モバイル端末上で動作するLLMにおける消費電力最適化のための基礎実験

日本大学文理学部情報科学科-谷研究室
内田大貴　宮﨑亮輔

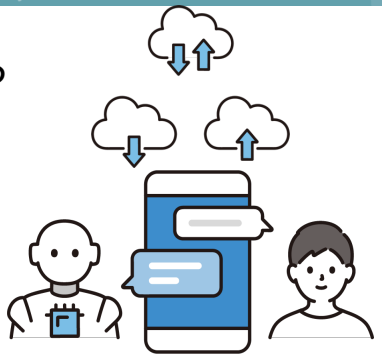
2026年1月29日　日本大学文理学部情報科学科　卒業演習発表会

目次

はじめに

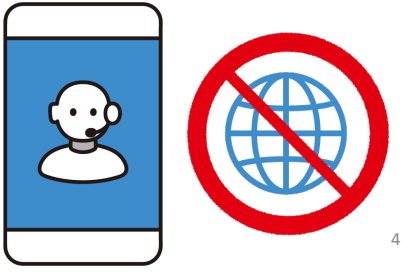
導入

今回はLLMをローカル環境で動かすことを考える



- クラウドAI（AIアプリなど）：
  入力データをWebサーバを経由して
  巨大モデルと通信し，結果を出力する

- オンデバイスAI（今回の手法）：
  クラウドAIより軽量モデルを端末内で動かし，
  データの一連の流れを端末内で完結できる



オンデバイスのメリット

秘匿情報やプライバシーの保護

送信したメッセージなどをクラウドに送信せず，デバイスの中でやり取りするため，情報漏えいのリスクが低下する

コスト削減

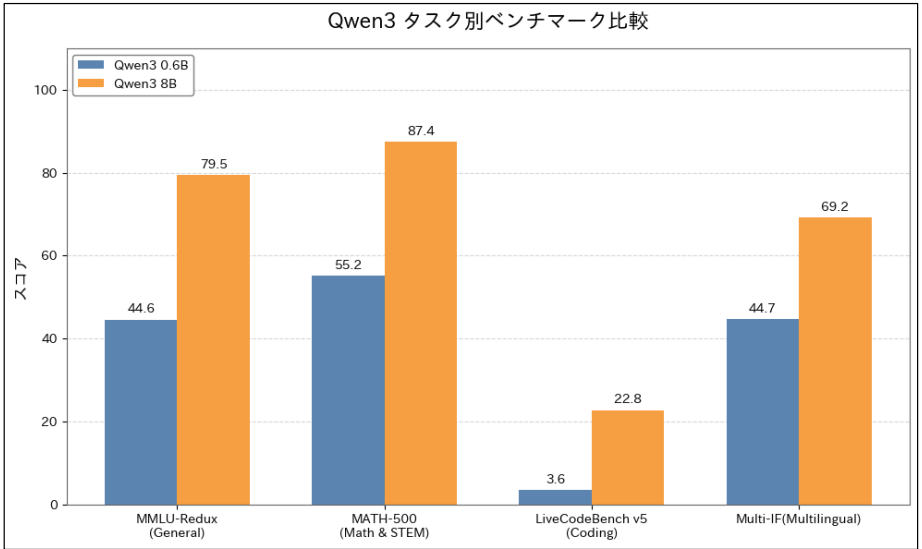
通信やトークンに関するコストを減らすことが可能

低遅延

インターネットを介さないため，通信環境が不安定な状態でもスムーズに利用可能

背景

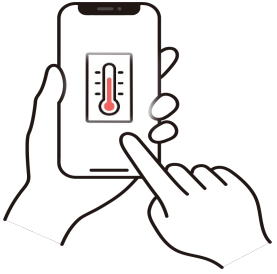
現状：サイズが大きいモデルは性能は良いが端末の消費電力が大きい
対して，小さいモデルは性能は劣るが消費電力は小さい

