

[Open in app](#)



Search



Coding Nexus

Member-only story

# Microsoft Just Declared War on the GPU Mafia: Meet bitnet.cpp

Bitnet.cpp will break the GPU Lock



Algo Insights

Follow

4 min read · 5 days ago

334

5

...

Microsoft has released Bitnet.cpp, an open-source framework that allows running large AI models on CPUs—no need for advanced GPUs.

That's huge.

For years, if you wanted to run large models, you needed costly hardware or cloud credits. Now, you can do it on the machine sitting on your desk.

Welcome to BitNet.  
How can I help you today?

What do you want to know?

CPU



By messaging BitNet, you agree to our [Terms](#) and [Privacy Policy](#).

## What is bitnet.cpp?

bitnet.cpp is the official inference framework for Microsoft's **1-bit LLMs** (such as their BitNet b1.58 models).

Instead of depending on large 16-bit or 8-bit weights, these models utilise **ternary weights**.

-1, 0, +1

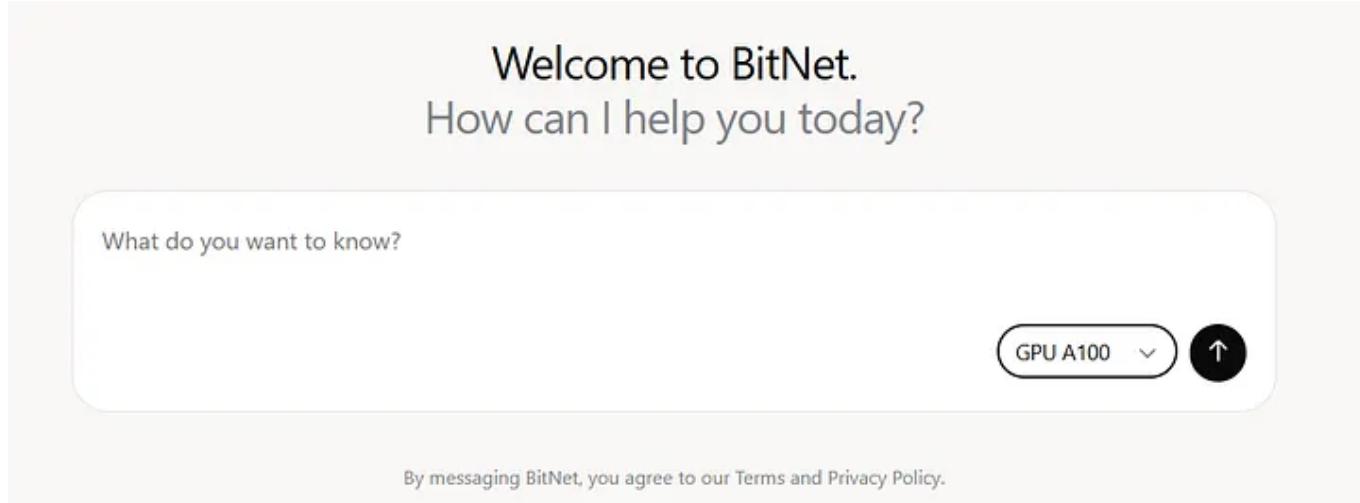
Combine that with **8-bit activations**, and you instantly reduce memory use and increase speed — without sacrificing performance.

Here's what that means in practice:

- Up to 6x faster inference

- Up to 82% lower energy consumption
- Run 100B parameter models on a single x86 CPU
- Achieve 5–7 tokens per second (approximately human reading speed!)

In short, Microsoft has made large-scale AI significantly more accessible.



## First Open-Source 1.58-bit Model

Along with the framework, Microsoft released **BitNet b1.58 2B4T** — the first fully functional open-source model that uses only 1.58 bits for weights.

Despite the extremely low precision, it still performs well on benchmarks.

That's why the release is gaining attention in the AI community — it's effective without being “dumbed down.”

Explain the Fibonacci series in the language of Srinivasa Ramanujan.

Certainly! The Fibonacci series is a sequence of numbers where each number is the sum of the two preceding ones, usually starting with 0 and 1. The series goes: 0, 1, 1, 2, 3, 5, 8, 13, 21, and so on.

