825	0.105
学習データ 3	0.000	0.000	0.000	0.066	0.008

3.2　内部表現による結果の比較

上記の結果より, 各学習データを通して ICL を用いて本タスクを解決したことが保証できるのは 2-shot と 4-shot 設定のみであるため, ICL 時の内部表現の比較結果は 2-shot と 4-shot の場合のみに限定する. 図 3 は, 出力ベクトル (エンコーダ最終層) のデータ同士のコサイン類似度行列を取得し, 各モデル間の相関係数を視覚化したものである. ここで, 学習データ 1,2,3 を学習したモデルを, それぞれモデル 1,2,3 と呼ぶとする. このとき例えば, 図 3 左列 1 段目の表からは, モデル 1,2,3 に分析データ 1 を推論させた時の内部表現が, 各モデル間でどれだけ類似しているかの傾向を知ることができる.

2-shot　2-shot (左列) では, どの分析データにおいてもモデル 1 と 2 が高い相関 (約 0.7) を示したことから, モデル 1 と 2 の推論時の内部表現が類似している傾向が示された. 一方, モデル 3 はモデル 1, 2 との相関係数は 0.2 ～ 0.4 台にとどまった.

4-shot　4-shot (右列) では, モデル 3 とモデル 1,2 の相関係数が 2-shot の場合と比べて著しく向上した. この結果は, 分析データに関係なく全体の傾向として見られることが図 3 に示されている.

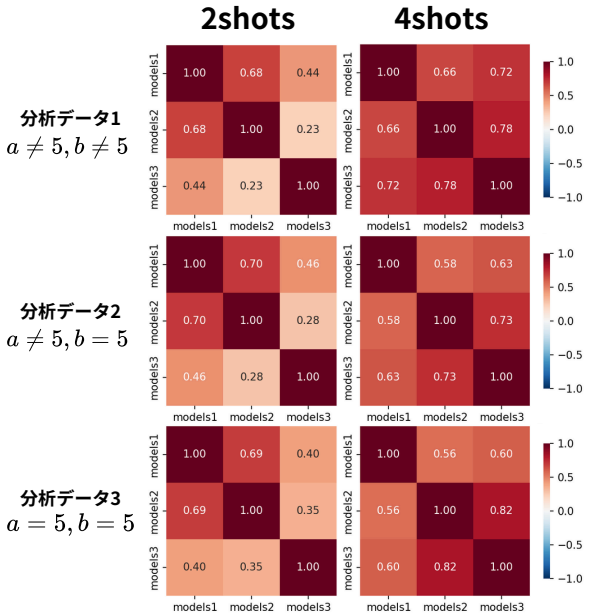

図 3　出力ベクトルのコサイン類似度行列から算出したスピアマン相関係数. models1,2,3 はそれぞれ学習データ 1,2,3 を学習したモデルを指す.

4　考察

ICL による外挿能力　以下, (a, b) の組み合わせをタスクと呼び, 議論を進める. まず, 学習データ 1 に関して, 2,4,8-shot の場合に検証データと評価データの正解率が約 1 を記録した (表 2 参照). 学習データ 1 は評価データのタスク範囲を含む (内挿) ので, この結果からは, ICL によって提示されたタスクを認識し, 学習済みタスクの中から適切に選択・適用可能であることが示唆される. 次に, 学習データ 2 は $a = 5$ を含まない, すなわち評価データと同様のタスクを学習していないにもかかわらず, 評価データにおいて 2-shot で 0.473, 4-shot で 0.825 の正解率を達成した (表 2 参照). これは, ICL によって学習済みのタスクを選択しているだけでなく, タスクを組み合わせることで, 部分的に未知のタスクを解決し得ることを示唆している. 具体的には, 学習データ 2 に含まれる $b = 5$ タスク, すなわち $(a, b) = (k, 5)$ $(k \in \{0, .., 4, 6, .., 9\})$ という形式のタスクを学習していたため, 評価データのタスク解決に繋がったと考えられる. 最後に, 学習データ 3 の評価データに対する正解率は, 4-shot を除きほとんど 0 であり, 本実験の範囲においては ICL による外挿能力に限界があることが示唆された. しかし, 4-shot において, 学習データ 3 の範囲外である評価データに正解している場合もある (正解率 : 0.066, 1,000 問中 66 問) ことは注目に値する. さらに, 図 3 からは, shot 数を増やすことで, 外挿時のモデルの内部表現が内挿時のモデルと類似する傾向が見られた. 例えば, 図 3 の最下段の表は, モデル 1 にとっては内挿, モデル 3 にとっては外挿である分析データ 3 において, 2-shot の場合と比べて 4-shot では内部表現の類似度が高まる傾向が見られた. この結果からは, 正解率には反映されなかったものの, shot 数の増加に伴い外挿可能となる可能性が示唆され, 実験条件の修正によってさらなる検証が必要だと考えられる.