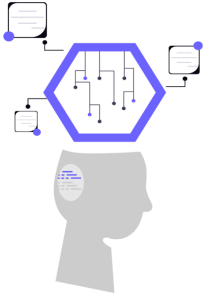


背景

課題：なるべく大きいモデルを動かしたいが，
　　バッテリーや端末への負担が大きく，長い時間稼働できない

目的：消費電力最適化と高性能なLLM利用の両立を目指す





演習の概要

バッテリーの状態に応じて２つのモデルを切り替えるシステムを試作し，
消費電力の最適化を図るための実機実験を行う

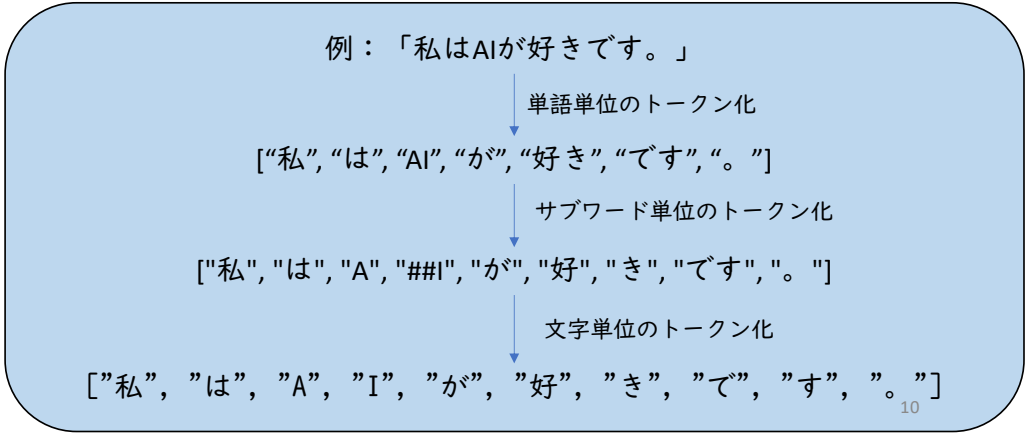
モデルA

モデルB

準備

トークン

- **LLMなどがテキストデータを処理するための最小単位**
  - テキストの入出力で許容できる文字数やコストの算出にトークンという値が使われる
  - テキストを小さな単位に分解するプロセスをトークン化という



パラメータ数

- モデル内部の「調整可能な数値（重みやバイアス）」の数
- 多いほど表現力は高いが，その分学習に大量のデータと
  計算資源が必要
- 一般的にパラメータ数が多いほど，モデルサイズも大きい

パラメータ数と消費電力

パラメータ数が増えるほど，AIモデルの実行に必要な
計算リソース（ex.計算量，計算コスト）やデータの必要量が増加

計算コストが増えることで，CPUやGPUの稼働率が上がり，
発熱やバッテリー消費が大きくなることにつながる

オープンソース

ソースコードなどの情報を公開し，他人が同じものを再現・改良できる状態にあるソフトウェアや機械学習モデル
ソフトウェアと機械学習で，公開しないといけない情報が異なる