Srinivasa Ramanujan, a brilliant Indian mathematician, discovered many results about the Fibonacci sequence while working on complex problems in number theory. He was known for his remarkable insights and often provided elegant solutions to mathematical problems without complete proofs.

Ramanujan's contributions to the Fibonacci sequence include the following notable results:

**1. Fibonacci Numbers and Pi:** He observed that the nth Fibonacci number is equal to  $(1/\sqrt{5}) * ((\sqrt{5}/2 + 1/\sqrt{5})^n - (\sqrt{5}/2 - 1/\sqrt{5})^n)$ . For example,  $F(1) = \pi$ ,  $F(2) = \pi/2$ ,  $F(3) = \pi/6$ , and so on.

**2. Sum of the Series:** He discovered that the sum of the first n Fibonacci numbers is equal to  $(1 - 1/2^{n+1} - 1/3^{n+1} - \dots - 1/((n-1)^2)) = 1 - 1/(2^2) - 1/(3^2) - \dots - 1/(n^2)$ . This is an approximation of  $\pi/4$ .

**3. Fibonacci Approximations:** He provided numerous approximations to  $\pi$  using Fibonacci numbers, such as  $F(1) \approx 0.5000$ ,  $F(2) \approx 0.5000$ ,  $F(3) \approx 0.5000$ , and so on.

Ramanujan's insights into the Fibonacci sequence are not only fascinating but also demonstrate the deep

How can BitNet help?

CPU



Task is completed on CPU [decode] generated tokens: 399 — total time: 13.8s — 28.913 tokens per second

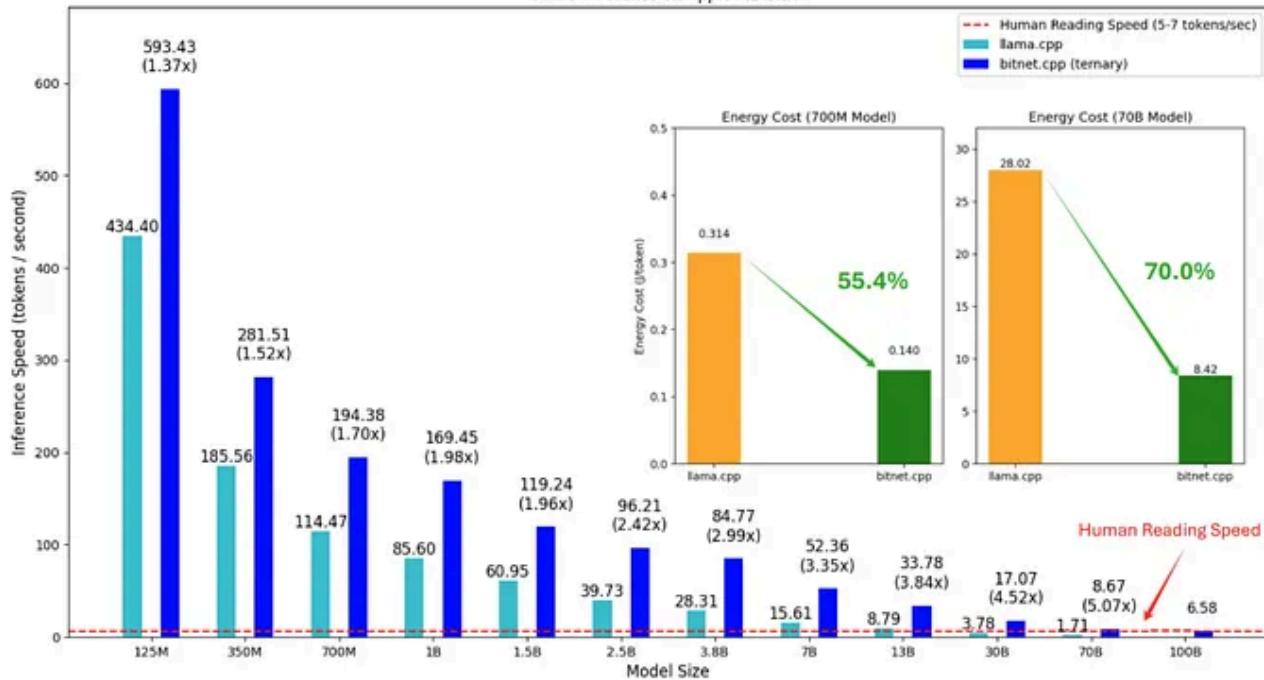
## Speed & Efficiency Benchmarks

Here's what Microsoft reported regarding Bitnet.cpp performance:

