$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

LLM-jp チューニングコンペティション 2026

ヒント・手順・Python コード生成、実行による多段パイプラインを用いた数学回答能力の向上

山坂巧（株式会社 DIVX）
橋本賢一

$\partial L / \partial w$

$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

メンバーの紹介

山坂巧

株式会社 DIVX

- 応用数学科卒、公立中学校の数学教員を約10年
- Web アプリケーションのフルスタック開発
- RAG チャットボットの精度改善、LLM を用いた機能開発に従事

橋本賢一

個人参加

- 業務アプリケーションの開発、運用、保守
- 計算機科学系を題材にしたインターンシップ開催
- 生成 AI はトレーニング、AI を用いたシステム、LLM 内部の理解に個人的に取り組む

$\partial L / \partial w$

$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

参加動機

大規模な計算リソースを用いて、AI システム構築の経験ができるから

- 自宅の RTX 3060, 4060, 5090 やクラウド GPU だけでは足りない
- ABCI (A100/H200) の大規模計算環境を活用した本格的な学習を経験したかった

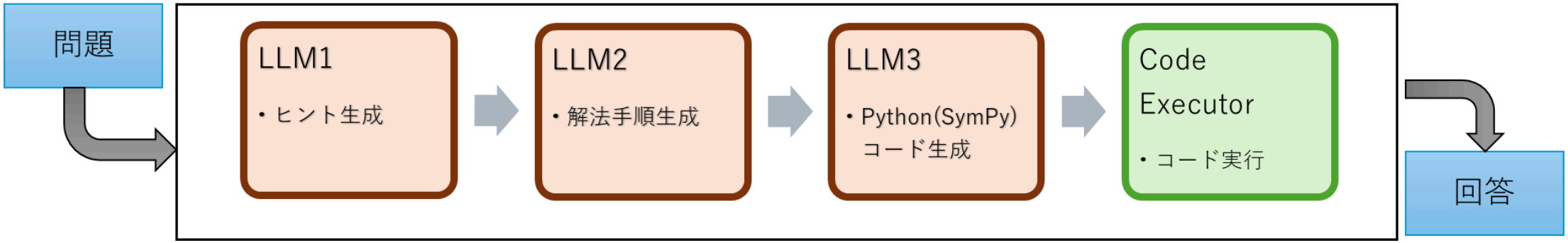
$\partial L / \partial w$

$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

私たちのチームの手法



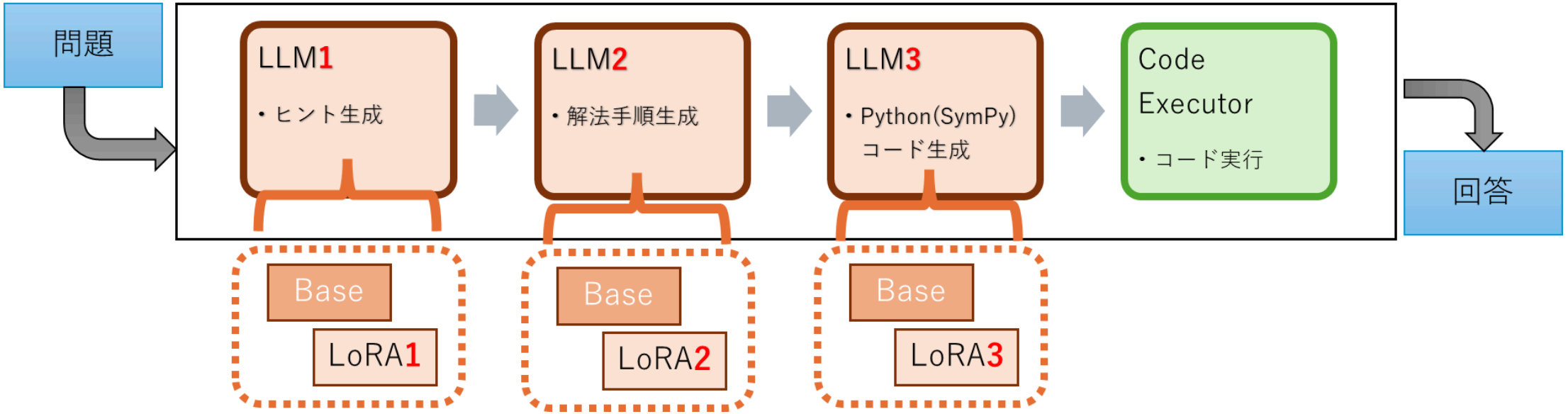
- 設計思想: 「思考」と「計算」を分離
- 言語モデルは計算が苦手 → 計算を Python (SymPy) に委譲
- CodeExecutor はルールベース（Python サブプロセス実行）