オープンソースを公開しているプラットフォームの代表例：
Hugging Face,GitHub

画像引用:https://huggingface.co/
https://github.com/

オープンソース

ソフトウェア

- 実行に必要なソースコード
- ライセンスファイル
- ビルドの手順書（できれば）

機械学習

- 実行に必要なソースコード
- 学習に使用したデータと詳細な情報（データの入手先など）
- 学習に使ったソースコード
- パラメータ
- ライセンスファイル

出典：Open Source Group Japan（2026-01-28 参照）

オープンソースのメリット

柔軟性が高くカスタマイズがしやすい

ソースコードの構造などの情報が全て公開されているため，各人の使用用途に合わせてカスタマイズができる。

コスト削減

商用モデルの場合，契約内容により機能が制限される可能性があるが，オープンソースの場合，ほとんどのモデルを無償で使える

安全性や透明性の確保

機械学習においては、学習データの構成やモデルの重みが公開されているため，偏ったデータや表現による学習がされていないかを，第三者やコミュニティによって検証可能

デバイスの状態に応じた
モデル自動切り替えシステム

Xcode

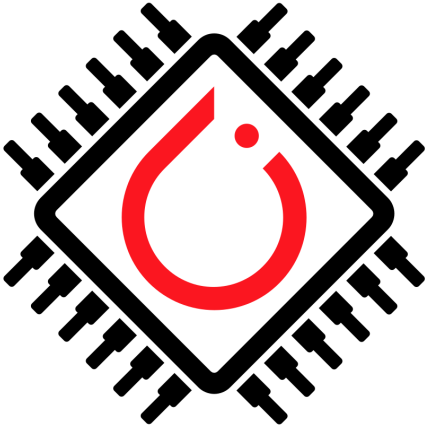
iPhoneやiPad，MacなどのApple製端末で使用する
アプリを生成する開発環境

画像引用:https://developer.apple.com/jp/xcode/

ExecuTorch

- Meta社が開発したモバイル端末でLLMを動かすためのフレームワーク
  - 限られたリソースでも動作するように，軽量化・最適化されている

ExecuTorchで動かせるモデル：
Llama, Qwen, phi, etc…



画像引用：https://docs.pytorch.org/executorch/stable/index.html

ExecuTorchのメリット

中間形式への変換が不要

通常，モデルをモバイル上で動くファイルに変換する際に，複数回変換を行う必要があるが，ExecuTorchなら，従来の変換形式を省略して変換可能

小型のランタイム

ランタイムが50KBと非常に小さいため，さまざまなデバイスで動作可能で，モデルに多くのリソースを割くことが可能

多くのバックエンドに対応可能

12種類のバックエンドに対応しているため，適切な設定を行うことで，多くのハードウェアの高速動作に対応可能

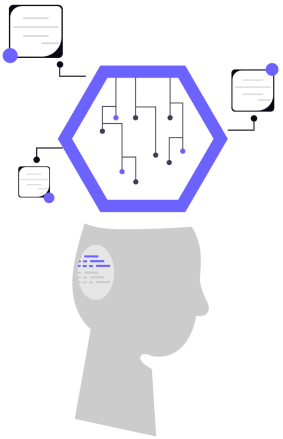
ランタイム・・・ExecuTorchで動くモデルファイルを動かすために必要な実行環境
バックエンド・・・ユーザーの操作を受け取り，裏側で処理や判断を行う仕組み

