解説

ローカルPCでのLLM（大規模言語モデル）について†

久富　望*1

1. はじめに

本稿は，ChatGPT のようなモデルを自分の端末で動かすこと，言い換えると，PC 上で動く LLM（大規模言語モデル，Large Language Models）について，2024 年 7 月における現状をまとめ，その影響について考察するものである．

ChatGPT の登場以後，生成 AI に関する記事が日々生まれている．しかし，インターネットから切断された PC においても動く LLM（以下，ローカル LLM と呼ぶ．本来はオンプレミス LLM というべきかもしれない）についての記事は相対的に少ないと私は感じている．

驚くことに, 2022 年末から全世界を驚かせた最初の ChatGPT（GPT-3.5）と同程度の LLM は，既に市販のノート PC で動かすことが可能である．しかし，LLM がノート PC で動く，という事実自体も理解してもらいにくい．実際，ChatGPT という何か人格のようなものをもつ知能のようなものの類似品が，自分のスマホに入る 100GB にも満たないサイズであり，ノート PC でも動かせるという事実には，理解が追いつかないのも無理はない．結果として，ローカル LLM がどのように社会を変えるのか，各分野に，組織や個人に影響を与えるのか，議論が不十分のように私は考えている．

本稿では，プログラムを書かない方々にも理解いただけるように配慮しながら，生成 AI の生成から利活用までを，ローカル LLM の情報を中心に概観する．結果として，LLM の専門家や日常的にプログラムを書く人には目新しい情報でなかったとしても，多様な分野の方がローカル LLM のもたらす影響について考える材料となればと考えている．ただし，特定のサービスにはできるだけ言及せず，「このようなものがある」という記述にできるだけ留めることをご了承いただきたい．気になる内容があれば現状について探していただきたい．

2. ローカル LLM の概観：動かすまでの手順

はじめに，インターネットから切断された PC において LLM を動かすための手順を簡単に記すと，以下のようになる．

1. 動かしたいモデルを決める．決める際には，LLM の性能ランキングやモデルの特徴を参考にする．
2. プログラミング言語 Python と，LLM を動かすために必要なパッケージをインストールした PC に，動かしたいモデルをダウンロードする．
3. モデルを動かすプログラムを実行すると，モデルが PC のメモリ（VRAM またはユニファイドメモリ）に読み込まれ，入力した言葉に対する回答を PC が計算・出力する．

1. について，そもそも LLM のモデルが様々な組織・個人によって日々作られていること自体，あまり知られていないだろう．たとえば，Open LLM Leaderboard [1] には Hugging Face に公開されている LLM のモデルがリストアップされ，ランキングは日々更新され，性能を競っている．Hugging Face とは（ローカル LLM 関係のものに限らない）機械学習のモデルやデータセットなどを共有する代表的なプラットフォームである．

他の LLM のリストとしては，GPT-4 のようなモデルが非公開なものも含めたリスト [2] もある一方，日本語を中心とした LLM のリスト [3] も，国立情報学研究所（NII）が主宰する LLM 勉強会（LLM-jp）[4] の資料の 1 つとして公開され，更新されている [5].

さて，LLM のモデルの Web ページを開いてファイル一覧を見ると，パラメータ数が 70 億，130 億，700 億（7b, 13b, 70b. 1b = 1 billion = 10 億．以下も同様）のモデルは，それぞれ約 13GB，26GB，130GB のファイル群であることが分かる．既に，初期の ChatGPT を超えたという 70b のモデルは複数ある．言い換えると，たった 130GB 程度のファイルが，他の情報を参照することなく，流暢に翻訳したり，誤りを修正したり，知識を披露したり，プログラムを書いたり，様々な機能を備えると共に，しばしば嘘を言って人を惑わしている．

加えて，量子化という操作（4.2 節）によって，若干の性能低下などと引き換えにファイルサイズを数分の 1 に減らすことができる．そのため，20GB 超の 70b のモデルも登場し，ローカル LLM を試すハードルは大きく下がった．

2. の手順については，Python を PC にインストールしない手段もいくつか存在する．たとえば，アプリケーションが PC 内に，LLM の動くサーバーの仮想環境を用意することで，プログラムなどの知識がなくても汎用的なローカル LLM の環境を比較的手軽に用意することは可能である．

手順 3. における「PC のメモリ」には注意が必要である．これは，市販の WindowsPC の仕様に記されているメモリではない．簡潔に言えば，GPU（Graphics Processing Unit）のメモリであり，しばしば VRAM と表記される（Video Random Access Memory の略．通常のメモリは RAM である）．GPU は画像を処理するための PC の部品の 1 つであるが，深層学習（Deep

---

† On LLM (Large Language Models) on Local PC

Nozomu KUTOMI

*1 京都大学大学院教育学研究科
Graduate School of Education, Kyoto University

Learning）を行うため，実質的に必須の部品である．

GPU は単体で売られ，最も広く使われているのが NVIDIA 社のものであり，VRAM は GPU に固定されている．一般向けに売り出されているデスクトップ用 GeForce であれば，VRAM が 6GB から 24GB のものが，3 万円程度から 30 万円前後で売られている．なお，LLM を開発している企業が用いる上位モデルの GPU はあるが，VRAM が 80GB の H100 は 1 台で 500 万円を超え，一般的な PC よりも遥かに高額である（最近，94GB の H100 も売られている）．

