Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 327 (2017) 327–351

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Computational mechanics enhanced by deep learning

Atsuya Oishi[a,*], Genki Yagawa[b]

[a] Graduate School of Technology, Industrial and Social Sciences, Tokushima University, 2-1 Minami-Johsanjima, Tokushima 770-8506, Japan
[b] Professor Emeritus, University of Tokyo and Toyo University, Setagaya, Tokyo 156-0044, Japan

Available online 4 September 2017

Abstract

The present paper describes a method to enhance the capability of, or to broaden the scope of computational mechanics by using deep learning, which is one of the machine learning methods and is based on the artificial neural network. The method utilizes deep learning to extract rules inherent in a computational mechanics application, which usually are implicit and sometimes too complicated to grasp from the large amount of available data A new method of numerical quadrature for the FEM stiffness matrices is developed by using the proposed method, where a kind of optimized quadrature rule superior in accuracy to the standard Gauss–Legendre quadrature is obtained on the element-by-element basis. The detailed formulation of the proposed method is given with the sample application above, and an acceleration technique for the proposed method is discussed
© 2017 Elsevier B.V. All rights reserved.

Keywords: Deep learning; Artificial neural network; Numerical quadrature; Element stiffness matrix

1. Introduction

The core of computational mechanics has been to find approximate solutions for a variety of partial differential equations, which describe natural, physical, chemical phenomena mathematically, and the numerical methods that form the basis of computational mechanics, such as the finite element method (FEM), the finite difference method (FDM), the boundary element method (BEM), the particle methods and so on, have been employed to solve the simultaneous linear equations derived by discretization of the above partial differential equations with the elements, the lattices or the nodes.

The size of the simultaneous linear equations to be solved has been growing larger and larger. Therefore significant amount of research resources have been devoted to solve large scale systems of linear equations efficiently, and many solution techniques have been successfully developed to tackle large scale problems [1–3]. This trend in computational mechanics has been supported by the development of computers in the last decades.

Extraordinary progress of digital computers has big impacts in information science and technology including computational mechanics. Machine learning is among them [4,5]. It includes various methods and technologies, and

* Corresponding author.
E-mail address: aoishi@tokushima-u.ac.jp (A. Oishi).

http://dx.doi.org/10.1016/j.cma.2017.08.040
0045-7825/© 2017 Elsevier B.V. All rights reserved.

many of them require a large amount of computation and would be useful only when faster computers are available. This is the reason machine learning has progressed with the advance of computers.

Machine learning has also received significant impacts from the unprecedented advance of networking technologies: the Internet that connects computers all over the world and the Internet of the Things (IoT) technology, where everything has its own Internet Protocol (IP) address and is connected to each other via Internet. These technologies have made it much easier to collect enormous amount of data. Deep learning emerged to analyze the big data to recognize implicit features out of them, to find inherent tendencies in them, and to classify them into several categories, and has become widely known after it won overwhelming victory over any other machine learning technique in benchmark test to recognize various images out of a great many image data collected via Internet [6]. Since then, deep learning has begun to be utilized in various fields [7].

It is well known that some machine learning methods, including the artificial neural network (ANN), which is the core technology of deep learning, based on the synaptic neuron model [8], the genetic algorithm (GA) based on the biological evolution model [9,10], the genetic programming (GP) [11,12], the evolutionary algorithm (EA) or the evolutionary strategy (ES) [10], the Monte Carlo method (MC) based on the probability theory [4], have been tested and utilized in a variety of computational mechanics fields. These methods have added soft or probabilistically ambiguous features to the rigid feature that is inherent in the conventional computational mechanics as the numerical solutions of partial differential equations, exploring new fields in computational mechanics that could not be tractable without them. As for their applications to computational mechanics, the GAs as well as the EAs have been employed in many applications as a tool for global search [13–19]. There are also several other methods applied to computational mechanics: the proper orthogonal decomposition (POD) and the singular value decomposition (SVD) [20,21], the support vector machine (SVM) [22], the multifidelity importance sampling [23], the particle swarm optimization (PSO) and the other similar algorithms inspired by the swarm intelligence [24,25], a set of methods based on the Bayesian statistics [26–28], and various machine learning methods [29–32].

Among others, the artificial neural networks have been successfully applied to some fields in the computational mechanics [33]. The artificial neural networks can be categorized into two groups based on their architectural structure: the feed forward neural networks and the mutually connected neural networks such as the Hopfield networks. The former, often called as the multilayer perceptrons, have much more applications to computational mechanics due to their capability of approximating nonlinear mappings: the estimations of the constitutive equations of materials, the nondestructive evaluations, the structural optimizations, and so on [34–39].

The neural networks have been applied to the modeling of materials equivalent to constitutive equations of materials of various types [40–45], applications related to the constitutive models with novel training algorithms and network structures [46–49], the optimization of material composition of composites [50], the modeling of hysteresis curves in various fields [51,52]. They have also been applied to various nondestructive evaluation methods [53–59] including the ultrasonic testing [54–56] and the electrical impedance tomography [57] and so on. There are also many applications to structural optimization problems, most of which use the neural networks to reduce time-consuming evaluations of individuals that occur in the iterative optimization loop driven by global search algorithm, such as the GA or the ES [60–67].

In most of the neural networks adopted in the applications described above, the data of learning patterns and the learning epochs are relatively small and simple, meaning that the mappings to be simulated by the neural networks in these applications are plain, or they are forced to be so by narrowing the scope of the solution space. The above mentioned simplifications of problems are due to the fact that neural networks often need very long training CPU time and show poor convergence if the problem to be solved is complex.

These bottlenecks of neural networks, however, have been resolved, since computer hardware and training algorithms for the neural networks have by far advanced as to be fast enough to train large neural networks. Accordingly, the neural networks including deep learning have become applicable to much wider class of problems that have been believed to be too complex to tackle.

Viewing the current situation above, we define first the framework of computational mechanics enhanced by deep learning and then show how it can be applied to develop the optimized numerical quadrature to calculate the FEM element matrices that are impossible to be challenged without deep learning.

As is well known, this problem is essential to the finite element method and requires a great amount of computation. Many researches, therefore, have been performed seeking faster or more efficient numerical quadrature: the speedup by using shortened quadrature rules [68], a fast quadrature using sum factorization [69], a fast quadrature implemented