使用モデル

Qwen3 8B

切り替え

Qwen3 0.6B



採用理由

ExecuTorchに対応したモデルの中で，直近に発表されたモデル(2025年4月)であり、性能が高いため

システムの流れ

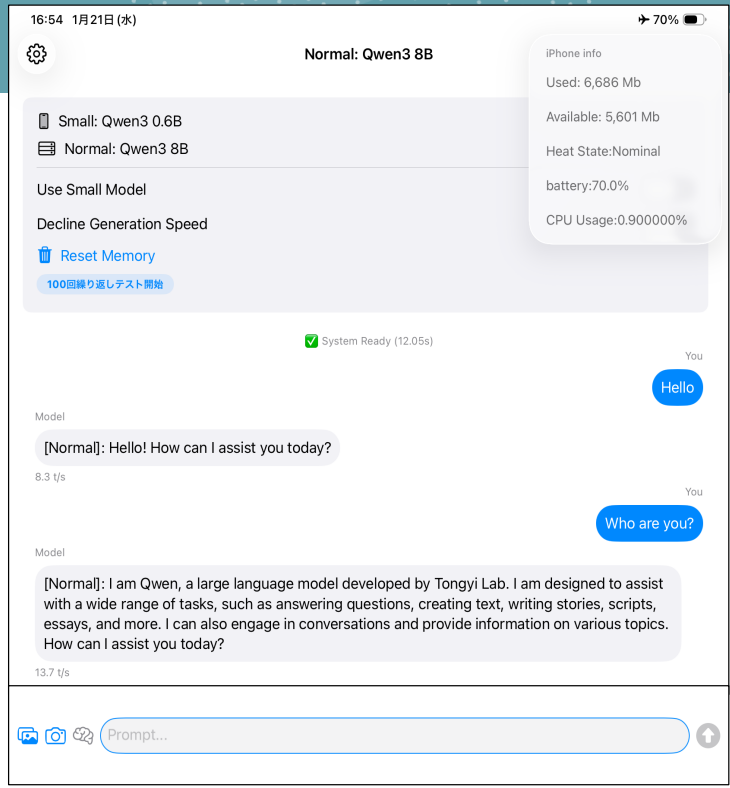
- アプリ起動
  - ホーム画面よりアプリを起動
- モデルの自動選択・ロード
  - バッテリーの状態により，使用するモデルを選択し，ロード
- メッセージ入力・送信
  - ユーザーがメッセージを入力し，送信
- モデル切り替えの判定
  - 送信ボタンを押すと，モデルを切り替えるかどうかの判定を行い，切り替える場合はメモリを解放し，ロード
- モデルの応答・文章出力
  - ユーザーからのプロンプトに，モデルが返答し，文章を出力

システム

モデルの情報 →

設定 →

モデルの応答 →



← 消費メモリやバッテリー残量などの情報

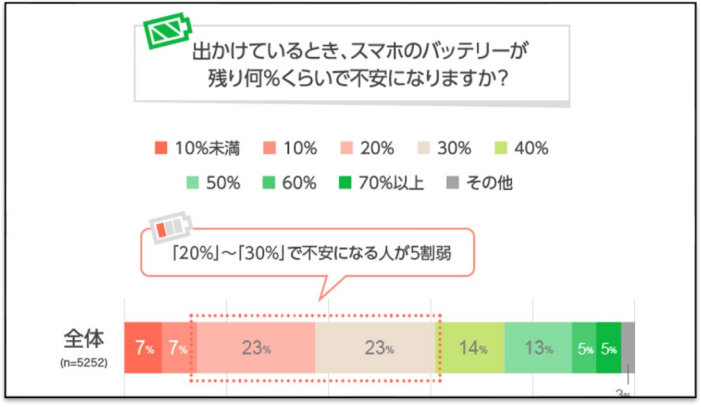
← ユーザーが投げたプロンプト

← プロンプトを入力し，送信

※GitHubに載っていた，
Xcodeファイルをもとに作成

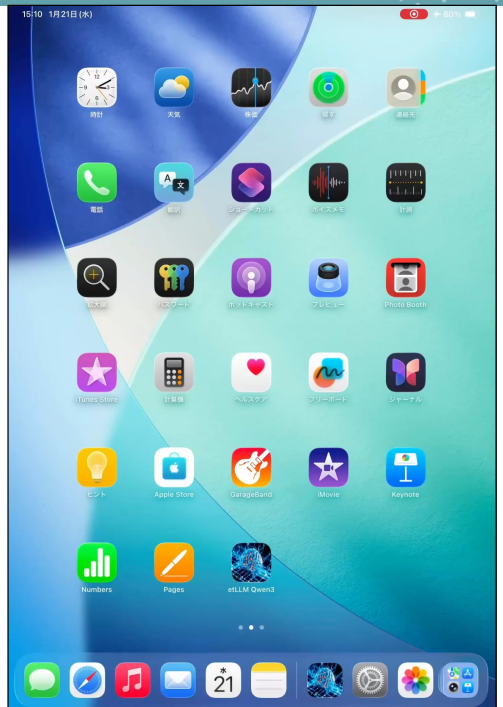
モデル切り替えや仕様変更の基準

- バッテリー残量が30%以下になる
  ⇨ 小さいモデル(Qwen3 0.6B)に切り替える
- バッテリー残量が20%以下になる
  ⇨ 生成文章の最大の長さを1024トークンから512トークンに下げる