ただし，Apple シリコン搭載の Mac は例外である．2020 年から登場し始め，今はすべての Mac が Apple シリコンを搭載しているが，これらは CPU と GPU で共用のメモリ（ユニファイドメモリ）を備え，LLM を読み込んで GPU の計算に利用可能である．メモリが最も大きな Mac は，デスクトップ PC であれば Mac Studio（M2 Ultra）の 192GB（838,800 円〜）や Mac Pro（M2 Ultra）の 192GB（1,288,800 円〜），ノート PC であれば MacBook Pro（M3 Max）の 128GB（764,800 円〜）である．

もっとも，VRAM でもユニファイドメモリでもフルに使えるわけではない．24GB の VRAM をもつ GeForce3090 でも 20GB 弱のモデルまでしか動かせない．ユニファイドメモリは CPU との共用であるため一層制限される．私が試す範囲では，少なくとも他のアプリケーションをすべて終了させた状態であれば，メモリが 64GB の Mac であれば約 20GB 超，128GB の Mac であれば約 50GB 超のモデルなら安定して動くようである．また，メモリへの読み込み時のサイズが，モデルのファイルサイズより少し大きくなる点も注意が必要である．

なお，CPU と通常のメモリ（RAM）でも LLM は一応動く．しかし，GPU と比べて数十倍の時間がかかり，基本的には実用的でない．LLM の一部を CPU のメモリに振り分ける手段もあるが，詳細は省く．

3. 生成 AI の生成から利活用までの概観

生成 AI の生成から利活用までは，概ね以下の 5 ステップに分けられると私は考えている．

1. 大規模言語データを処理できるアーキテクチャが設計され
2. 大規模なデータで訓練されパターン的なものを学習し
3. 人間の振る舞いのような反応をするよう調整されたうえで
4. 誰でも使いやすいようなサービスとして提供され
5. 社会・制度・倫理観に影響を与えている

以下では，ローカル LLM の影響に焦点を当てながら，1 つずつ記していく．

3.1 アーキテクチャの開発

今は，人工知能ブームの第 3 期と言われるが，その引き金は 2010 年代前半からの深層学習のブレイクスルーである．まず，飛躍的な成果が画像処理において生まれ，自然言語処理などの他分野にも波及してきた．深層学習は人間の脳の神経回路を模して開発されたが，現在作られているモデルは，人間の脳の回路とは別物だと言われている．

深層学習と言っても様々なものがあり，その有効性が広く認められて以降，RNN（Recurrent Newral Network）や LSTM（Long Short Time Memory）のような様々な仕組みが提案されてきた．しかし，LLM において決定的な役割を果たしたのは Transformer と呼ばれる仕組み [6] であり，用語 GPT（Generative PreTraining Transformer）にも表れている．もっとも，Transformer を必須としても，それを含めた様々な仕組みをどのように組み合わせれば性能の良い LLM ができるかについて，様々な可能性がある．その組み合わせ方を LLM のアーキテクチャと呼ぶことがあり，本稿ではそれに倣う．

代表的なアーキテクチャは大きく 2 つに分けられる．非公開なものと，リソースが公開されているものである．

非公開なアーキテクチャの代表例は OpenAI 社のものであろう．初代 GPT（パラメータ数 0.117b）では公開されていたが，GPT-2（パラメータ数 1.5b）の発表時は悪用などを恐れて公開されなかった．これには世界中の研究者やプログラマからさまざまな批判を浴びたが，GPT-3（パラメータ数 175b），3.5，4 においては今でも公開されていない[1]．他にも，Google 社の Gemini や，Anthropic 社の Claude なども非公開である．

一方，ローカル LLM は基本的にリソースが公開されている．すなわち，利用可能な LLM のモデルが，アーキテクチャや 3.2，3.3 節で取り上げる学習データと共に公開されている．リソースが公開されていると，開発者が利用料などを得ることは難しくなる一方，多数のプログラマが自由に開発に参加できることで，サポートや安全性への監視と改善，様々なバージョンへ派生する柔軟性などをもたらす．また，それらのコミュニティからの知見を取り込むことで，開発元にとっても開発コストの削減に繋がることがある．なお，「リソースが公開されている」と言ってもライセンスは色々である（3.5 節）．

リソースを公開している最も有名な LLM は Meta 社による Llama であろう．後述するような訓練を行なったモデルを商用可能（ただし，月間アクティブユーザー数が 7 億人を超える場合は別途ライセンス契約が必要）な形で公開している．2024 年 4 月に Llama3（パラメータ数 8b, 70b，7 月末の Llama 3.1 ではパラメータ数 405b も追加）が公開されるや否や，ファインチューニング（3.3 節）等を独自に行ったモデルや，量子化によって軽量化されたモデルが大量に作成・公開され，前述の Open LLM Leaderboard [1] を賑わせている．同様にリソースを公開する方針は，いくつかの企業や，大学などの研究機関が取っている．

