Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2020-0185

Vol. **28**, No. 5, pp. 1639-1670
November 2020

Machine Learning and Computational Mathematics

Weinan E[1,2,3,i]

[1] *Department of Mathematics, Princeton University, Princeton, NJ 08544, USA.*
[2] *Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, USA.*
[3] *Beijing Institute of Big Data Research, Beijing, P.R. China.*

Received 21 September 2020; Accepted (in revised version) 19 October 2020

In memory of Professor Feng Kang (1920-1993)

Abstract. Neural network-based machine learning is capable of approximating functions in very high dimension with unprecedented efficiency and accuracy. This has opened up many exciting new possibilities, not just in traditional areas of artificial intelligence, but also in scientific computing and computational science. At the same time, machine learning has also acquired the reputation of being a set of "black box" type of tricks, without fundamental principles. This has been a real obstacle for making further progress in machine learning.

In this article, we try to address the following two very important questions: (1) How machine learning has already impacted and will further impact computational mathematics, scientific computing and computational science? (2) How computational mathematics, particularly numerical analysis, can impact machine learning? We describe some of the most important progress that has been made on these issues. Our hope is to put things into a perspective that will help to integrate machine learning with computational mathematics.

AMS subject classifications: 65Z99, 62G08, 62H99

Key words: Neural network-based machine learning, machine learning-based algorithm.

1 Introduction

Neural network-based machine learning (ML) has shown very impressive success on a variety of tasks in traditional artificial intelligence. This includes classifying images, generating new images such as (fake) human faces and playing sophisticated games such as

[i]Corresponding author. *Email address:* `weinan@math.princeton.edu` (W. E)

Go. A common feature of all these tasks is that they involve objects in very high dimension. Indeed when formulated in mathematical terms, the image classification problem is a problem of approximating a high dimensional function, defined on the space of images, to the discrete set of values corresponding to the category of each image. The dimensionality of the input space is typically 3 times the number of pixels in the image, where 3 is the dimensionality of the color space. The image generation problem is a problem of generating samples from an unknown high dimensional distribution, given a set of samples from that distribution. The Go game problem is about solving a Bellman-like equation in dynamic programming, since the optimal strategy satisfies such an equation. For sophisticated games such as Go, this Bellman-like equation is formulated on a huge space.

All these are made possible by the ability to accurately approximate high dimensional functions, using modern machine learning techniques. This opens up new possibilities for attacking problems that suffer from the "curse of dimensionality" (CoD): As dimensionality grows, computational cost grows exponentially fast. This CoD problem has been an essential obstacle for the scientific community for a very long time.

Take, for example, the problem of solving partial differential equations (PDEs) numerically. With traditional numerical methods such as finite difference, finite element and spectral methods, we can now routinely solve PDEs in three spatial dimensions plus the temporal dimension. Most of the PDEs currently studied in computational mathematics belong to this category. Well known examples include the Poisson equation, the Maxwell equation, the Euler equation, the Navier-Stokes equations, and the PDEs for linear elasticity. Sparse grids can increase our ability to handling PDEs to, say 8 to 10 dimensions. This allows us to try solving problems such as the Boltzmann equation for simple molecules. But we are totally lost when faced with PDEs, say in 100 dimension. This makes it essentially impossible to solve Fokker-Planck or Boltzmann equations for complex molecules, many-body Schrödinger, or the Hamilton-Jacobi-Bellman equations for realistic control problems.

This is exactly where machine learning can help. Indeed, starting with the work in [10, 20, 21], machine learning-based numerical algorithm for solving high dimensional PDEs and control problems has been one of the most exciting new developments in recent years in scientific computing, and this has opened up a host of new possibilities for computational mathematics. We refer to [17] for a review of this exciting development.

Solving PDEs is just the tip of the iceberg. There are many other problems for which CoD is the main obstacle, including:

- Classical many-body problem, e.g. protein folding.
- Turbulence. Even though turbulence can be modeled by the three dimensional Navier-Stokes equation, it has so many active degrees of freedom that an effective model for turbulence should involve many variables.
- Solid mechanics. In solid mechanics, we do not even have the analog of the Navier-