AI and Mathematics

Most modern computers run on 64-bit architecture—i.e., they have CPUs that can handle 64 separate binary bits of "data" (charges or lack thereof) being fed into their registers at a time. This means that the biggest exact integer they can handle is 2 to the 64th power--about 18 quintillion. This may sound large, but it is actually miniscule—infinitesimal, even--in comparison to what you can manage. In fact, in a few seconds, you could write out two numbers on a sheet of paper, both far larger than 18 quintillion (a quintillion has only eighteen zeroes), and in less than a minute, you could *directly* add them together accurately and find their *exact* sum. Your computer cannot do this.

Now, if we are instead referring to "double-precision floating-point" numbers that a 64-bit architecture can handle, this does increase the magnitude of the numbers that the CPU can crunch, but only at the cost of sacrificing the exactness and instead dealing with an extremely rough approximation. Such a number can be as much as almost 2 to the 308th power, but only seventeen or so of those hundreds of digits will be meaningful. The remaining hundreds of digits are null, and we humans are left to interpret that nullity as zeroes which, as it were, "uphold" the magnitude of large number under consideration. Even then, a human can easily manage numbers far larger than this—and not with some approximation including only seventeen meaningful digits, but with a *perfect* exposition of the entire number.

To illustrate AI's total inability to deal with large numbers that even a grade school child can manage, I asked a few of the most advanced AI systems available today to give the exact sum of two integers which were each one thousand digits long. None could do it. Some gave an approximation, others said they could not compute it, and still another gave an answer—to this extremely simple question my nine-year-old could easily get correct within minutes—that was completely wrong to a comical degree.[1]

When I fed a far easier addition problem into Microsoft Excel (consisting of less than one hundred digits for each number), it pretended to give an answer. But when I directed it to show me the digits it had hidden, they were absent; all replaced with zeroes. It had surreptitiously (giving no error message or warning) chopped off almost all the input I provided, and instead only added together the first dozen or so digits, leaving the remaining ones occluded by exponential notation.

Depending upon the question you ask of it, there are times when an AI might seem to display an understanding of these concepts, but it is in fact only regurgitating some answer it scraped from a human-authored text. In other cases, there are of course workarounds. A human programmer, cognizant of this fundamental computational limitation, can direct an algorithm to chop up a large number into parts, deal with those separately, then stick them back together, so that the end user is given the impression of the AI actually dealing directly with the large numbers, when in fact it is not.

Moreover, real math problems—word problems--always begin as *philosophy* problems, for they simply recount some potential real-life situation, which one must abstractly consider, compare to logical axioms and common sense, and only then generate a formula to model it. *After* this formula is presented, an AI can often seamlessly solve it. But generating that formula in the first place is what requires intelligence, not solving it. AI-driven LLMs (ChatGPT, etc.) are terrible at these, since they cannot abstractly ponder what is being described. Now, it often *appears* that they do an excellent job at word problems, simply because they are fed word problems whose solutions are already in their database.

It can sometimes be difficult to notice just how incompetent AI is at solving word problems since databases containing massive amounts of them (along with their solutions) heavily populate the material they scrape. Sample tests of innumerable sorts, along with their solution manuals, have long been popular internet uploads, and there are untold billions of human-authored word problems and solutions in these texts. Most LLMs have archived all of these in their own massive databases.

Therefore, if you simply feed some Chatbot a word problem you have stumbled upon in a text or on the internet, it is likely that problem already exists in its database. If you merely change the numbers or the words (e.g., "four apples" instead of "five oranges,") the AI can still use basic search-engine techniques to identify sufficient similarity and regurgitate the answer it plagiarized from a human problem solver, along with those superficial replacements. If, however, you are in the habit of feeding descriptions of actual real-world problems you come across (not basic textbook ones with available solutions in databases), the solving of which requires understanding and critical thinking, you will quickly discover that all AI systems completely lack any intelligence.

If, moreover, you happen to pose a word problem to an AI which lacks a sufficiently similar one (with a solution) in that AI's database, you will only be given comically errant results. As I was writing this section, I walked over to my bookshelf and grabbed the first

textbook I saw from my engineering studies. It happened to be one on the design of machinery. I opened a random page, selected one of its simplest problems, and entered it into one of today's most powerful AI LLMs. The question was this: "*Design a simple, spur gear train for a ratio of -9:1 and diametral pitch of 8. Specify pitch diameters and numbers of teeth. Calculate the contact ratio.*"[2]

Now, a "spur gear" is just the most basic type of gear anyone has understood since childhood. One need not be an engineer to understand some things about their operation; for example, they obviously need at least several teeth, and adjacent ones in a train of such gears rotate in opposite directions. Even a toddler playing with a plastic gear toy begins understanding this. This "ultra powerful AI," however, "solved" that problem by telling me that the pinion (the smaller gear in the simple train) should have negative nine teeth, the larger gear should have one tooth, and the contact ratio between the two will be negative 72.[3]

None of this is even coherent, much less correct. No human, no matter how ignorant of spur gear engineering, would answer a question about their design by positing that such a gear should have a negative number of teeth, or one tooth. And no human who, being told the definition of a "contact ratio" (the average number of teeth in contact with each other on adjacent gears at a given moment), would claim that this ratio could be such an outlandish ratio as "negative 72." (The ratio is generally between 1 and 2 and cannot be far from that range, and it certainly cannot be of a greater magnitude than the number of teeth!)

Observe that the problem here is not merely that we are dealing with a program that could not answer the question. The problem is that it did indeed present a "solution." It did so with the appearance of great confidence, assuring me that what it described was indeed "the design" I had requested, but as a machine, it was entirely oblivious to the fact that it had no clue what it was doing and was presenting an answer not only errant, but incoherent.

---

[1] https://dsdoconnor.com/aitest/

[2] Robert L. Norton. Design of Machinery. An Introduction to the Synthesis and Analysis of Mechanisms and Machines. 2008. 4th Edition. Page 523. Question 9-6.

[3] https://dsdoconnor.com/aitest/