

OpenMathReasoning に基づく日本語数学推論モデルの構築

佐々木峻¹ 山本大輝¹ 樋口慎¹ 吉岡駿¹

¹ アクロクエストテクノロジー株式会社

{sasaki,yamamoto,higuchi,s_yoshioka}@acroquest.co.jp

概要

本稿では、FT-LLM2026 コンペティションにおける我々のアプローチを報告する。NVIDIA の AIMO-2 優勝手法である OpenMathReasoning [1] の枠組みに基づき、日本語数学推論に特化したデータセット構築とモデル学習のパイプラインを構築した。具体的には、(1) StackMathQA から Chain-of-Thought (CoT) データセットを作成し、(2) CoT モデルを用いて Tool-Integrated Reasoning (TIR) 用の問題およびデータセットを生成し、(3) ベースモデルに対して SFT を実施した。本手法により、開発用評価データセット（100 問）において正解率 88.0% を達成した。

1 はじめに

FT-LLM2026 コンペティションは、指定されたベースモデルに対してファインチューニングを行い、数学推論タスクの性能を競うコンペティションである。

本研究では、NVIDIA の AIMO-2 優勝手法である OpenMathReasoning [1] の枠組みを日本語数学推論に適用する。OpenMathReasoning では、(1) 大規模な問題セットに対する CoT 解法の蒸留、(2) Python コード実行を統合した TIR 解法の生成、(3) GenSelect による解法選択、の 3 つの柱が提案されている。

本稿では、このうち (1) CoT データセット構築と (2) TIR データセット構築を日本語で実現し、ベースモデルに対する SFT により数学推論性能の向上を図る手法を報告する。図 1 に全体のパイプラインを示す。

2 データセット構築

2.1 CoT データセットの作成

StackMathQA [2] の問題に対して、DeepSeek 3.2 [4] を用いて Chain-of-Thought 形式の解法を生成した。図 2 に CoT データ作成の処理フローを示す。

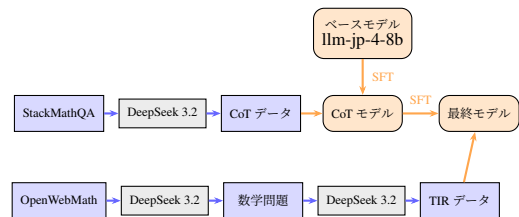


図 1 データセット構築からモデル学習までのパイプライン。青はデータ、橙はモデルを表す。

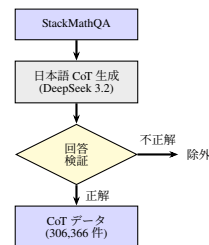


図 2 CoT データ作成の処理フロー

具体的な処理手順は以下の通りである。

Step 1: 日本語 CoT 解法の生成 StackMathQA の問題に対して DeepSeek 3.2 を用いて、日本語で思考過程 (<think>タグ内) と最終回答 (\boxed{} 形式) を含む解法を生成する。プロンプトで日本語での出力を指示することで、翻訳と解法生成を同時に行う。推論過程を明示的に記述させることで、ステップバイステップの数学的思考を学習データに含める。

Step 2: 回答の検証 生成された解法から \boxed{} 内の回答を抽出し、元データの正解と照合する。照合は文字列一致、数値比較（許容誤差 10^{-6} ）、および LLM による等価性判定の 3 段階で実施し、いずれかで一致したデータのみを採用する。

Step 3: 学習形式への変換 検証に通過したデータを、system/user/assistant の 3 ターン形式に変換する。

最終的に 306,366 件の CoT データを生成した。データは system/user/assistant の 3 ターン形式で、システムプロンプトで数学の専門家としての役割を指定し、アシスタントメッセージに <think> タグで囲

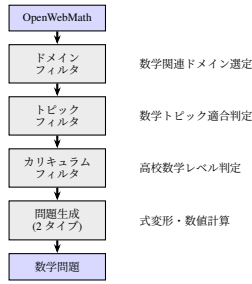


図3 TIR 問題作成の処理フロー

まれた思考過程と `\boxed{}` 形式の最終回答を含む。

2.2 TIR 問題の作成

TIR データセットの元となる問題は、OpenWebMath [3] から抽出した。日本の高校数学レベルに相当する問題素材を得るため、ドメインおよびカリキュラムに基づくフィルタリングを実施した。図3に問題作成のフローを示す。