もっとも，アーキテクチャの訓練には，以降で紹介するような多大なコストや労力がかかり，アーキテクチャから開発することは，世界的に規模の大きい企業でないと手が出しづらい．

3.2 事前学習：大量の文章で訓練する

LLM を作るには，まず，大量の文章によってアーキテクチャの訓練が行われる．これは，機械学習における事前学習にあた

1 いずれもパラメータ数は非公開．ただし，GPT-4 が 1,760b，GPT-5 は 52,500b との噂もある [7]．なお，ChatGPT に用いられている GPT-3.5 のパラメータ数を約 3,550 億としているサイトは多数あるが，実際には非公開である．ChatGPT の公開直後に行われた，ChatGPT へのインタビューにおける誤情報が拡散された結果との指摘がある [8].

る．事前学習によって作られたLLMを基盤モデルと言う．

一般的に，大量の文章はWeb上から集められている．パラメータ数175bのGPT-3の場合，ウェブ上のデータを定期的に集めて公開している非営利団体Common Crawlのデータが最も多く，学習データの6割を占めている．これは，45TBのWebデータをフィルタリングし，学習に使えるようにした570GBのデータである．他に，WebText2，書籍コーパスであるBooks1，Books2と，英語のWikipediaが使われている[9]．

なお，大量の文章をモデルに学習させるためにはMary and Tom have fruits and water.から[576, 35, 612, 69, 156, 35, 89, 3]のように，単語やピリオドのような記号にも番号づけがされている（and → 35，ピリオド→ 3）．実際には，単語をもう少し細かく分けられたトークンと呼ばれる単位（上記の例であれば，fruitsはfruitとsに分けられるかもしれない）で番号付けがなされる．GPT-3に用いられた570GBのCommon Crawlのデータの場合は約4,100億トークンへと分けられ，これら番号の羅列から，番号の出現パターンのようなものが学習される．

この事前学習には，1回で数億円かかることも珍しくなく，どのような大量の文章を学習させると質の良い基盤モデルを作ることができるか，という知見を積み重ねるのも難しい．

最大のネックは莫大な数のGPUが必要なことだろう．たとえば，Meta社は2節で述べた500万円を超えるGPUのNVIDIA H100を何十万台と購入し，24,576基を搭載したシステムを構築しているという[10]．これらの購入費用だけでも凄まじいが，計算を実行するときの消費電力も1基あたり数百Wと大きく[2]，メンテナンスも容易ではない．ほとんどの組織にとって，このような大量のGPUを備えた計算環境は，自前で構築できる規模を超えている．このため，このような環境を構築できる組織は，その計算機能の一部をクラウドサービスとして提供していることもある．その多くは民間企業によって運用されているが，大学や研究所による例もある．

たとえば，国内では7大学と2研究所で運用されているデータ活用社会創成プラットフォームmdxがある．LLM勉強会[4]がアカデミアや産業界の研究開発のために公開しているLLM-jp-13B v2.0は，上記のmdxにあるVRAM 40GBのNVIDIA A100を128基用い，約1,300億トークンの日本語（日本語Common Crawl，日本語Wikipedia），約1,200億トークンの英語（英語Pile，英語Wikipedia）と約100億トークンのプログラムコード，合計で約2,600億トークンを事前学習して作られた[11]．他にも，LlamaやGPT-NeoXなどのアーキテクチャを用いて事前学習した日本語LLMは複数作られている[5]．

さて，一度作成された基盤モデルに対して，追加のデータで学習させることも可能であり，追加事前学習と呼ばれる．このため，リソースが公開されているLLMに対し，様々な追加事前学習を行ったモデルも多く作られている[5]．多くの基盤モデルは英語を中心とした事前学習になっているため，日本語を含めた他言語のデータによる追加学習には意義も大きい．

2 NVIDIA社のH100 SXM（大規模なデータセンターに適した接続方法）は最大700Wの電力を消費する．

Meta社が莫大な投資をどのように回収する考えでいるのかは私には分からないが，少なくとも，次節のファインチューニングも含め，世界中の組織・個人を巻き込みながら，LLMの開発コミュニティをMeta社が活性化し，LLMの発展に寄与していることは間違いない．

3.3 ファインチューニングとモデルの評価

大量の文章から単語間の関係性を学習させただけの基盤モデルに，人間らしい会話をさせるためのステップがファインチューニングである．

一般に，ファインチューニングとは，大規模なデータセットで学習した汎用的なモデルを，比較的少数の特定のタスクやデータセットによって訓練し，何らかのタスクに適したモデルへと変えるものである．今回であれば「人間らしい会話」を実現するようなタスクやデータセットによって訓練する．初期のChatGPTは，GPT-3.5に対して，(1)「回答例」を学習させ，(2)「回答例の順位づけ」を学習させ，(3) 2つを組み合わせて再学習するプロセスを経て作られた．データ数は比較的少数なので個人でも参画しやすい．

