DOI:10.36471/JCCM_May_2020_02

AI theorist? Not yet

AI Feynman: A physics-inspired method for symbolic regression
Authors: Silviu-Marian Udrescu and Max Tegmark
Sci. Adv. 6 : eaay2631, 2020

Recommended with a Commentary by Ilya Nemenman, Emory University

One of the goals of science is in detecting statistical dependencies in experimental observations, summarizing them, and using these summaries to extrapolate: from data, we want to predict outcomes of new experiments, not yet done. Due to the "unreasonable effectiveness of mathematics in natural sciences" [1], it seems that making such predictions quantitative is the easiest if one summarizes the dependencies in terms of mathematical equations. If those are applicable broadly, we call them "laws of nature".

With the recent explosive growth in computing and artificial intelligence (AI) methods – most visibly deep neural networks – a natural question is whether one can do a theorist's job of learning laws of nature on a computer automatically. This question has created a lively field of *automated symbolic inference* – inferring mathematical equations (static formulas, differential equations, partial differential equations, and so on) describing various experimental data, or *in silico* data sets modeling real data. A few of the relevant references are [2, 3, 4, 5].

The recent publication by Udrescu and Tegmark, the subject of this commentary, is a new entry in the field. The authors compiled a set of physics formulas from the famous *Feynman's Lectures* and other classic physics textbooks. These equations relate various physical quantities among themselves. The author then took different values of variables on the right hand sides of the equations, calculated the left hand sides, and finally sprinkled the calculated values with some noise, modeling experimental errors. Their goal then was to reconstruct the original equations from such noisy "experimental" data sets. For example, knowing the force on a planet and its coordinates relative to the Sun, all recorded with some experimental error, they aimed to reconstruct the iconic Newton's law of Universal Gravitation. Their approach, which they dubbed *AI Feynman* because of the textbooks where the test equations were collected, proved to be much more accurate that competitors, and it was able to reconstruct nearly all equations in their compiled set. The main reason for their success was in effectively searching through the combinatorially large space of all possible equations (which were represented as strings of operations connecting variables). In its turn, this was achieved through incorporation of "physical intuition" into the search procedure through a few relatively straightforward steps. First, the authors pointed out that all physical variables have dimensions, and there is no point to search through equations that have dimensions mismatched. Second, they noticed that most equations in physics involve low order polynomials of the constituent variables, are formed through addition or multiplication of simple terms, incorporate relatively few elementary functions, and are almost everywhere smooth. Thus the authors prioritized the search over the space of all possible equations to first look at expressions that satisfy these biases.

However, the real golden nugget in *AI Feynman* was noticing that many physical laws obey translational, rotational, or scaling symmetries, and then implementing an explicit search for symmetry properties in the analyzed data before fitting them. To detect, say, a translational symmetry in a relation $y = f(x_1, x_2) = f(x_1 - x_2)$, one needs to notice that every time the arguments x_1 and x_2 are incremented by the same a, the value of y doesn't change. However, in any finite size, real-valued dataset, the probability that there will be two different pairs (x_1, x_2) with exactly the same difference $x_1 - x_2$ is nearly zero. To circumvent the problem, the authors relied on the power of machine learning. Feed-forward neural networks – the most common modern machine learnign paradigm – are essentially universal interpolators of smooth functions. From finite – though often quite large, $\sim 10^5$ samples or more – amounts data, the authors were able to train relatively standard neural networks to produce approximators $\tilde{f}$ of the functions being sought, e.g., $f(x_1, x_2)$ in our example, for arbitrary inputs and to a very high accuracy. One then explicitly checks not if the function f has a translational symmetry, but if its approximation does: does $\tilde{f}(x_1, x_2) \approx \tilde{f}(x_1+a, x_2+a)$ for many different pairs (x_1, x_2)? If the equality holds within some accuracy ϵ, then one only needs to search for translationally invariant functions f, which is a lot easier that searching through *all* possible functions. The authors dealt with other traditional physical symmetries very similarly.

AI Feynman is a very good illustration of how physical intuition – that is, knowing which kind of laws we expect to find – can augment brute-force machine learning. One could hope that, in the future, such combined approaches will not only detect known equations from synthetic datasets, but also new physical laws from experimental data.