具体的な処理手順は以下の通りである。

Step 1: ドメインフィルタ OpenWebMath のドキュメントから、数学関連ドメイン (math.stackexchange.com, artofproblemsolving.com 等) のみを選定する。

Step 2: トピックフィルタ DeepSeek 3.2 を用いて各ドキュメントの数学トピックを分類し、高校数学に関連するトピック (代数, 幾何, 微積分等) を含むものを抽出する。

Step 3: カリキュラムフィルタ LLM を用いて日本の高校数学カリキュラム (数学 I/II/III/A/B/C) との適合性を判定し、適切な難易度のドキュメントのみを採用する。

Step 4: 問題生成 フィルタを通過したドキュメントから、DeepSeek 3.2 を用いて2種類の問題を生成する。(1) 式の変形問題: 数式の導出過程を問う問題, (2) 数値計算問題: 具体的な数値を求める問題。

2.3 TIR データセットの作成

OpenMathReasoning [1] の手法に倣い、自然言語による推論と Python コード実行を組み合わせた TIR 形式のデータセットを構築した。図4に TIR データ作成のフローを示す。

具体的な処理手順は以下の通りである。

Step 1: TIR 解法の生成 問題に対して DeepSeek 3.2 [4] を用いて、TIR 形式の解法を生成する。プロンプトでは Python コードを用いた段階的な推論を指示し、`<think>` タグ内に思考過程、コードブロッ

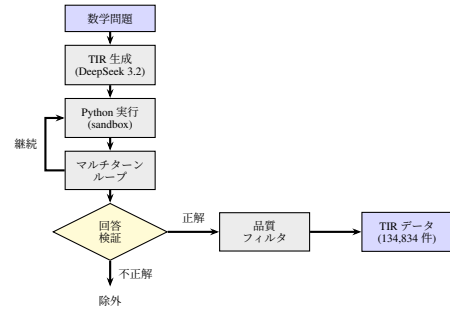


図4 TIR データ作成の処理フロー

ク内に Python コードを記述させる。

Step 2: Python コード実行 生成されたコードブロックをサンドボックス環境で実行し、出力結果を “`output`” ブロックとしてモデルにフィードバックする。実行エラーが発生した場合もエラーメッセージをフィードバックし、モデルに修正を促す。

Step 3: マルチターンループ 最大5回までコード実行とフィードバックを繰り返し、最終回答 (`\boxed{}` 形式) が生成されるまで対話を継続する。

Step 4: 回答検証 生成された最終回答を元の正解と照合する。照合は文字列一致、数値比較、LLM ジャッジの3段階で実施する。

Step 5: 品質フィルタ 正解したデータに対して、コードの新規性 (novelty) と重要性 (significance) を評価し、低品質なデータを除外する。最終的に134,834件の TIR データを生成した。

3 モデルの学習

3.1 ベースモデル

ベースモデルとして `llm-jp-4-8b (v4-8b-decay2m-ipt_v3.1-instruct4)` を使用した。`llm-jp-4-8b` は、LLM-jp プロジェクトが開発した80億パラメータの日本語大規模言語モデルである。

3.2 SFT

OpenMathReasoning [1] の学習手法に倣い、CoT データと TIR データを用いた教師あり微調整 (SFT) を実施した。

学習の設定を表1に示す。CoT SFT ではベースモデルから学習を開始し、TIR SFT では CoT SFT で得られたモデルを初期値として継続学習を行った。

表1 SFT 学習設定

パラメータ	CoT SFT	TIR SFT
ベースモデル	llm-jp-4-8b	CoT モデル
学習率	2e-5	2e-5
エポック数	2	2
バッチサイズ	96	64
最大系列長	4096	4096
オプティマイザ	AdamW	
重み減衰	0.1	0.1
ウォームアップ	20 steps	20 steps
スケジューラ	CosineAnnealing	
精度	bf16-mixed	
データ数	306,366	134,834

4 推論と評価

4.1 推論パイプライン

推論は Singularity コンテナ上で実行した。TIR モードでの推論時には、モデルが生成する Python コードブロックを検出し、サンドボックス環境で実行した後、実行結果をモデルにフィードバックする仕組みを構築した。

