Zipping physics compactly: power of tensor network

Sora Shiratani

Being exactly solvable is important: it is well-known fact in scientific fields. Then what about a similar concept: is solving exactly important? The answer is NO! In this essay, I will show you the power of “inexact” techniques.

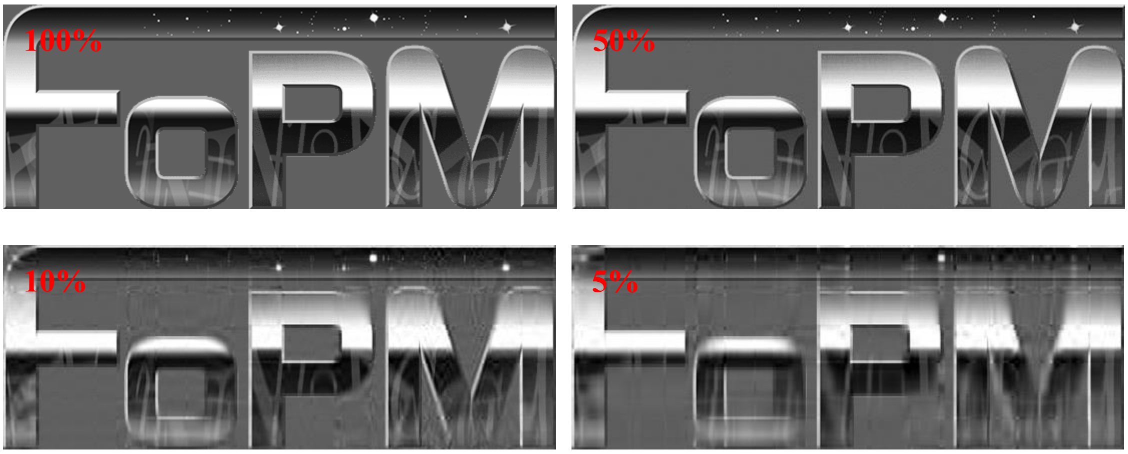
As an example, let me introduce a strong tool for “the highest-accuracy” approximation, tensor network.

Tensor is, in short, “a lot of numbers gathered together”. For example, image data is a well-known counterpart. Actually, anything associated with many numbers is, whether it appears in the physical context or not, potentially tensor.

As you might expect, tensor network is a network of tensors. It is typically expressed as a diagram: each node corresponds to a tensor, and an edge represents a relationship between two connected tensors. If you want to evaluate the network itself, just sum up all the elements of tensors according to the edge connections. Unfortunately, exact calculation is basically impossible due to computational cost.

To overcome this problem, you can rely on approximation. The most popular approximation scheme has two remarkable traits: it reduces the size of tensors at the price of numerical errors, and the errors are always minimal.

As an example, I prepared four pictures compressed by this technique, up to 100 %, 50 %, 10 %, and 5 % of the original.



You may feel that the quality of pictures, especially the latter two, is seriously impaired. Luckily, the data in physics typically keep high qualities even after compression, by the fact that few dominant components govern the overall behavior of the system, and the truncation does not affect them. The only exception is critical phenomenon: the whole components of a system correlate with each other, and this nontrivial nature results in a serious breakdown of the approximation, as the pictures shown above.

This truncation process gets more powerful when combined with the diagrammatic representation, taking the changes in the geometry into account. As an example, let me introduce the tensor renormalization group (TRG).

The TRG transforms a square tensor network (imagine an Othello board) into another smaller one tilted by 45 degrees: it exploits the hidden diagonal symmetry of the square grid, which is obvious when expressed as a diagram. Applying the process repeatedly, you can shrink the whole network into a single tensor. Truncation is essential to the TRG, because it keeps the size of tensors constant throughout the full process. The TRG contains numerical errors, but the errors are controllable by adjusting the truncation scheme, and there are many improved algorithms that give more accurate results. It is noteworthy that the TRG and its variant are also applied to other tensor networks.

Now you may be aware of the potential of tensor network formulation: it enables us to approximate or calculate intuitively and schematically.

Another strength of tensor network comes from its generality. Tensor network has certain corresponding mathematical expressions consisting of only summation and multiplication. This fact implies that you can reduce wide variety of things into tensor network formulation. For example, statistical physics, quantum mechanics, quantum computing, and machine learning are good applications, and recent research is expanding its possibility further.

Additionally, tensor network technique is model-independent for many cases. In other words, you do not have to care about the detailed values of each tensor, but only the abstract shape of network matters. This is because the truncation can be done always optimally regardless of the target tensor. It is in contrast to the Monte Carlo method, which is also a famous approximation technique, being basically model-dependent.

Another remarkable characteristic of tensor network is that the calculation itself is very fast and easy to encode. In fact, tensor contraction is one of the most efficient computational operations that sometimes reach petaflops order (10^{15} operations per second). Fortunately, you do not have to encode from scratch: all you need to do is to embed libraries in your code.

Though it is not famous now because tensor network is a relatively recent concept and calculation of it has long been too costly to calculate, tensor network will soon be one of the most frequently used numerical recipes in the world. Since the application of tensor network is extremely wide, it is not only a method born in the physics domain but also a general framework. In the future, people will use it without notice, because tensor network will be so common that it will be regarded as one of the lowest-level structures of computation as the Krylov subspace methods, the fast Fourier transformation, and the Runge-Kutta methods. I know most of the readers are not familiar with them. To emphasize that your daily life heavily depends on little-known advanced techniques, I intentionally mentioned.

In the future, tensor network may be calculated by quantum computers. As mentioned above, you can simulate a quantum circuit by tensor network, and vice versa. Like other numerical methods regarding quantum mechanics, tensor network also suffers from exponentially-growing memory consumption problems when treated exactly. This problem can be solved by quantum computing. The mutual connection between quantum computing and tensor network also implies that "classical" infamous problems can be solved by quantum computing, with the help of tensor network.

Reference:

Levin, Michael, and Cody P. Nave. "Tensor renormalization group approach to two-dimensional classical lattice models." *Physical review letters* 99.12 (2007): 120601.