$\partial L / \partial w$

$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$ $KL(P\|Q)$

私たちのチームの手法 — LoRA による専門化



- 各段を専用 LoRA アダプタで専門化、ベースモデルは共有
- ヒント・手順は英語で生成（推論時の中間出力言語を学習時と統一）

$\partial L/\partial w$ $\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

結果

カテゴリ	ベースライン	提案手法	改善幅
中1	.667	**.933**	+.266
中2	.667	.611	-.056
中3	.438	**.625**	+.187
数学IA	.067	**.667**	**+.600**
数学IIB	.154	**.462**	+.308
数学IIIC	.400	**.600**	+.200
Overall	**.380**	**.630**	**+.250**

- **Overall +25pt の性能向上** (0.380 → 0.630)
- **数学IA** でコード実行による正確な計算が特に効果的 (+60pt)
- **数学IIB・IIIC** はコード生成品質に改善の余地あり

$\partial L / \partial w$

$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

この方法に至った経緯 (1)

LLM-jpで数学問題を試した結果

- 既存のLLM-jpモデルで問題を試したら悲しい結果に
  - 一方、合成データを大量に学習した他の小型モデルは解けていた
- **事後学習で改善するのが王道**

しかし、ファインチューニングだけでは面白くない

- **CoT (Chain-of-Thought)** のようなものは効果がある
- **計算機をツールとして呼び出す**のもあり

  ↓
- **事後学習以外の挑戦もしてみたい**

$\partial L / \partial w$

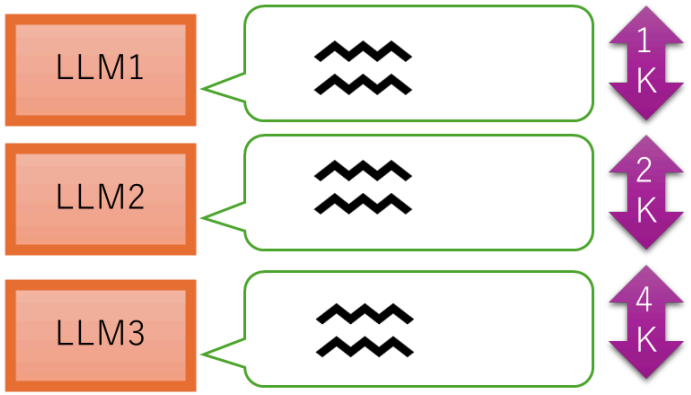
$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$ $KL(P\|Q)$

この方法に至った経緯 (2) — CoT の限界

CoT の限界と多段パイプラインの着想

- 8K コンテキスト長に収まるか不安
  - 長すぎる推論は使えない
- 思考→解法→答え が連続した時
  - **各段階の安定した検証が難しい**
    ↓
- **ヒント,手順,解法の推論を分離する案**を構築
  - コンテキスト長の制御
  - 各段階を独立して検証できる



$\partial L/\partial w$ $\mathrm{ReLU}\ f(x) = \max(0, x)$

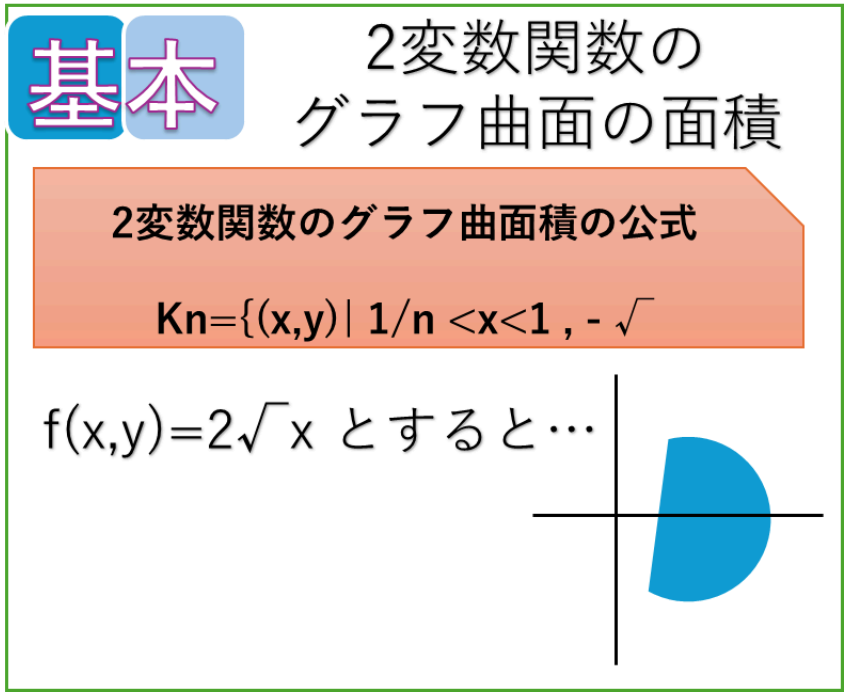
$f(x+h) \approx f(x) + f'(x)h$

$KL(P\|Q)$

この方法に至った経緯 (3) — 数学教育の知見

数学教育の知見を活用

- 問題を**チャート式のようにヒントで分類**する
  - 手順, 解法を探し、問題を解く
    → **この知見を採用**
- 分類したヒントから手順を求める
  - 案1:LLM 推論（今回採用）
  - 案2:ヒントから解法を類似検索（RAG）