推論時には、1 問あたり 15 個の回答候補を生成し、多数決 (majority voting) により最終回答を決定する。複数の推論パスから得られた回答を集約することで、単一の推論よりも安定した正解率を実現する。

推論に関する設定は、1 問あたりのサンプル数 15 (多数決用)、最大コード実行回数 5 回、最大生成トークン数 4096 である。

4.2 評価結果

開発用評価データセット (100 問) における正解率を表 2 に示す。

表2 カテゴリ別評価結果 (開発用 100 問)

カテゴリ	正解率
中 1	1.000
中 2	1.000
中 3	0.938
数学 I・A	0.867
数学 II・B	0.731
数学 III・C	0.800
全体	0.880

中学数学では高い正解率を達成し、高校数学においても数学 II・B を除いて 80% 以上の正解率を達成した。数学 II・B の正解率が相対的に低い原因とし

て、数列や確率などの複雑な問題が含まれていることが考えられる。

5 おわりに

本稿では、OpenMathReasoning [1] の手法を日本語数学推論に適用し、StackMathQA からの CoT データセット構築、TIR データセット構築、および SFT によるモデル学習のパイプラインを報告した。

今後の課題として、GenSelect の導入による推論時の解法選択の改善、データセットの品質向上、および学習手法の改善が挙げられる。

参考文献

- [1] Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, Igor Gitman. AIMO-2 Winning Solution: Building State-of-the-Art Mathematical Reasoning Models with OpenMathReasoning dataset. arXiv:2504.16891, 2025.
- [2] Ting Zhang, Jiahao Ying, Shaowei Lin, Yuyu Luo, Nan Tang. StackMathQA: A Curated Collection of 2 Million Mathematical Questions and Answers Sourced from Stack Exchange. 2024.
- [3] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, Jimmy Ba. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text. arXiv:2310.06786, 2024.
- [4] DeepSeek-AI. DeepSeek-V3 Technical Report. arXiv:2412.19437, 2024.

A 実行コマンド

本研究で使用した主要なコマンドを以下に示す。

A.1 CoT データ生成

Listing 1 CoT データ生成コマンド

```
python stackmathqa_to_cot.py \  
--config stackmathqafull-lq1a \  
--split train \  
--out dataset/smqa \  
--streaming \  
--concurrency 16
```

A.2 TIR 問題生成

Listing 2 TIR 問題生成コマンド

```
# フィルタリングOpenWebMath  
python openwebmath_filter.py  
  
# 問題生成TIR  
python tir_dataset_from_text_parallel.py \  
--in_jsonl high_school_math_filtered.jsonl \  
--out_dir out_tir_dataset \  
--concurrency 8
```

A.3 TIR データ変換

Listing 3 TIR データ変換コマンド

```
python convert_tir_jsonl.py \  
--inputs tir_v1 tir_v2 \  
--output combined_chat.jsonl
```

A.4 推論実行

Listing 4 推論実行コマンド

```
singularity run --nv --writable-tmpfs \  
dist/submission.sif \  
--model_path models/yamalex-tir \  
--input_path input.jsonl \  
--output_path output.jsonl \  
--num_samples 15 \  
--max_steps 5
```

B データフォーマット例

B.1 CoT データ

Listing 5 CoT データのフォーマット例

```
[system] あなたは熟練した数学の専門家であり、注意深い問題解  
決者です。  
... 最終的な答えを  
\boxed{...} で示してください。  
[user] なぜ球の体積は  $\frac{4}{3} \pi r^3$  なのですか？  
[assistant] <think>ユーザーは球の体積公式に現れるがどこ  
4/3から来るのか不思議に思っているようだ  
...  
</think>球の体積の公式における  
4/3 は、積分や幾何学的な比較から自然に導かれます  
...
```

B.2 TIR データ

Listing 6 TIR データのフォーマット例

```
[system] あなたは「自然言語の推論」と「コードの実行」を組み合  
わせて  
Python数学問題を解くアシスタントです。この問題で使えるコード  
実行回数は  
  
...最大  
3 回。  
[user] 正の整数  $a_0$  に対して数列を次の漸化式で定義する  
...  
[assistant] <think>  
Python を使ってシミュレーションしよう。  
</think>  
```python  
def compute_sequence(a0):
 a = a0
 ...
...
```output  
a_0 = 11, a_1 = 88, ...  
```最終答え  
: \boxed{...}
```