「人間らしい会話」といっても様々な人間らしさがある．実際，ここでどのようなタスクを課すかで，モデルの性質は大きく変わってくる．特定の領域，たとえば医療に強いLLMになるよう訓練することも可能である．このため，様々な人が自分でファインチューニングして公開し合うコミュニティが形成され，ファインチューニングに用いられたデータも含め，Hugging Faceに公開されている．

さて，モデルの性能を比べ，「ChatGPTを超えた」のような主張を行うには，その性能を測るテストが必要である．このため，特定の分野に関するテストや，様々な内容をバランスよく含めたテストが多数公開されている．代表的なテストについては簡便に評価まで計算できるようプログラムが用意されており，それらの評価値をもとにOpen LLM Leaderboard [1]のランキングが作られる．

このようなテストには，答えが決まった問題で構成されたものが多く，LLMの回答の信頼度を評価するうえで有用である．その一方で，人々は，決まった答えのないようなタスク，たとえば何かの提案を求めたり，物語の創作を求めることがある．

このような答えが決まっていないタスクを集めたテストもあり，回答に対する採点基準を設定して評価されている．たとえば，日本のELYZA社の作成したタスク集Elyza-task 100 [12]は，日本語による提案や創作を含む複雑な指示・タスクから成っている．これらに対し，評価対象のLLMに回答を求め，採点基準に沿って3人が5点満点で評価を行い，その平均値をもってモデルの評価値となる．タスク・回答例・採点基準・3人の評価値は公開されている[13]ので，評価値の妥当性や，実際の回答のニュアンスも確認する事ができる．

もっとも，答えが決まっていないタスクの評価を，GPT-4に評価させる例もよく見られる．実際，いくらかGPTは評価が甘い場合もあるが，人とGPTは既に同レベルとの報告もある[14]．こういう意味では，既に生成AI同士で評価しあってAIが高め合う構造は構築されつつある．

この他，注目すべき方法はモデルの結合であろう．モデルの形が同じであれば，複数のモデルを結合する手段がある．これが注目に値するのは，日本語に特化したモデル A と数式に特化したモデル M を結合してできたモデル AM が，日本語にも数式にも強いモデルができただけでなく，モデル A より日本語に強く，モデル M より数式に近い場合が報告されていることである [15].

3.4 使いやすいサービスとしての提供

LLM を開発した企業が，LLM と直接会話しているようなチャット形式を提供するやり方が，最も一般的なサービスであろう．しかし，LLM を持たない企業が，機能の一部に LLM を用いたサービスを提供している場合も多い．

たとえば，より自然な会話ができると謳った AI や，文書の要約を行う Web サービスなどが，2023 年にはたくさん生まれた．これらのサービスには内部で API を通じて ChatGPT を用いているものも多く，サービス利用料などを元に，サービスの提供元が OpenAI 社に費用を払っている．このような API を通じたサービスは，OpenAI 社が同様のサービスを始めてしまうと維持が難しくなる．

どちらのやり方であっても，ローカル LLM に置き換えることは難しくない．Python の関数などを用いて PC 内の LLM を呼び出せばよい．ただし，Python を用いて直接呼び出すのではなく，ローカル LLM を PC 内部の仮想サーバー内で動くようにし，http://localhost:xxxx として呼び出す場合もある．言い換えると，サーバーに GPT-3.5 を置いている OpenAI 社の環境を，自分の PC 内の仮想サーバー内に自前の LLM を置いて実現し，そこへアクセスするやり方である．この場合，接続先の http 以降を変えるだけで，サービスが接続する LLM を切り替えられる．たとえば，ChatGPT のようなユーザー・インターフェースを用意し，接続先を OpenAI 社の API にすれば実質的に ChatGPT であるし，PC 内の仮想サーバーにすればローカル LLM のチャットサービスになる．もちろん，サービスの質は，用いる LLM だけでなく，LLM に伝える際のプロンプトにも大きく左右される．

3.5 社会・制度・倫理観への影響

AI が誤情報を生成し，ハルシネーションを起こす危険のあることはよく知られているだろう．それを防ぐための努力は多数なされているが，現在知られている LLM のアーキテクチャには，知性や論理的正しさを明示的に学習するような仕組みを持っていない．個人的には，疲れを知らない「世界中の文書を読み込んだ 4 歳児」[16] くらいだと思って付き合うのがいいのではないかと思っている．

このような LLM の性質を踏まえた適切な利用方法については，国際機関や政府のレベルでの議論はもちろん，研究において，教育において，報道において議論は尽きないが，本節では，ライセンスの観点から少し取り上げたい．

3.2 節において，LLM のリソースが公開され，開発に関わるプログラマを増やし，LLM の発展が促されていることを述べた．このようなオープンなコミュニティにおけるソフトウェアの発展は，過去にも OS，ブラウザ，メールソフトなどで見られたものである．このようなオープンな LLM の発展の一端を担っている Meta 社は，Llama を open source としている [17].

