

大規模言語モデルの事前学習ツール jax-llmの開発と input-methodへの応用

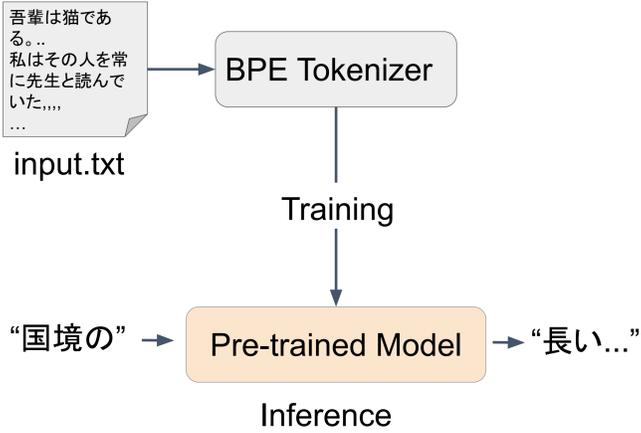
杉浦 一瑛

sugiura.issa.q29@kyoto-u.jp



What is jax-llm?

1つのtxt file をもとにLLMの事前学習を簡単に
行えるツール



背景

多くの既存実装はコードが膨大で内部の
仕組みがわかりにくい 🙄

→シンプルな実装でLLMの学習の経験者
・理解者を増やしたい

jax-llm の特徴

JAXを使用

- Numpy + 自動微分
- 関数型言語指向

シンプルなコード

- 機能が最小限
 - pure jax/flax training loop

データ並列に対応

- 複数GPUで分散学習できる

実行例

学習時間:

1GPU (A100) で1hour

学習データ:

青空文庫 80M Tokens

モデルパラメータ数:

80M Parameters

推論例:

Prompt: 国境の長いトンネルを抜けると雪国であった。

Output: 国境の長いトンネルを抜けると雪国であった。このあたりは、一冬に満たねばならなかった。そして、その冬枯の季節、冬の日、その季節、冬の間、寒の季節に、大気の冷える夜だった。 [...]



First Two Character Input Method ~ jax-llm の応用先として ~

First Two Character Input Method

- 英単語の最初の2文字(first two-char)からなる文字列を単語列に変換する手法
 - fitwchinme → first two character input method
- 言語情報処理特論の課題で出題された Toy Problem

Input Method の応用先

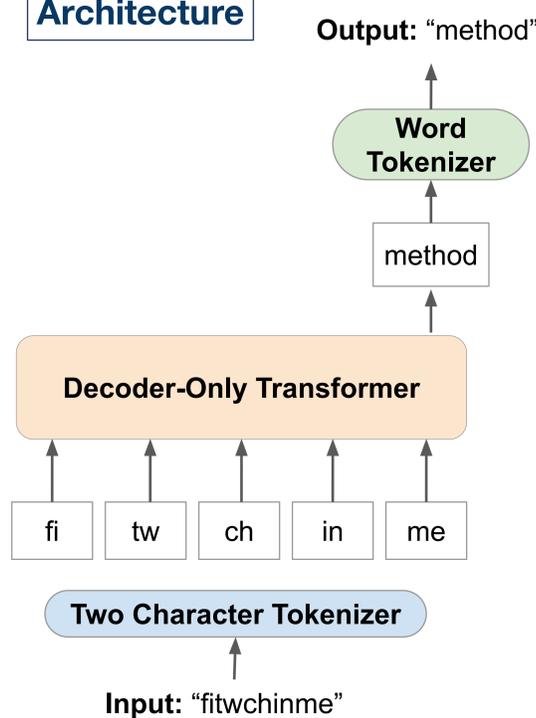
- ユーザーがテキストを入力するために使用する手法やシステム
 - i.e., かな漢字変換, ピンイン入力

仮定: オンライン予測

- アルファベット2文字が入力された段階で、直近のfirst two-charに対応する単語を予測する
 - fi → first
 - fitw → two
 - fitwch → character
 - fitwchin → input
 - fitwchinme → method

提案手法: Decoder-Only Transformer モデルによる系列変換

Architecture



💡 入出力で異なる Tokenizerを使う

Two Character Tokenizer

- Vocab = {"a", "b", ..., "aa", ..., "zz"}

Word Tokenizer

- Vocab = {"a", "an", ..., "word", ...}

学習方法

- 入力列(fi, tw, ch, in, me)に対して、直近のfirst two-char (me)に対応するword (method)を予測する
- Next Token Predictionに類似

既存研究

Attention Is All You Need [Vaswani+, 2017]

- Encoder-Decoder アーキテクチャを系列変換に用いる

Improving Language Understanding by Generative Pre-Training [Radford+, 2018]

- Decoder アーキテクチャを言語モデルに用いる

実験

比較対象

- n-gram モデル
 - バックオフを利用
 - i.e., n-gram の頻度が0 → (n-1)-gram を使用
- Transformer モデル
 - GPT-2 like アーキテクチャを利用

評価方法

- Accuracy

データセット

- Shakespeare
- WikiText2

変換例

Prompt: fitwchinme (First two character input method)

Transformer モデル (Block size 16)

Output: first two characters in mexico

n-gram モデル (n=2)

Output: first two children in mexico

結果

- n-gramモデルにおいては2-gramが最も性能が良い
- Transformerモデルは2-gramと同程度の性能

n-gram モデル				
n (of n-gram)	shakespeare		WikiText2	
	Train Acc	Test Acc	Train Acc	Test Acc
1	0.472	0.473	0.429	0.417
2	0.549	0.522	0.512	0.455
3	0.765	0.505	0.739	0.447
4	0.965	0.499	0.950	0.441
5	0.997	0.499	0.995	0.440
6	1.000	0.499	0.999	0.440

Decoder-Only Transformer モデル				
Block Size	shakespeare		WikiText2	
	Train Acc	Test Acc	Train Acc	Test Acc
4	0.498	0.495	0.460	0.438
8	0.526	0.513	0.476	0.447
16	0.570	0.518	0.515	0.456
32	0.671	0.515	0.534	0.462
64	0.807	0.503	0.622	0.457

Code