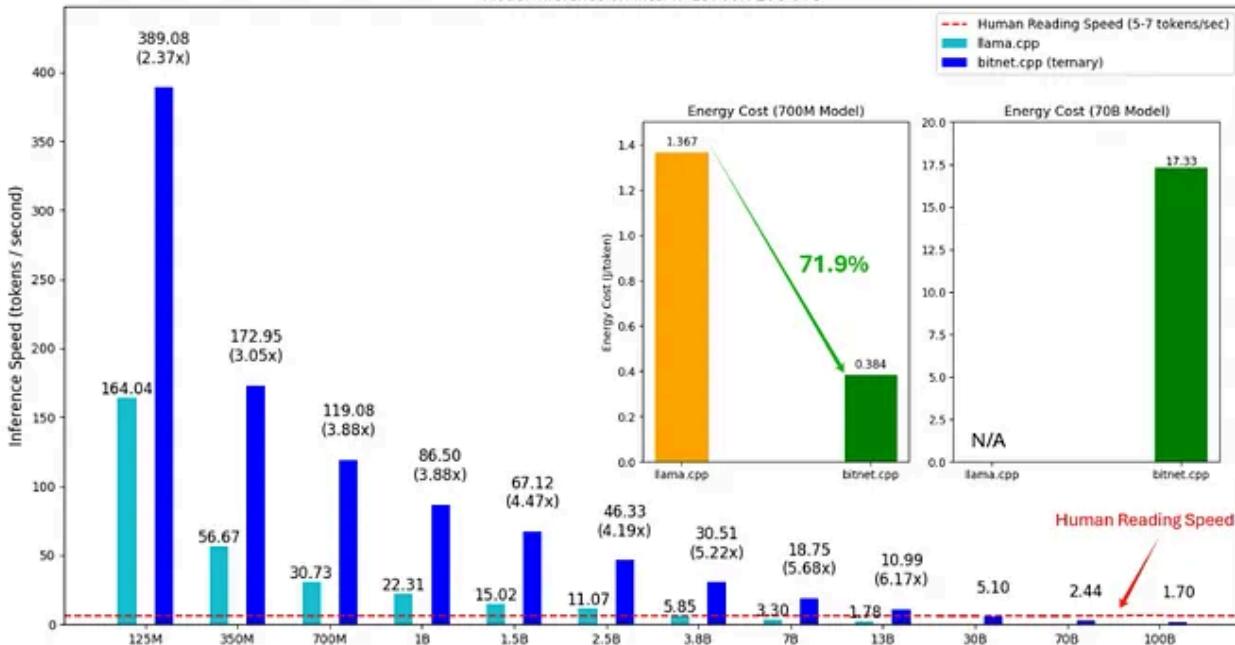
- On ARM CPUs — 1.37x to 5.07x faster + 55–70% lower energy
- On x86 CPUs — 2.37x to 6.17x faster + 71–82% lower energy

Imagine running a 100B parameter model on your laptop CPU — without damaging your hardware. That's the promise here.

Model Inference on Apple M2 Ultra



Model Inference on Intel i7-13700H 20C 64G



The tested models are dummy setups used in research to demonstrate the inference performance of bitnet.cpp.

## How to Get Started

If you're curious and want to try it out, Microsoft has made the setup surprisingly easy.

## 1. Clone the Repo

```
git clone --recursive https://github.com/microsoft/BitNet.git  
cd BitNet
```

## 2. Create a Conda Environment (recommended)

```
conda create -n bitnet-cpp python=3.9  
conda activate bitnet-cpp
```

## 3. Install Dependencies

```
pip install -r requirements.txt
```

## 4. Download the Model

```
huggingface-cli download microsoft/BitNet-b1.58-2B-4T-gguf --local-dir models/Bi
```

## 5. Set Up the Environment

```
python setup_env.py -md models/BitNet-b1.58-2B-4T -q i2_s
```

## Running Inference

Here's the easiest way to try it:

```
python run_inference.py -m models/BitNet-b1.58-2B-4T/ggml-model-i2_s.gguf \
-p "You are a helpful assistant" -cnn
```

This will start a chat-style conversation with the model.