タスクの多様性の重要性　図 2 や表 2 の結果によると, 8-shot では正解率が 1 に収束した一方で, 学習データ 2 および 3 では正解率が伸び悩む結果となった. この差異の要因として, 学習データ内に含まれるタスクの多様性が ICL 能力の創発にとって重要である可能性が考えられる. 本研究のタスク設計では, 学習データ 1 は a, b がそれぞれ 0 から 9 までの 10 通りであり, 計 100 通りのタスクがある. 一方, 学習データ 2 は $a = 5$ を除外して 90 通り, 学習データ 3 はさらに $b = 5$ を除外して 81 通りとなっている. タスクの多様性が ICL 能力に及ぼす影響について, 実際に学習データの多様性を調整することで評価データの正解率が大きく変化することを確認した (付録図 4 参照). これは, 学習データ内のタスクの多様性が高いことで, モデルは各タスクに個別の解決方法を学習するするのではなく, 各タスクに汎用的に適用できる解決方法を獲得する可能性が考えられ, ICL の外挿能力について議論するにはさらなる検証が必要になることが示唆された.

5　おわりに

本研究では, 算術タスクを用いて LLM の ICL による外挿能力を分析した. 正解率と内部表現から分析を行った結果, 以下の知見が得られた. (i) ICL によって, 学習済みのタスクを組み合わせてタスク解決が可能であること. (ii) shot 数を増加させることで, 内挿時と外挿時の内部表現が類似する傾向にあること. (iii) 学習データ内のタスクの多様性が ICL 能力の創発に重要であること.

今後は, 三変数一次関数など, タスクの多様性をさらに確保し, shot 数を増やした場合の外挿能力を詳細に検証することで, より有用な実験的知見を提示できると考えられる. これにより, 未だ統一的な見解が確立していない ICL のメカニズム理解に, 本研究が大きく貢献することが期待される.

謝辞

本研究は JST CREST JPMJCR20D2 の助成を受けたものです.

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.

[2] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 1107–1128, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[3] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 12284–12314, Toronto, Canada, July 2023. Association for Computational Linguistics.

[4] Qingyu Yin, Xuzheng He, Chak Tou Leong, Fan Wang, Yanzhao Yan, Xiaoyu Shen, and Qiang Zhang. Deeper insights without updates: The power of in-context learning over fine-tuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 4138–4151, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[5] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In **International Conference on Learning Representations**, 2022.

[6] Jiaoda Li, Yifan Hou, Mrinmaya Sachan, and Ryan Cotterell. What do language models learn in context? the structured task hypothesis. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 12365–12379, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[7] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In **Proceedings of the 40th International Conference on Machine Learning**, ICML'23. JMLR.org, 2023.

[8] Jannik Kossen, Yarin Gal, and Tom Rainforth. In-context learning learns label relationships but is not conventional learning. In **The Twelfth International Conference on Learning Representations**, 2024.

[9] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In **Proceedings of the 36th International Conference on Neural Information Processing Systems**, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[10] Tianyu He, Darshil Doshi, Aritra Das, and Andrey Gromov. Learning to grok: Emergence of in-context learning and skill composition in modular arithmetic tasks. In **The Thirty-eighth Annual Conference on Neural Information Processing Systems**, 2024.

[11] Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. **Transactions of the Association for Computational Linguistics**, Vol. 10, pp. 291–306, 2022.

[12] C. Spearman. The proof and measurement of association between two things. **The American Journal of Psychology**, Vol. 15, No. 1, pp. 72–101, 1904.

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **International Conference on Learning Representations**, 2019.

A　参考情報

A.1　学習設定

本実験において, ハイパーパラメータは次のように設定した.

