

報告書

1 今週の進捗

- gpt-oss-20B でのアニメとアニメ以外の質問と回答の比較
- Wikipedia の md での gpt-oss-20B の継続事前学習および教師あり学習
- llm-jp-7.2b-instruct3 を用いた継続事前学習

2 gpt-oss-20B でのアニメとアニメ以外の質問と回答の比較

gpt-oss-20B において継続事前学習および教師あり学習をしたモデルに対し、アニメとアニメ以外の質問の回答を比較した。

3 Wikipedia の md での gpt-oss-20B の継続事前学習および教師あり学習

gpt-oss-20B を用いて継続事前学習および教師あり学習をした。

3.1 訓練データ

継続事前学習の訓練データは、Wikipedia から集めたアニメに関する記事の約 18,000 件のマークダウンファイルである。

また、教師あり学習の訓練データは、以前作成したアニメに関する質問と回答のペアの QA データセットである。表 1 に例を示す。

3.2 gpt-oss-20B のモデルアーキテクチャ

モデルは 24 層の Decoder Layer から構成されており、各層は Self-Attention ブロックと MoE 構造を持つ MLP ブロックを含む。MoE ブロックは 32 個の Expert 層を持ち、計算効率とモデル性能の両立を図っている。

```
GptOssForCausalLM(  
  (model): GptOssModel(  
    (embed_tokens): Embedding(201088, 2880, padding_idx=199999)  
    (layers): ModuleList(  
      (0-23): 24 x GptOssDecoderLayer(  
        (self_attn): GptOssAttention(  
          (q_proj): Linear4bit(in_features=2880, out_features=4096, bias=True)  
          (k_proj): Linear4bit(in_features=2880, out_features=512, bias=True)  
          (v_proj): Linear4bit(in_features=2880, out_features=512, bias=True)  
          (o_proj): Linear4bit(in_features=4096, out_features=2880, bias=True)  
        )  
        (mlp): GptOssMLP(  

```

表 1: 訓練データセットの例

| |
|---|
| 質問: ぽにぽにだっしゅ!の主人公は誰ですか? |
| 出力: ぽにぽにだっしゅ!の主人公はレベッカ宮本です。 |
| 質問: ぽにぽにだっしゅ!の主人公は誰ですか? |
| 出力: ぽにぽにだっしゅ!の主人公は、10歳でMITを卒業した天才少女、レベッカ宮本です。 |
| 質問: 「ぽにぽにだっしゅ!」で主人公として描かれているキャラクターは誰ですか? |
| 出力: ぽにぽにだっしゅ!の主人公はレベッカ宮本です。 |
| 質問: 「ぽにぽにだっしゅ!」で主人公として描かれているキャラクターは誰ですか? |
| 出力: ぽにぽにだっしゅ!の主人公は、10歳でMITを卒業した天才少女、レベッカ宮本です。 |
| 質問: レベッカ宮本が主人公を務める作品は何ですか? |
| 出力: レベッカ宮本が主人公を務める作品は「ぽにぽにだっしゅ!」です。 |
| 質問: レベッカ宮本が主人公を務める作品は何ですか? |
| 出力: 彼女が主人公の作品は「ぽにぽにだっしゅ!」です。この作品で彼女は天才少女教師として描かれています。 |
| 質問: レベッカ宮本は、どの作品で主人公として登場しますか? |
| 出力: レベッカ宮本が主人公を務める作品は「ぽにぽにだっしゅ!」です。 |
| 質問: レベッカ宮本は、どの作品で主人公として登場しますか? |
| 出力: 彼女が主人公の作品は「ぽにぽにだっしゅ!」です。この作品で彼女は天才少女教師として描かれています。 |

```

(router): GptOssTopKRouter(
  (linear): Linear(in_features=2880, out_features=32, bias=True)
)
(experts): GptOssExperts(
  (gate_up_projs): ModuleList(
    (0-31): 32 x Linear4bit(in_features=2880, out_features=5760, bias=True)
  )
  (down_projs): ModuleList(
    (0-31): 32 x Linear4bit(in_features=2880, out_features=2880, bias=True)
  )
)
)
(input_layernorm): GptOssRMSNorm((2880,), eps=1e-05)
(post_attention_layernorm): GptOssRMSNorm((2880,), eps=1e-05)
)
)
(norm): GptOssRMSNorm((2880,), eps=1e-05)
(rotary_emb): GptOssRotaryEmbedding()
)
(lm_head): Linear(in_features=2880, out_features=201088, bias=False)
)

```

3.3 学習手法と対象パラメータ

継続事前学習では、パラメータ効率の良いファインチューニング手法である Low-Rank Adaptation (LoRA) を採用した。LoRA は、元のモデルの重みを凍結したまま、特定の線形層に低ランク行列分解を利用したアダプター層を追加し、そのアダプターの重みのみを学習する。これにより、少ない計算リソースで大規模モデルの挙動を適応させることが可能となる。

本研究では、モデルの挙動に大きく影響を与える以下の層を学習対象 (target_modules) として設定した。

- **q_proj, k_proj, v_proj, o_proj**: これらは Self-Attention 機構を構成する主要な線形層である。Query, Key, Value, Output の射影をそれぞれ担当し、文中のどの情報に注目するかという注意パターンを学習する。
- **gate_proj, up_proj, down_proj**: これらは MLP ブロック (本モデルでは MoE の各 Expert 層) を構成する線形層である。Attention からの出力を処理し、モデルが持つ知識や表現力を更新する役割を担う。

3.4 実験設定と結果

3.4.1 学習パラメータ

表 3 に継続事前学習の主要なハイパーパラメータを示す。

表 3: 学習ハイパーパラメータ

| パラメータ | 値 |
|---------------------|---------------------|
| モデル | unsloth/gpt-oss-20b |
| LoRA ランク (r) | 16 |
| LoRA アルファ (alpha) | 32 |
| LoRA ドロップアウト率 | 0.05 |
| バッチサイズ (per device) | 8 |
| 勾配蓄積ステップ | 2 |
| エポック数 | 3 |
| 学習率 | 2e-4 |
| オプティマイザ | AdamW (8-bit) |

4 llm-jp-7.2b-instruct3 を用いた継続事前学習

llm-jp-7.2b-instruct3 のモデルを用いて継続事前学習をした。データセットとして Wikipedia から集めた銀魂のみのデータを用いている。

現在、この継続事前学習済みを用いて教師あり学習を実施するところである。

5 今後の課題

- 継続事前学習後の llm-jp-7.2b-instruct3 での教師あり学習
- gpt-oss-20B のコードの見直し

参考文献