Local AI

Description

Large language models (LLMs) are called â??largeâ?• as they are trained on very large volumes of text data, about 570 Gb for ChatGPT 4.0â??s base model (prior to fine tuning). It sounds like a lot, but my iPhone 16 Pro has a storage capacity of about 500 Gb. Other smartphone models have twice that. So a smartphone could easily store the training text-data for a base level LLM.

The size of a model trained on text data is related indirectly to the size of the training data, but as far as I can tell, they are generally of the same order. The values of weights and biases on the nodes in the layers of a trained high performance neural network amounts to about 100 billion (100,000,000,000) parameters, each typically represented as a 32-bit floating-point number. That would take up over 500 Gb of storage.

The process of deriving inferences (i.e. responding to queries and prompts) from an LLM is computationally intensive. So, effective use of an LLM requires that the entire model is accessible to calculation as rapidly as if stored in a deviceâ??s active memory (RAM). My iPhone has just 8Gb of instant access active memory (RAM).

In order to respond to LLM-type queries and prompts my iPhone needs a processor that supports parallel processing, at least in part. My smartphone is equipped with an A18 Pro chip that features a CPU, a GPU, and a Neural Engine chip, which enable parallel processing. But data would still have to be moved in and out of active memory. Without performance sacrifices and â??workaroundsâ?• localised access to fully functional LLM inference would be prohibitively slow.

Advantages of local AI

Why aim for local AI? Running an LLM locally on your smartphone would provide substantial advantages. Notably, it would reduce the load on networks and cloud servers. You could communicate with the LLM anywhere while offline, and potentially with lower latency (i.e. with faster response times).

Thereâ??s the potential to enhance privacy as prompts and responses are kept on your device and not passed through the Internet. This also enhances â??data sovereignty.â?• Users retain control over their

personal information and conversational history without relying on third-party servers.

With enhanced processing capability at the local device, and with suitable network connectivity, LLM services can exploit â??edge computingâ?• strategies that ship heavy-duty processing tasks to high performance servers just when needed. Initial processing might occur on-device, and more intensive tasks are offloaded to the cloud, enabling responsive yet efficient interaction.

Multi-user access

Local AI reduces the overhead of providing multi-user access to a large language model. It has long intrigued me that thousands (or millions) of people can access the same model simultaneously. As it happens, a single LLM instance cannot generate responses for thousands of users simultaneously, because inference requires real-time access to large model weights held in active memory. Practical deployment of multi-user access therefore involves shared, distributed computing infrastructure.

Providing multi-user access to an LLM is conceptually simple. The obvious solution to the multi-user challenge is for the LLM server to create thousands of duplicates of the model and provide each user with access to their own version for the duration of their session. In fact, multi-user processing is more efficient (and complicated) than that.

LLM servers deploy techniques for efficiently distributing prompts from multiple users across multiple model instances. They use â??prompt queuingâ?• techniques that take account of the availability of model instances. They also split models across different processing units (GPUs) that operate in parallel. As with web search and other high use applications they also exploit distributed cloud hosting.

An LLM on your smartphone

Without the overhead of a centralised multi-user LLM, local AI can exploit dedicated and otherwise dormant capacity on your local device. As it happens, local LLMs are available thanks to the (limited) parallel processing capability of modern smartphone chips and various adjustments such as reducing the context window size and other workarounds.

ChatGPT helped explain in this paragraph how a local LLM app can deal with smartphone RAM limitations. It might seem that a local LLM app would need to load the entire model into active memory to function, but in practice this is not the case. Only portions of the modelâ??such as the weights relevant to the current computation stepâ??are loaded into memory at any given time. This makes it possible to run significantly compressed or segmented versions of models on devices with limited RAM, such as an iPhone with only 8 Gb of active memory.

Meta (Facebookâ??s) LLaMA LLM runs on a smartphone. Meta reduced the size of its model by limiting its parameter count and reducing parameter accuracy from 32 bits to 16 bits, thereby reducing model size to about 140 Gb. Other efficiencies are described in a short article by Harry Guinness.

Benchmarking LLaMA

Iâ??ve been trying out a version of the LLaMA LLM. There are several apps on the Apple App Store that enable you to download and interrogate various trained models. Iâ??ve used the *LLM Farm* and *PrivateLM* apps that will download and host a dozen or so trained models listed on a pull-down menu. I downloaded a trained LLM file called â??LLaMA-3.2-1B-Instructâ?• (under 2.5 Gb). Once downloaded I turned on â??airplane modeâ?• to sever network connections from my smartphone, and test its capabilities as a locally run AI.

It works well. On request it was able to randomize the word order of: â??I am not the author of this sentence.â?• In the process LLaMA explained why this is a paradoxical sentence. On request it presented other examples of paradoxical sentences. In other tests I asked how to transplant bamboo into a pot which it explained plausibly and with advice about watering. I also asked about Scottish water sprites, which it did in mock-Scottish dialect!

Others have conducted similar benchmark tests and with greater rigour (e.g. Hugging Faceâ??s transformers.js, MLCâ??s GitHub demos, or projects using LLaMA.cpp). Local AI seems to fare pretty well against centralised models such as OpenAIâ??s ChatGPT, paving the way for further developments in location-based AI relevant to mobile and place-based applications. See post: Mobile AI does fieldwork.

Reference

- Guiness, Harry. â??Meta AI: What is Llama 4 and why does it matter?â?• *Zapier*, 8 April, 2025. Accessed 28 May 2025. https://zapier.com/blog/llama-meta/

Category

1. Artificial Intelligence

Tags

1. distributed
2. location-based
3. smartphone

Date Created
May 31, 2025
Author
rcoyne99