しかし，この “open source” は「オープンソース」という語がもつ一般的な意味とは異なり，source がオープンであることを意味しているに過ぎないようだ．たとえば，Meta 社は商用利用に関する緩い制限（3.1 節）や，他人に危害を加えるような利用方法などへの制限を課している．他にも，LLM が作った人間らしい会話の例を，別の LLM の学習に使うことは禁じられている場合がある．Llama でも，Llama ベースのモデルが生成したものは，Llama ベースのモデルの訓練にしか用いてはいけない．これらのいくつかは，健全な LLM の発展のために妥当な制限と思われるかもしれないが，開発者への制限が極力抑えられている一般的なオープンソースライセンス（MIT や Apache-2.0 など）とは相入れないものであり，OSI（Open Source Initiative）は，Meta 社の Llama 2 ライセンスはオープンソースでないと声明を出している [18].

いずれにせよ，ローカル LLM の存在は，AI について考えるべきことを増やす．たとえば，年齢制限のある現在のクラウド上の LLM のサービスに比べ，ローカル LLM は制限が緩く[3]，ゼロからオープンソースライセンスで開発されたモデルには制限が一切ない．開発元が制限を付けて安全性を担保するべきか，制限がない代わりに適切な使い方を草の根で形成していくか，今後も世界中で模索が続くだろう．OSI も Open Source AI について議論を始めている [20].

教育と生成 AI の関係については，私自身も研究会や授業でも何度も取り上げてきた．もちろん，日本中で様々な真摯な議論がなされており，高等教育では多数の議論・発信がなされているし，初等中等教育では文部科学省が様々な発信 [21] をしている．引き続き，ローカル AI の可能性も想定して，利用者側の制限についても，逆に，適切な AI の発展に寄与できる担い手の育成についても，議論が深められていくべきだろう．

4. ローカル LLM の実現に繋がる動き

ローカル LLM の実現を簡便にするには，モデルを小さくし，デバイス上のメモリを大きくすればいい．この他に，回答にかかる時間を減らすことが多くの用途で求められる．

4.1 メモリを大きくする：現状での最適な実現方法

ローカル LLM の実現という観点では，2024 年 7 月現在では Mac ユーザーが有利である．Mac を買い替える際にメモリを増設して注文すれば，ノート PC でも実現できてしまう手軽さと，メンテナンスのしやすさを得られる．LLM を作るつもりはなく，日常会話のような素早い応答は求めないならば，メモリの大きな Mac で十分である．2024 年 7 月現在の Apple Store で売られている範囲では，30 万程度の VRAM 24GB の GPU の限界を少し超えられる 64GB 以上のメモリ，できれば 96GB

3 Meta 社の Llama では年齢制限はされていない．ただし，未成年者の搾取などに繋がる行為などの禁止事項がある．また，責任ある使用のためのガイド [19] が公開されている．

以上のメモリを備えたMacが選択肢となるだろうか．デスクトップPCであればMac Studio（M2 Ultra）は50万円以内，ノートPCであればMacBook Pro（M3 Max）は80万円以内で購入できる．

それ以外の場合は，デスクトップPCにGPUを付けるしかない．Windowsでも Linuxでもよい．注意すべきはGPUが多くの電力を消費し，熱を発生させることである．電源装置が十分な出力を備えていない場合や，効率的に排熱できるようなファンの設置などをしておかないと，GPUの動作が不安定になってしまう．仮に，複数のGPUを繋ぐとなれば尚更難しくなるし，マザーボードが対応できるかも分からない．

もちろん，5年経てばVRAM 100GBのGPUが安く手に入るかもしれない．VRAMのサイズがボトルネックである事は知られつつあるので，需要過多さえ過ぎればその可能性はある．また，iPhoneの場合は最近5年間で，搭載するメモリは4GBから8GBと2倍になっているし，その前の5年間では1GBから4倍に増えている．仮に，現状の8GBから6倍くらいになり，Macのユニファイドメモリのように使えるようになれば，7bのモデルくらいならスマホでも動く可能性はあるだろう．消費電力がネックになりうるが，新たなデバイスが解決するかもしれない（4.4節）．

4.2 モデルを小さくする：量子化

モデル内での計算は小数で行われているが，単純に言えば，その小数を適当な桁数に丸めてしまうことが量子化である．一番極端な場合は−1，0，1の3種類の値に絞る（2量子化）が，3量子化，4量子化と徐々に精度を上げることができる．精度が高いほど性能は上がるが，ファイルサイズも大きい．量子化されたLLMのフォーマットはGGUFやGPTQが一般的であり，そのためのライブラリも整備されている．自分で量子化を実行してもよいが，代表的なローカルLLMについては，誰かが量子化して公開していることが多いので，それを使えばよい．

ローカルLLMの観点からは，量子化の最大の効能はモデルの軽量化である．たとえば，2量子化すればファイルサイズが10分の1程度になる．4量子化ならば4分の1程度のサイズになり，多くのタスクにおいて性能もほとんど落ちない．このため，30bや70bのような大きなモデルを試したければ，メモリに収まるような量子化されたモデルを使うとよい．