- optimizer : AdamW [13]
- learning rate ： 0.0001
- batch size ： 64
- epochs : 100

また, 検証データの損失が 30,000 ステップ以内に改善しない場合, 過学習を防ぐため早期終了 (early stopping) を適用した.

A.2　データの多様性の ICL における影響

A.2.1　データセット構築

本実験タスクにおいて, (a, b) の組み合わせは最大 100 通り存在する. ICL による外挿能力の分析のためには, このデータ内のタスクの多様性によって, ICL 能力の創発にどのような影響が存在するかを検証しておく必要がある. そこで, 以下のように学習データ 2 において, a, b の取る値によって 4 種類のデータセットを構築し, これらを本実験と同様の設定で学習させた.

- 学習データ 2-1：$a \in \{0, .., 4\}$ $b \in \{0, .., 5\}$
- 学習データ 2-2：$a \in \{0, .., 4, 6\}$ $b \in \{0, .., 5, 6\}$
- 学習データ 2-3：$a \in \{0, .., 4, 6, 7\}$ $b \in \{0, .., 5, 6, 7\}$
- 学習データ 2-4：$a \in \{0, .., 4, 6, .., 9\}$ $b \in \{0, .., 5, .., 9\}$

これらは上から順に, 30 通り, 42 通り, 56 通り, 90 通りのタスクが学習データに存在することとなる. なお, 上記の実験データ 2-4 に関しては, 本文内の実験で用いた学習データ 2 と同様のものである.

A.2.2　結果と考察

結果　図 4 から, 検証データ (上段) の正解率は, すべての学習データにおいて 1 に収束していく様子が見られた. 一方, 評価データ (下段) の正解率は, 学習データの多様性を高めるにつれて増加している様子が確認できる. 表 3 は, 最も検証損失の低かった時点における評価データ正解率であり, タスクの多様性を増加させることで正解率が大きく変化することが分かる. 特に, 学習データ 2-1 では, 検証データの正解率が 1 に収束したにもかかわらず, 評価データの正解率は 0.003 にとどまった.

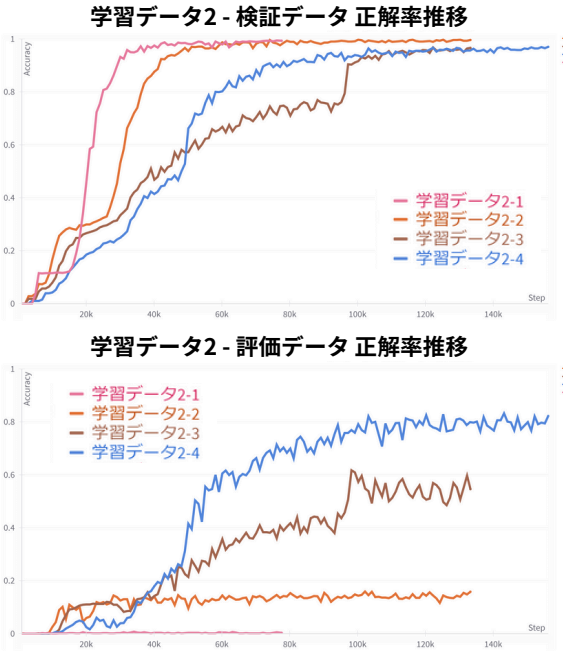

図 4　各学習データごとの検証データ正解率 (上段) と評価データ正解率 (下段) の推移.

表 3　各学習データに対する評価データ正解率

データセット	正解率
学習データ 2-1	0.003
学習データ 2-2	0.158
学習データ 2-3	0.569
学習データ 2-4	0.825

考察　これらの結果は, 学習データ内のタスクの多様性が ICL の能力獲得に重要であることを示唆している. 本文中における実験結果においても, shot 数が 8 の場合にタスクの多様性が変化することによって, 評価データの正解率が大きく変化することが確認された. これは, 学習データに多様なタスクが含まれていることで, 特定のタスクの解法ではなく各タスクに汎用的な解決方法を獲得することが可能になるためだと考えられる. これらの結果からは, ICL 外挿能力の分析について議論するために, タスクの多様性をさらに高めたデータセットでも検証する必要性が示唆される. さらに, 正解率は上記の結果の通り, 条件を変えるだけで大きく変化してしまうため, 内部表現の比較も並行して多面的な分析を行うことも重要である.