However, as every work on the frontier of knowledge, this one does not solve all problems, and leaves a lot of question marks and some possibilities for improvement. For example, to discover the iconic law of Universal Gravitation, the authors needed to assign to the Newton's constant G its correct dimensions. They also needed about 10^6 training samples to detect the translational and the rotational symmetries in the law. In contrast, Newton did not know the dimensionality of G and, in fact, had to introduce this constant, and he certainly did not explicitly analyze a million data points to come up with his famous law. Another concern is that the sensitivity of the current algorithm to the introduced "experimental" noise is very high, so that the approach fails when the relative noise is often as low as 10^{-4} – something that only a few experiments can achieve.

It is also unclear whether the neural networks are an essential to detect symmetries in data. Indeed, the authors do not actually use their universal interpolation properties fully since, for example, to detect the translational symmetry, one only needs to check if the function values are (nearly) the same when both arguments are shifted by (nearly) the same a, and not by arbitrary amounts. However, this can be checked without neural networks. Indeed, when testing for the translational symmetry, one may have an access to two experimental data tuples $y_1 = f(x_1, x_2)$ and $y_2 = f(x_1 + a, x_2 + a + \delta x)$, where δx is small, but nonzero, preventing an explicit observation of the symmetry. Instead of training a network, one could have written $y_2 \approx f(x_1 + a, x_2 + a) + \left.\frac{\partial f}{\partial x_2}\right|_{\vec{x}+a} \delta x$, so that $y_2 - y_1 \approx f(x_1 + a, x_2 + a) - f(x_1, x_2) + \left.\frac{\partial f}{\partial x_2}\right|_{\vec{x}+a} \delta x$, which would reduce to $y_2 - y_1 \approx \frac{\partial f}{\partial x_2}\delta x$ if the system was, indeed, translationally symmetric. Thus detecting the presence of the symmetry is equivalent to checking for a linearity of the relation between $y_2 - y_1$ and δx for small δx – something that can be done by linear regression analysis quickly for many different values of (x_1, x_2). I suspect that this approach would be more robust than training neural networks, which is still often an art.

However, the biggest open problems, in my opinion, are the following. First, we need to understand why some equations can be discovered from just a handful of samples, and others require millions. As discussed in [3], this may be because neural networks struggle when functions they approximate have special points, which many physics laws do: for example, the law of Universal Gravitation has a divergence when the distance between the two gravitating bodies goes to zero. I suspect that understanding why and how methods such as *AI Feynman* fail (or, at least, slow down to a crawl) will be as important for discovery of new physical laws as the actual application of the algorithms to data. Indeed, when an algorithm does not find a solution for a particular new problem, this would tell us something specific about properties of the system we are analyzing, thus suggesting which approach to try next. The second – and an even harder – open problem is to understand which symmetries and other constraints one should include when biasing the search for equations. That is, we do not necessarily expect data from biological systems to be symmetric, or to be expressed in terms of low order polynomials, even if we believe that these data have a relatively simple underlying structure. What replaces traditional global symmetries in such scenarios remains to be seen.

In summary, *AI Feynman* is very effective in discovering textbook equations (for which we know that our physical intuition does hold!) from simulated data. However, there's still a lot of work to be done before *AI Feynman* would be able to compete with Feynman – or even with a lot less distinguished scientists – in finding new laws of nature.

References

[1] EP Wigner. The unreasonable effectiveness of mathematics in the natural sciences. *Commun Pure Applied Math*, XIII:1–14, 1960.

[2] M Schmidt and H Lipson. Distilling free-form natural laws from experimental data. *Science (New York, NY)*, 324(5923):81–85, April 2009.

[3] BC Daniels and I Nemenman. Automated adaptive inference of phenomenological dynamical models. *Nature Communications*, 6:8133, 2015.

[4] SL Brunton, JL Proctor, and JN Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc Natl Acad Sci (USA)*, 113(15):3932–3937, April 2016.

[5] R Iten, T Metger, H Wilming, L Del Rio, and R Renner. Discovering Physical Concepts with Neural Networks. *Phys Rev Lett*, 124(1):010508, January 2020.