## Benchmarking

Want to check how quickly your machine performs? Use the built-in benchmark script:

```
python utils/e2e_benchmark.py -m models/BitNet-b1.58-2B-4T/ggml-model-i2_s.gguf
-n 200 -p 256 -t 4
```

This example generates 200 tokens from a 256-token prompt using four threads.

## Converting Models

If you've got a `.safetensors` model and want to try the model in BitNet, you can convert it:

```
huggingface-cli download microsoft/bitnet-b1.58-2B-4T-bf16 --local-dir ./models/  
python ./utils/convert-helper-bitnet.py ./models/bitnet-b1.58-2B-4T-bf16
```

## Why This Matters

For years, the AI industry has been hindered by **GPU bottlenecks**.

Want to train or run a large model? Buy Nvidia cards or use cloud providers. That's the "GPU mafia" Microsoft is now targeting.

With `bitnet.cpp`, we're suddenly looking at a world where:

- Local-first AI becomes possible again.
- Developers without GPUs can still run serious models.
- Energy efficiency stops being an afterthought.

This isn't just about cost—it's about democratising access to AI.

Microsoft's **bitnet.cpp** isn't just another AI release. It's a shift in how we think about running models. By demonstrating that 100B parameters can be efficiently processed on a CPU, they show that AI doesn't have to be limited to high-end GPUs.

Repo: [BitNet on GitHub](#)

Demo: [Try it here](#)

The GPU mafia might not be gone yet, but tools like this are definitely weakening their hold on AI.

Microsoft

Gpu

AI

Ai Agent

Llm



Published in Coding Nexus

7.2K followers · Last published 3 hours ago

Follow

Coding Nexus is a community of developers, tech enthusiasts, and aspiring coders. Whether you're exploring the depths of Python, diving into data science, mastering web development, or staying updated on the latest trends in AI, Coding Nexus has something for you.



Written by Algo Insights

3.3K followers · 6 following

Follow

## Responses (5)



Bgerby

What are your thoughts?

---



Julien Chable

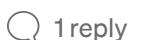
1 day ago



Nothing new, this project is one year old, and the title just to make some clicks on this article. It offers limited usages by addressing only several use cases in NLP and 'SLM' (like ultra low CPU powered devices) and with poor performance (less... [more](#))



6



1 reply



Syed Munawar Hussain Bukhari

3 days ago



Great post! Hope you'll check out mine too 😊



4



Norman Fung

10 hours ago



Oh no another Uncle Update

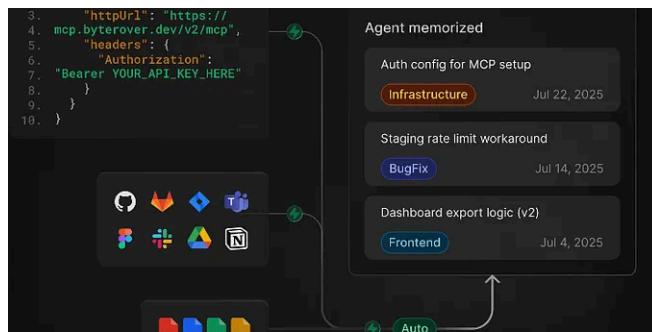


1



[See all responses](#)