4.3 モデルのパラメータ数を抑える

100bや1,000bといった大きなLLMへのチャレンジがある一方で，パラメータ数を増やさずに性能を向上させ，3b程度のモデルが小規模言語モデル（SLM; Small Language Model）として話題になることもある．Google社，Microsoft社，Apple社など，いわゆるビッグテックも開発を進めている．

個人レベルでも，この規模であればモデルの訓練はしやすく，オープンソース的な開発に相性がいいかもしれない．アーキテクチャの改良によるパラダイムシフトの可能性もあるだろうし，学習に用いるデータを良質にする努力や，小さなLLMにより多くのデータに対して訓練できるような工夫が実を結ぶかもしれない．また，モデルの結合（3.3節）も，モデルの学習に用いられたデータの量が実質的には増えているという点で，広い意味ではデータの改善に含まれるかもしれない．いずれにせよ，この1年あまりの進化は目覚ましく，7b程度の小さなローカルLLMが場合によっては実用に耐えうるという意見も出始めている．

4.4 計算の最適化：NPUの発展

ここまで，LLMの計算はGPUが行うことを前提に記してきたが，新たな流れとしてNPU（Neural Processing Unit）による計算がある．NPU自体はLLMが今のように話題になる前の2018年頃から聞かれるようになっていたし，Apple社のNeural EngineもNPUに該当する．だが，近年はLLMの使用に最適化されたNPUが模索されているようだ．

NPUは量子化（4.2節）に関係している．単純化すれば，2量子化は1桁の精度しかもたないのだから，1桁の精度の計算に最適化されたデバイスさえあれば十分である．そうすれば，デバイスが単純になってたくさん保持でき，計算も速くなり，実行時の電気代も抑えられる．実際にはここまで極端ではないが，int8やfp16といった通常の4分の1ないし半分の精度の計算に最適化されたデバイスを各社が検討している．

ちょうど2024年6月に，Microsoft社が新世代のAI PCの指標となるCopilot+ PCを発表[22]し，その要件にはNPUの性能も含まれている．NPUが発展したうえで，たとえばCPU，GPU，NPUでメモリを共有できる[4]ようになれば，ローカルLLMの実現は一層手軽になるかもしれない．

5. 1人1LLMについて：パーソナルLLM

5.1 パーソナルLLMの実現方法

私は，2023年4月にblogに書いた記事「GPTと小中高の教育」において「1人1GPT環境」という考えを取り上げた[23]．この環境は，Apple社のティム・クックCEOが提案した「パーソナルな人工知能」がある環境に非常に近い．この提案は，Apple社のデバイス上でChatGPTを使えるようにすると発表された際の，以下の発言に見られる．

「アップルの人工知能は，あなたが一番大切に思うことを手助けできるように，パワフルで直感的で使いやすくあるべきです．何よりもあなたを理解し，あなたの習慣や人間関係，コミュニケーションなどの個人的な文脈に根ざしたものでなければなりません．それは『パーソナルな人工知能』とも言え，アップルにとって次の大きな一歩です．」[24]

では，パーソナルな人工知能はどのように提供されるだろうか．それは，以下の3つのいずれかになるだろう．パーソナルな人工知能のうち言語に関するモデル（以下，パーソナルLLMと呼ぶ）についての具体例を添えれば以下のようになる．

1. クラウド上で作成から実行までを完結する．たとえば，クラウド上に保存された個人情報や，個人端末から送られ一時的に保存される個人情報を用い，クラウド上で汎用的な

4 Apple社のユニファイドメモリは，既に，Neural Engineともメモリをシェアしているようだ．ただ，現状ではMacでローカルLLMを動かすときは，Neural EngineでなくGPUが動かしている．

LLM がファインチューニング・保存される．パーソナル LLM とのやりとりもクラウド上で実行される．
2. 個人の端末内で作成から実行まで完結する．たとえば，汎用的な LLM がダウンロードされ，端末内の個人情報やクラウドからダウンロードされた個人情報を元に，端末上でファインチューニング・保存される．パーソナル LLM とのやりとりも端末内で完結する．
3. 上記 2 つの折衷案．たとえば，汎用的な LLM のファインチューニングまでは 1. が行われ，個人の端末にダウンロード・保存されると共に，個人情報もパーソナル LLM もクラウド上から完全に削除される．パーソナル LLM とのやりとりは端末内で完結する．

ただし，ファインチューニングの一部ないし全てが，プロンプトの工夫で代替されるかもしれない．また，LLM と外部から取得した情報を組み合わせて生成する RAG（Retrieval Augmented Generative）という手法の発展にも影響を受けるだろう．

1. のタイプは，いくつかのクラウドサービスにより実質的に実現していると言えるだろう．

2. が本稿の扱ったローカル LLM の応用例である．PC が何か不正アクセスを受けなければ，プライバシーが保護され，外部からのコントロールを受けない LLM の使用である．

個人的には，折衷案である 3. が一般的になると考えている．音声アシスタントであれば，既に，ユーザーが言語や声色を選んでダウンロード・インストールする仕組みがある．さらに，端末内の個人情報を用いた推薦のサービスもある．パーソナル LLM の場合は，ファインチューニングや，プロンプトのカスタマイズや，何か別のアルゴリズムによるプロセスが加わるだけである．

