



News & Highlights

Pushing Mathematical Limits, a Neural Network Learns Fluid Flow

Dana Mackenzie

Senior Technology Writer



Drop a pebble into a flowing stream of water. It may not change the pattern of flow very much. But if you drop a pebble into a different place, it may change a lot. Who can predict?

Answer: A neural network can. A group of computer scientists and mathematicians at the California Institute of Technology (Caltech) in Pasadena, CA, USA, has opened up a new arena for artificial intelligence (AI), by showing that a neural network can teach itself how to solve a broad class of fluid flow problems, much more rapidly and more accurately than any previous computer program [1].

“When our group got together two years ago, we discussed which scientific domains are ripe for disruption by AI,” said Animashree Anandkumar, a professor of computing and mathematical sciences and co-leader of the artificial intelligence for science (AI4Science) initiative at Caltech. “We decided that if we could find a strong framework for solving partial differential equations, we could have a wide impact.” Their first target was the Navier–Stokes equations in two dimensions, which describe the motion of an infinitely thin sheet of water (Fig. 1) [1]. Their neural network, which they call a “Fourier neural operator,” dramatically outperforms any previous differential equation solver on this type of problem, exceeding their speed by a factor of 400 and increasing their accuracy by 30%.

Partial differential equations (PDEs) are the kind of equation that Isaac Newton’s laws of motion naturally lead to. For this reason, they are fundamental to science, and any major advance in solving them would have broad ramifications. “We are having discussions with so many teams, from industry and academia and national labs,” said Anandkumar. “We are already doing experiments on fluid flow in three dimensions.” One good use case would be the equations for modeling nuclear fusion, Anandkumar said. Another would be materials design, she added, especially plastic and elastic materials, an area in which team member Kaushik Bhattacharya, a professor of mechanics and materials science, “has deep experience.”

Computers emerged, in part, out of efforts during the Second World War to predict projectile motion using differential equations [2]. They have been used ever since to solve differential equations, with varying degrees of accuracy and success. But previous approaches, whether they involved traditional computer programming or AI, have always worked on one “instance” of an equation

at a time. For example, they can figure out how one pebble dropped in one place affects the flow of water. Then they can learn how a pebble dropped in a different place changes it. But they will not generalize to understand how any pebble dropped in any place changes the flow. That is the ambitious goal behind the Caltech Fourier neural operator.

There is, of course, a good reason why this has not been done before. Neural networks excel at learning associations between what mathematicians call finite-dimensional spaces. For example, the Google AI program AlphaGo, that beat the strongest human Go player, learned a function between Go positions (which are finite, though astronomical, in number) and Go moves [3]. By contrast, the Fourier



Fig. 1. Water flows in a thin sheet over a fountain. The Caltech AI4Science team reports that a neural network can predict the motion of such two-dimensional fluid flow much more rapidly and accurately than computer programs using standard methods to solve differential equations [1]. Their work, which has potentially broad ramifications for advancing science through improved modeling of natural phenomena such as nuclear fusion, continues with experiments on fluid flow in three dimensions. Credit: Pixabay (public domain).

neural operator takes as input the initial velocity field of a fluid and produces as output the velocity field a certain time later. Both of these velocity fields live in infinite-dimensional spaces—which is just a mathematical way of saying that there are infinitely many ways in which you can toss a pebble into flowing water.

The Caltech team trained the Fourier neural operator by showing it a few thousand instances of a Navier–Stokes equation solved by traditional methods [1]. The network is then evaluated by a “cost function,” which measures how far off its predictions are from the correct solution, and it evolves in a way to gradually improve its predictions. Because the network starts with a curated set of inputs and outputs, this is called “supervised learning.” Google’s original version of AlphaGo learned by a combination of supervised and unsupervised learning (though a later version used unsupervised only) [3]. Other neural network programs used in image processing typically employ supervised learning [4].

But no matter how much training data you have, you might not be able to explore more than the tiniest part of an infinite-dimensional space. You cannot try out all the places where you could put a pebble into a stream. And without some kind of prior assumptions, your network is not guaranteed to correctly predict what happens when the pebble is dropped into a new place.

For this and other reasons, “We wanted to take the relevant parts of neural networks and combine them with domain-specific understanding on the math side,” said Andrew Stuart, another AI4Science team member and a professor of computing and mathematical sciences.

Specifically, Stuart knew that linear PDE—the simplest kind of PDE—can be solved with the well-known method of Green’s functions, a device used to solve difficult ordinary and PDE which may be unsolvable by other methods [5]. Basically, it provides a template for an appropriate solution to the equation. This template can be approximated in a finite-dimensional space, so it reduces the problem from infinite dimensions to finite dimensions.

The Navier–Stokes equations are nonlinear, so no such template is known for them. But if there were something similar to a Green’s function for the Navier–Stokes equation, a nonlinear but still finite-dimensional template, then a neural network should be able to learn it. There was no guarantee that this would work, but Stuart called it a “well-informed gamble.” Experience has shown time and time again that neural networks are extremely good for learning nonlinear maps in finite-dimensional spaces, he said.

Learning a nonlinear operator between infinite-dimensional spaces is a “holy grail” of computational science, said Daniele Venturi, assistant professor of applied mathematics at the University of California, Santa Cruz in Santa Cruz, CA, USA. Venturi, whose

research involves differential equations and infinite-dimensional function spaces, said he is not convinced that the Caltech group has gotten there yet. “It is in general impossible to learn a nonlinear map between infinite-dimensional spaces based on a finite number of input–output pairs,” he said. “But it is possible to approximate it. The main question is really the computational cost of such approximation, and its accuracy and efficiency. The results they have shown are really, really impressive.”

In addition to unprecedented speed and accuracy, the Caltech group’s method has other remarkable properties [1]. By design, it can predict the fluid flow even in places where you have no initial data and predict the result of disturbances not seen before. The program also confirms an emergent behavior of solutions to the Navier–Stokes equations: Over time, they redistribute energy from long to short wavelengths. This phenomenon, called an “energy cascade,” was proposed by Andrei Kolmogorov in the 1940s as an explanation for turbulence in fluids [6].

The next frontier for the Fourier neural operator is three-dimensional fluid flow, where turbulence and chaos are major obstacles. Can neural networks tame chaos? “We know that chaos means we cannot precisely predict the fluid motion over long time horizons,” Anandkumar said. “But we also know from theory that there are statistical invariants, such as invariant measures and stable attractors.” If the neural network could learn where the attractors are, it would be possible to make better probabilistic predictions, even when precise deterministic projections are impossible. Anandkumar points out that the network could control a chaotic system so that it does not head toward an undesirable attracting state. “In nuclear fusion, for example, the ability to control disruptions, such as loss of stability of the plasma, becomes very important,” she said.

References

- [1] Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, et al. Fourier neural operator for parametric partial differential equations. 2020. arXiv:2010.08895.
- [2] McCartney S. Eniac: the triumphs and tragedies of the world’s first computer. New York: Walker and Company; 1999.
- [3] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of Go without human knowledge. *Nature* 2017;550:354–9.
- [4] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014 Jun 23–28; Columbus, OH, USA; 2014. p. 580–7.
- [5] Stakgold I, Holst M. Green’s functions and boundary value problems. 3rd ed. Hoboken: Wiley Interscience; 2011.
- [6] Frisch U. Turbulence: the legacy of A. N. Kolmogorov. Cambridge: Cambridge University Press; 1995.