1-bit quantization breakthrough:

Microsoft researchers just published the most important discovery in LLMs for a long time: 1 bit is enough to store LLM weights!

When you want to make your LLM lighter and quicker, you reduce the number of bits on which each parameter is encoded. But although you can reduce your encoding from 32 to 16 bits, going to 8 or even 4 bit was really difficult and came with performance drops.

Now a bunch of Microsoft researchers decided to go full berserk:
🤔💭 "Why not encode each parameter on 1 bit?"
On top of reduced parameter size, it would have the great benefit to make matrix multiplications calculable as much faster additions!

🚫 Obviously it did not reach good performance at first: encoding all parameters as a 0 or 1 loses too much information compared to floats.

➡️ So they said "Maybe we went a bit (lol) extreme: let's backtrack and add more numbers": each parameter is now a "ternary bit" in {-1, 0, 1}. (So it effectively takes 1.58 bits)
🤯 And this works amazingly well: implemented in a 3B parameters architecture, their 1-(ternary)-bit model is competitive with a full-size 3B model.

Of course, with reduced size + the ability to add instead of multiply, this 1-bit encoding creates huge gains:
- 🏎️ **Throughput in tokens/s is x10**
- 🏋️ **Memory footprint is divided by a factor up to 7**.

I expect this architecture to take the world by storm.

Read the full paper and check for the updates here: ➡️ https://lnkd.in/env9gHdz