$\partial L/\partial w$

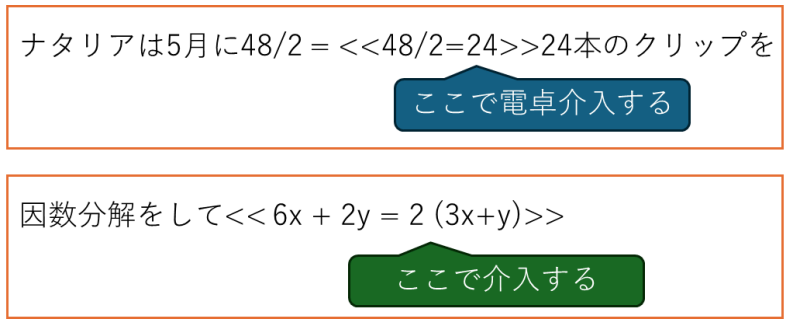
$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$ $KL(P\|Q)$

この方法に至った経緯 (4) — 数式処理の課題

数式処理の課題

- 問題には数式がある — 単なる計算なら電卓でよいが **数式の変形は電卓では無理**
- 数式パーサを作り検証することも考えた（AST の構築は経験済み）→ 規模が大きすぎる



SymPy コード生成方式を選択

- LLM に**数式処理のコード (SymPy) を書かせる**方式に決定
- LLM = 「計算機」ではなく「プログラマー」→ Python が計算を実行、**決定的で正確**

$\partial L/\partial w$ $\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$ $KL(P\|Q)$

学習データ生成

gpt-oss-120B による合成データ生成パイプライン

1. **問題生成** — トピック・学年を指定し日本語で数学問題を生成
2. **ヒント生成** — 解法のヒントを英語で生成
3. **手順生成** — ヒントからステップバイステップの手順を英語で生成
4. **解法生成** — 手順に従い Python/SymPy コードを生成
5. **コード実行検証** — コードを実行し、期待解と一致したもののみ保持

- LLM1, LLM2, LLM3 用にそれぞれ messages 形式の JSONL に整形して学習

$\partial L/\partial w$ $\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

学習データ生成 - データ統計

トピック	件数
Algebra	9,557
Calculus	8,378
Complex	8,871
Linear Alg	9,326
Prob/Stat	9,920
Sequence	9,718

カテゴリ	件数
中1	9,610
中2	9,761
中3	9,454
数学IA	9,881
数学IIB	11,089
数学IIIC	3,145

合計: **108,710 問** (重複除去済み、コード実行検証済み)

$\partial L / \partial w$

$\text{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

トレーニングパラメータ

LoRA 設定

パラメータ	値
rank (r)	16
alpha	32
dropout	0.05
target_modules	q, k, v, o_proj

学習ハイパーパラメータ

パラメータ	値
learning_rate	2e-4
epochs	3
実効バッチサイズ	16 (4 x 4)
max_seq_length	512 / 1024
dtype	bfloat16
EarlyStopping	patience=3

- ベースモデル: `v4-8b-decay2m-ipt_v3.1-instruct4` (LLaMA 8B)
- フレームワーク: HuggingFace TRL `SFTTrainer` + PEFT `LoraConfig`
- 計算環境: ABCI rt_HG (A100 GPU)、Singularity コンテナ

$\partial L / \partial w$

$\mathrm{ReLU}\ f(x) = \max(0, x)$

$f(x+h) \approx f(x) + f'(x)h$

$KL(P \| Q)$

さいごに

謝辞: LLM-jp チューニングコンペティションの開催および ABCI 計算資源の提供をいただいた LLM-jp プロジェクトに深く感謝いたします。

山坂のコメント

- 多段推論 + SymPy という個性ある手法を形にできた
- LoRA の対象層やハイパラの探索をもっとやりたかった
- 3段が本当にベストかは未決着 — 2段で悪化した原因の理解が不十分
- 「人間の思考過程を踏めば LLM も良くなるはず」という思い込みに気づけたのは収穫

橋本のコメント

- 短期間で 2 人チームで成果が出せていること。チームメンバーの山坂の意欲に感服した
- 今回の学習結果を得てメカニスティック・インタープリタビリティの観点でモデルを調べたくなった

$\partial L / \partial w$

$\mathrm{ReLU}\ f(x) = \max(0, x)$