出典：リサーチノートpoweredbyLINE.「スマホのバッテリー、残り何%で不安になる？」.https://lineresearch-platform.blog.jp/archives/37830874.html,(2026/01/23 参照)

デモンストレーション


15:10 1月21日(水)
時計
天気
株価
メモ
連絡先
電話
翻訳
ショートカット
ボイスメモ
計測
虫メガ
パスワード
ポッドキャスト
プレビュー
Photo Booth
iTunes Store
計算機
ヘルスケア
フリーボード
ジャーナル
ヒント
Apple Store
GarageBand
iMovie
Keynote
Numbers
Pages
etLLM Qwen3

消費電力比較実験

消費電力比較実験

概要：モバイル端末上でQwen3 0.6B, 8Bと本システムを動かし，
　　モデルサイズの違いが端末のバッテリー消費に与える影響を計測

環境

- iPad Pro 11-inch（2025年10月発売 A3357, M5, 12GB）
- OSバージョン：26.2
- ネットワーク：機内モード ON, Wi-Fi ON
- 画面の明るさ：最低（自動調整OFF）
- 他のアプリ：全て終了

消費電力比較実験

実験の流れ

1. 端末を80%まで充電し，発熱していない状態であることを確認

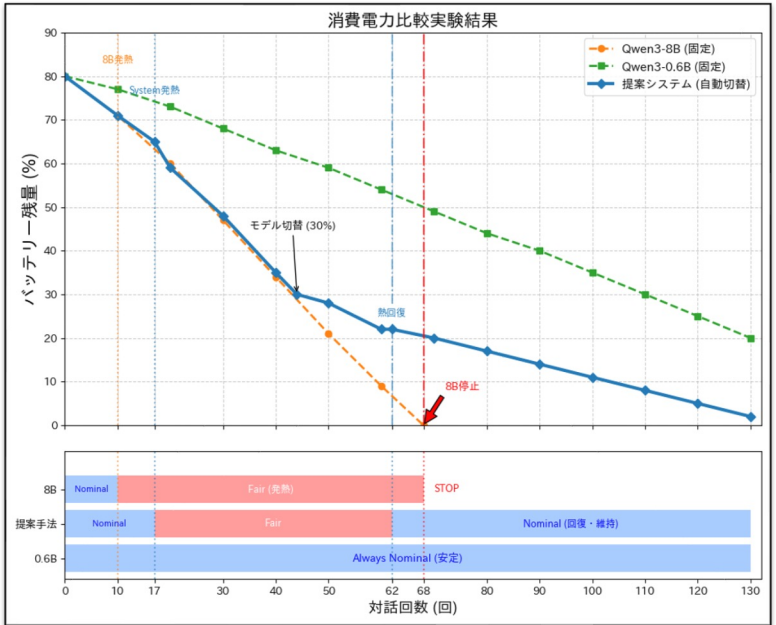
2. 1分間隔で以下のプロンプトを入力
「Write a long and detailed history of World at least 1000 words.」