5.2 1 人 1LLM 環境に備える

私が，前述の blog [23] で取り上げた，1 人 1GPT 環境に備えて考えるべき 4 点は以下のようなものである（ただし，GPT を LLM と置き換えた）．

1. 各端末で自分用の LLM は動かせるのか
2. LLM の年齢利用制限
3. 子どもや教師は，LLM から必要な答えを引き出す質問ができるのか
4. LLM の答えについて子どもや教師は検証できるか

1. が本稿の主要テーマであった．2. について，blog では「家庭で無茶苦茶な使い方をする前に，小学校で段階的に教えた方が私はいいと思う．」と記したが，今は先進的な学校を中心にそのような取り組みが適切に行われているように感じる．また，高校や大学の授業で使っていて，思ったほど LLM は簡単には浸透していかないように感じている．個人的には，その要因として，3. の問題と，LLM からの長い回答を読むのがしんどい点があると考えている．私は，これからの学校文化は，広い意味での科学的態度，言い換えれば「後日検証可能性を残そうとする志向」を涵養していくように変革が必要だと考えている [25] が，3. と 4. に取り組みながら徐々に対応されればと考えている．

5.3 ローカルであることのメリットは実は少ない

ここまでの議論をひっくり返すようであるが，実は，LLM がローカルであることのメリットはあまり多くない．

機密情報や個人情報を全く気にせず LLM を扱える点は魅力だろう．しかし，AI との会話履歴をモデルの学習には使われないよう設定できるクラウドサービスは多数ある．AES 256 と TLS1.2+により通信データは暗号化され，企業の利用に耐えうると謳われたサービス ChatGPT Enterprise もある．

様々な分野に特化した LLM が世界中のプログラマによって作られていることは心強いが，ChatGPT Plus に契約すれば ChatGPT をカスタマイズする GPTs も利用可能であり，特定の領域に特化したモデルを自ら構築できる．

LLM が偏りのある内容や誤情報を生成する可能性は，ローカルでも避けられない．たしかに，オープンソース的な LLM は，どのようなモデルにどのようなデータで訓練したのか追えるため，モデルが引き起こした問題について検討する材料を揃えることは可能である．しかし，大量のデータと深層学習のモデルに対する検討は容易ではない．

また，ローカル LLM の安全性の問題もある．公開されているモデルの安全性は誰かが使ってみないと分からない部分もある．たとえば，特定の分野だけは回答が偏る LLM が潜んでいるかもしれない．未発見の脅威，いわゆるゼロデイ攻撃には脆弱である．量子化において標準的なフォーマットの一つである GGUF（4.2 節）に脆弱性が見つかったこともある [26].

ローカル LLM を実現できたとしても，その環境の維持も容易ではない．正直なところ，プログラム初心者が GPU を自前で持って維持することはハードルが高い．少なくとも，ある程度の覚悟と，詳しい友人が必要になるだろう．

6. おわりに

本稿では，ローカル LLM の概要と，未来の 1 つとしてパーソナル LLM ないし 1 人 1LLM 環境について考察してきた．

1 人 1LLM 環境における教育に適切に向き合うには，分野を超えた議論が必要であるが，少なくとも，各自が被教育経験も教育経験も相対化する事が必要だろう．1 人 1LLM 環境で教育を受けた人は皆無であるから，自らにとって良かった被教育経験も，自らの上手くいった教育経験も，バイアスとして悪さをするかもしれない．

一方で，過去の前例を全て忘れればいいわけではなく，時代を超えて繰り返された議論の蓄積も参照されるべきである．たとえば，単に「議論が必要」として教育に持ち込めば，実はクラスに権力性を引き込む危険を孕んでいることは，教育学の中では議論されている．批判的思考の大切さもよく言われるが，話はそう簡単ではないだろう．どこまで疑えばいいのだろうか？ 人間の批判的思考は，AI の参照する圧倒的なリソースに対して優位性は維持できるだろうか？ この難題に立ち向かうには，少なくとも，情報学と教育学と教育現場が手を結んで，互いの強みを活かしながら取り組むべきだろう [23,25].

IT 業界に全く人手が足りておらず [27]，コロナ禍におけるデジタル敗戦も取り沙汰される日本の現状を思うと，いくらか気が重くはなるが，本稿が何かの助けになれば幸いである．

参 考 文 献

[1] Hugging Face: Open LLM Leaderboard, https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard [accessed Jul. 31, 2024]

[2] LifeArchitect.ai: Models Table, https://lifearchitect.ai/models-table/ [accessed Jul. 31, 2024]

[3] LLM-jp: 日本語 LLM まとめ, https://llm-jp.github.io/awesome-japanese-llm/ [accessed Jul. 31, 2024]

[4] 国立情報学研究所（NII）: LLM-jp, https://llm-jp.nii.ac.jp [accessed Jul. 31, 2024]

[5] K. Sugimoto: Exploring Open Large Language Models for the Japanese Language: A Practical Guide, 2024, https://doi.org/10.51094/jxiv.682 [accessed Jul. 31, 2024]

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin: "Attention is All You Need," arXiv:1706.03762, 2017.