## More from Algo Insights and Coding Nexus

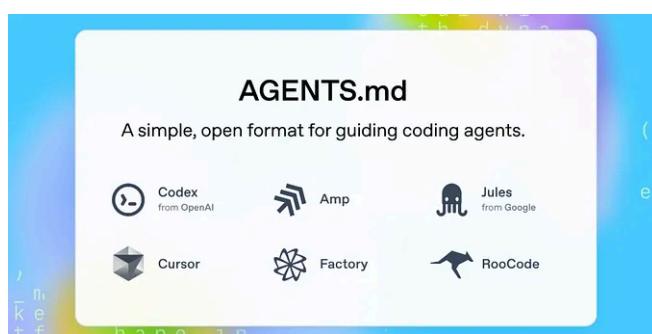


 In Coding Nexus by Algo Insights

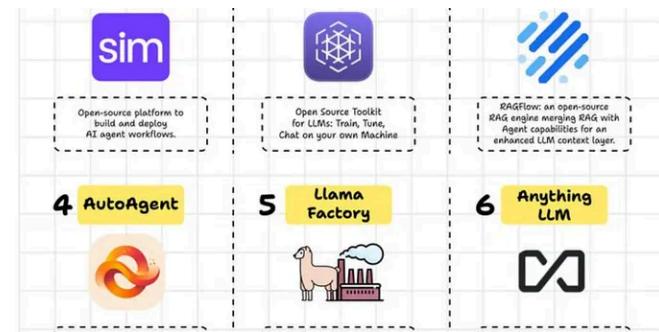
### 4 Open-Source Tools That Made Me Rethink My Dev Setup

I've been coding for a while now. Most of the tools we use every day... they've been the...

⭐ Sep 3 ⌘ 386 💬 8



 In Coding Nexus by Civil Learning



 In Coding Nexus by Civil Learning

### 6 Open-Source AI Projects You Must Try (Agents, RAG & Fine-...

The AI world is complex right now. Every week there's a new repo, a new framework, and a...

⭐ Sep 18 ⌘ 657 💬 5



 In Coding Nexus by Algo Insights

## AGENTS.md: The File That Saves You From Dumb AI Code

If you've ever thought, "This AI code is smart but also dumb," you'll get this.

★ Sep 18 564 8

W+ ...

[See all from Algo Insights](#)

## oLLM: How I Ran an 80B Model on My 8GB GPU

I didn't expect this. Running a 160GB model on a card with 8GB VRAM feels impossible.

★ Sep 30 505 6

W+ ...

[See all from Coding Nexus](#)

## Recommended from Medium



 In Data Science Collective by Erdogan T

### Build Your Private Language Model: Local and Specialized For...

A complete step-by-step guide from setup to deployment of local language models, makin...

★ Oct 3 726 4

W+ ...



 In Level Up Coding by Yasar Sandeepa

### Google's New AI – Nano Banana: Top 10 Use Cases That Will Blow...

How Nano Banana is redefining the future of AI Image Gen

★ Oct 2 208 2

W+ ...



 In Towards Deep Learning by Sumit Pandey

## Meet oLLM: The Secret Sauce to Run Huge AI on Tiny Hardware

oLLM slashes LLM memory use: Run 100k context GPTs on 8GB GPUs. A lightweight...

 Oct 2  162  9

...



 In Coding Nexus by Algo Insights

## Unsloth: Fine-Tune GPT, DeepSeek, Gemma, Qwen, Llama & Mistral 2...

Fine-tuning Large Models Without Losing VRAM

 6d ago  11 

...



 In Python in Plain English by Rizqi Mulki

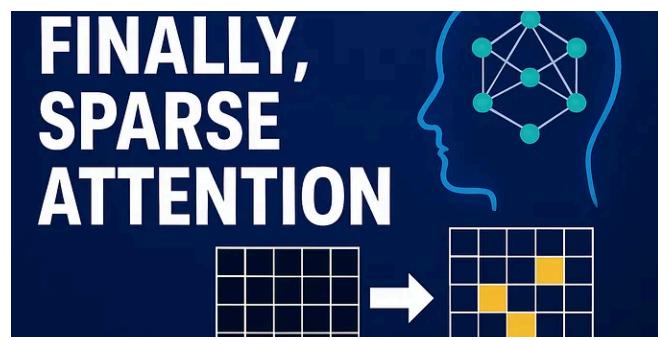
## The Hidden Python Framework That's Faster Than Node.js

For years, developers have accepted a fundamental truth: Python is slow. When...

 5d ago  68

...



 Ignacio de Gregorio

## DeepSeek is Finally Back, Solving Sparse Attention.

A Years-old Mystery, Solved

 4d ago  959 

...

[See more recommendations](#)