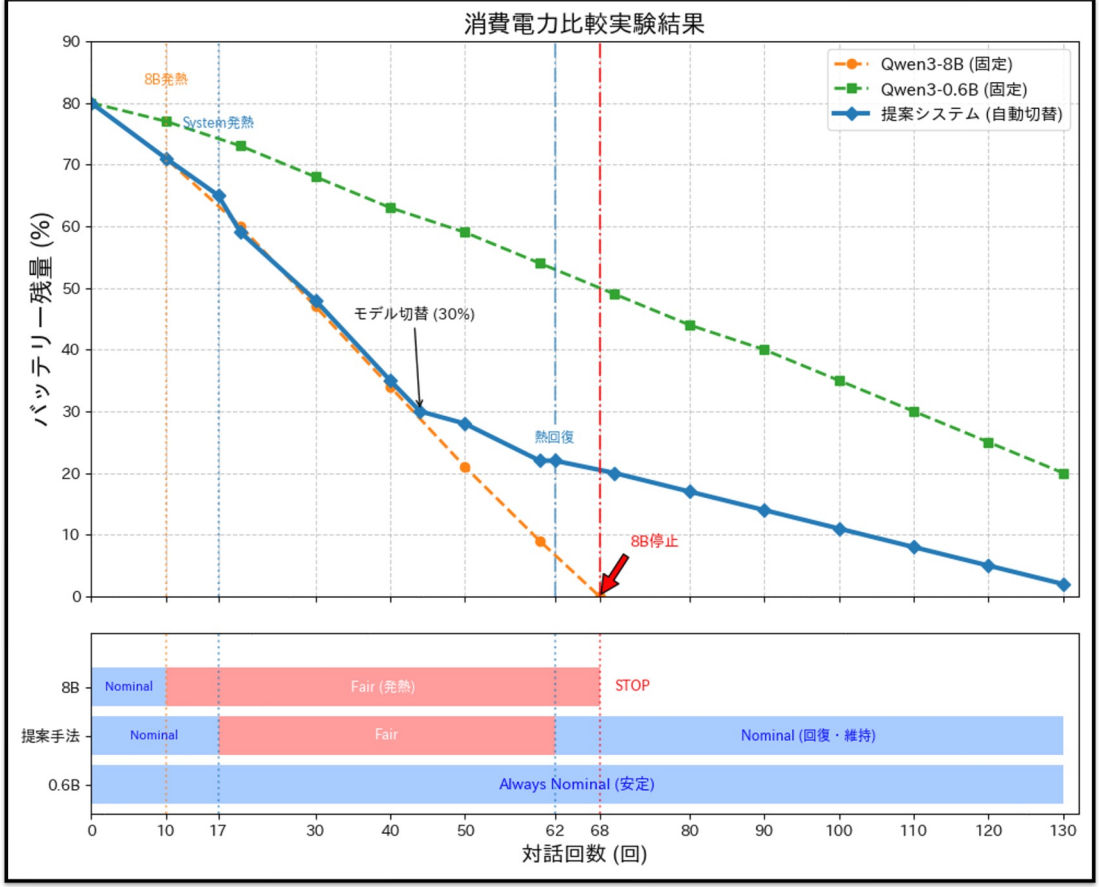
3. 10回ごとにバッテリー残量・熱状態を記録する

4. これを0.6B,8Bモデルと本システムで行い，バッテリー消費量の差を計測

消費電力比較実験

結果：両方とも開始は80%




消費電力比較実験結果
Qwen3-8B (固定)
Qwen3-0.6B (固定)
提案システム (自動切替)
8B発熱
System発熱
モデル切替 (30%)
熱回復
8B停止
バッテリー残量 (%)
8B
提案手法
0.6B
Nominal
Fair (発熱)
STOP
Nominal
Fair
Nominal (回復・維持)
Always Nominal (安定)
対話回数 (回)

考察・今後の展望

考察・今後の展望

考察

8Bモデル単体では消費電力が大きく，約70回でバッテリーが枯渇し動作不能となったため，ローカル環境で８Bモデルのような大きいモデルのみを動かすのは実用的でない

本システムでは約130回の対話を完走しており，高性能なモデルを使いつつ，バッテリー消費も削減することが可能である

負荷の低いモデルへの切り替えにより，端末での稼働を続けながら放熱することにも成功

今後の展望

バッテリー残量に加え端末の「熱状態」も切り替えの基準とし，発熱による制御も行えるようにする

現在は２種類のモデルのみの切り替えだが，中間サイズのモデルを導入し，消費電力をさらに最適化する