[7] LifeArchitect.ai: GPT-5, https://lifearchitect.ai/gpt-5 [accessed Jul. 31, 2024]

[8] 奥村晴彦: GPT-3.5 のパラメータ数の謎, 2023, https://okumuralab.org/~okumura/misc/230613.html [accessed Jul. 31, 2024]

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei: "Language Models are Few-Shot Learners," arXiv:2005.14165, 2020.

[10] Engineering at Meta: Building Meta's GenAI Infrastructure, https://engineering.fb.com/2024/03/12/data-center-engineering/building-metas-genai-infrastructure/ [accessed Jul. 31, 2024]

[11] 国立情報学研究所（NII）: 大規模言語モデル「LLM-jp-13B v2.0」を構築 ～NII 主宰 LLM 勉強会（LLM-jp）が「LLM-jp-13B」の後続モデルとその構築に使用した全リソースを公開～, 2024, https://www.nii.ac.jp/news/release/2024/0430.html [accessed Jul. 31, 2024]

[12] A. Sasaki, M. Hirakawa, S. Horie, and T. Nakamura: ELYZA-tasks-100: 日本語 instruction モデル評価データセット, 2023, https://huggingface.co/datasets/elyza/ELYZA-tasks-100 [accessed Jul. 31, 2024]

[13] 株式会社 ELYZA: ELYZA-tasks-100 評価結果シート, https://docs.google.com/spreadsheets/d/1mtoy4QAqDPk2f_B0vDogFoOrbA5G42DBEEHdqM4VmDI [accessed Jul. 31, 2024]

[14] 菅原朔, 宮尾祐介: 評価・チューニング WG 進捗報告（2024/05/28 第 9 回 LLM 勉強会）, 2024, https://llm-jp.nii.ac.jp/resources/ [accessed Jul. 31, 2024]

[15] T. Akiba, M. Shing, Y. Tang, Q. Sun, and D. Ha: "Evolutionary Optimization of Model Merging Recipes," arXiv:2403.13187, 2024.

[16] 久富望: GPT と小中高の教育 (2/2), 2024, https://nkutomi.educ.kyoto-u.ac.jp/?p=75 [accessed Jul. 31, 2024]

[17] Meta: Introducing Llama 3.1: Our most capable models to date, 2024, https://ai.meta.com/blog/meta-llama-3-1 [accessed Jul. 31, 2024]

[18] S. Maffulli: Meta's LLaMa 2 license is not Open Source, 2023, https://opensource.org/blog/metas-llama-2-license-is-not-open-source [accessed Jul. 31, 2024]

[19] Meta AI: 責任ある使用のためのガイド, 2023, https://about.fb.com/ja/wp-content/uploads/sites/15/2023/10/Llama-2-Responsible-Use-Guide_Japanese.pdf [accessed Jul. 31, 2024]

[20] Open Source Initiative (OSI): Join the Discussion on Open Source AI, https://opensource.org/deepdive [accessed Jul. 31, 2024]

[21] 文部科学省: 生成 AI の利用について, https://www.mext.go.jp/a_menu/other/mext_02412.html [accessed Jul. 31, 2024]

[22] Microsoft: Copilot+PC, https://www.microsoft.com/ja-jp/windows/copilot-plus-pcs [accessed Jul. 31, 2024]

[23] 久富望: GPT と小中高の教育 (1/2), 2024, https://nkutomi.educ.kyoto-u.ac.jp/?p=46 [accessed Jul. 31, 2024]

[24] NHK: "iPhone に生成 AI" アップル発表 ChatGPT も利用可能に なぜ？, 2024, https://www3.nhk.or.jp/news/html/20240611/k10014476841000.html [accessed Jul. 31, 2024]

[25] 久富望: "教育におけるデータ利活用–教育 DX に向けた学校文化変革のために," 世界と日本の事例で考える学校教育 × ICT, 京都大学大学院教育学研究科教育実践コラボレーション・センター, 西岡加名恵編, 明治図書, pp. 78-92, 2023.

[26] N. Archibald: GGML GGUF ファイルフォーマットの脆弱性, 2024, https://www.databricks.com/jp/blog/ggml-gguf-file-format-vulnerabilities [accessed Jul. 31, 2024]

[27] みずほ情報総研株式会社: 人材需給に関する調査 調査報告書, 2019, https://www.meti.go.jp/policy/it_policy/jinzai/houkokusyo.pdf [accessed Jul. 31, 2024]

（2024 年 06 月 27 日 受付）

［問い合わせ先］

〒 606-8501 京都市左京区吉田本町

京都大学 教育学研究科

久富 望

E-mail: kutomi.nozomu.83e@kyoto-u.jp

著 者 紹 介

久富（くとみ） 望（のぞむ）［非会員］

2004 年京都大学大学院理学研究科修了．高校教員・塾講師等を経て，2014 年京都大学大学院情報学研究科博士後期課程（単位取得退学），2018 年より京都大学大学院教育学研究科助教（情報担当）．日本デジタル教科書学会理事・事務局長．