Author(s): Neittaanmäki, Pekka; Repin, Sergey

Title: Artificial Intelligence and Computational Science

Year: 2022

Version: Accepted version (Final draft)

Please cite the original version:

Neittaanmäki, P., & Repin, S. (2022). Artificial Intelligence and Computational Science. In T. T. Tuovinen, J. Periaux, & P. Neittaanmäki (Eds.), Computational Sciences and Artificial Intelligence in Industry : New Digital Technologies for Solving Future Societal and Economical Challenges (pp. 27-35). Springer. Intelligent Systems, Control and Automation: Science and Engineering, 76. https://doi.org/10.1007/978-3-030-70787-3_3

Artificial Intelligence and Computational Science

Pekka Neittaanmäki and Sergey Repin

Abstract In this note, we discuss the interaction between two ways of scientific analysis. The first (classical) way is known as Mathematical Modeling (MM). It is based on a model created by humans and presented in mathematical terms. Scientific Computing (SC) is an important tool of MM developed to quantitatively analyze the model. Artificial Intelligence (AI) forms a new way of scientific analysis. AI systems arise as results of a different process. Here, we take a sequence of correct input-output data, perform Machine Learning (ML) and get a model (hidden in a network). In this process, computational methods are used to create a network type model. We briefly discuss special methods used for this purpose (such as evolutionary algorithms), give a concise overview of results related to applications of AI in computer simulation of real life problems, and discuss several open problems.

Key words: scientific computations, machine learning, artificial intelligence, evolutionary algorithms

1 Classical Method of Scientific Knowledge

The scientific approach that currently dominates in science and technology is the result of a long evolution of human knowledge. It has a long history dating back to antiquity. At the core of this approach is what in modern terminology is called *model*. First models were very simple. They were intuitively motivated and verified in simple physical experiments (as, e.g., those done in Pisa by Galileo Galilei). The

Pekka Neittaanmäki (✉)
University of Jyväskylä, Faculty of Information Technology, FI-40014 University of Jyväskylä, Finland, e-mail: pekka.neittaanmaki@jyu.fi

Sergey Repin
University of Jyväskylä, Faculty of Information Technology, FI-40014 University of Jyväskylä, Finland, e-mail: sergey.repin@jyu.fi

means of elementary mathematics were quite enough to formalise such models and their verification was done by direct comparison with experimental data.

As the science developed the corresponding mathematical models became more and more complex. Being initially algebraic, they later start using differential and integral calculus. Nowadays, mathematical models are typically formed by systems that include partial differential equations coupled with integral and algebraic equations and other relations. Mathematical models often give good quantitative results that are useful for analysis and forecasting, but getting then is impossible without serious computations based on powerful computers, correct approximation methods, adaptive numerical algorithms, and error control. Therefore, the complexity of mathematical models have generated a new scientific direction, *computational science*, which is also called scientific computing (SC). It is an essential part of the classical scientific method focused on a computational model and analysis of numerical results. We can define it as follows:

> Scientific computing is a multidisciplinary field that uses advanced computing capabilities to understand and solve complex mathematical problems.

Currently, SC is a huge scientific field, which includes a wide range of subdirections. Among the most important we mention:

- Computer simulation methods for various mathematical models;
- Optimization and optimization control;
- Analysis of inverse problems;
- Signal analysis;
- Data-analysis and decision-making support;
- Logistics and combinatoric problems;
- Verification and validation of computer programs;
- Computational finance;
- Computational biology;
- Computational science in engineering.
- Big data and cloud resource management

It should be outlined that using mathematical models and scientific computations leads to several difficult problems related to the *reliability* of the results. In the recent decades, new methods have been developed to control errors that occur when replacing a continual mathematical model by a discrete model and errors arising in the process of computations (see, e.g., Neittaanmäki and Repin [13], Mali et al. [12] and references cited therein). For well-verified mathematical models and clearly defined data this method of scientific analysis is quite suitable.

SC is an essential part of the *classical way of scientific analysis* schematically presented in Fig. 1. Human is at the center of the scheme. A researcher (or a team of researchers) creates a Mathematical Model formulated in terms of a specially constructed virtual world of notion and symbols called *Mathematics*. Humans perform

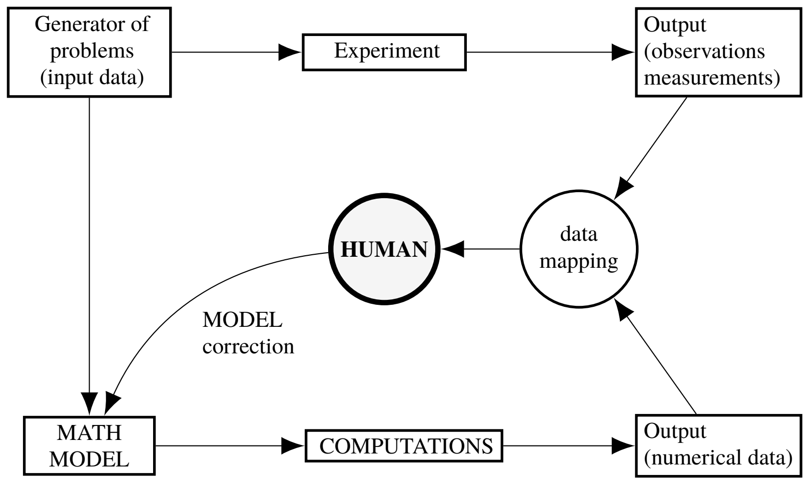

Fig. 1 Mathematical modeling

measurements/observations and develop computational methods for the mathematical model. Also, they compare results obtained in physical and numerical experiments and modify the model if it is required. In the classical scheme (Fig. 1), computer simulation methods play an important but subsidiary role, while the most crucial part related to the model is completely in hands of humans. The figure depicts one cycle of a repeatable process, which starts with an original mathematical model based on certain theoretical analysis. Certainly, the model can be specified and modified, but this is exclusively a matter of the researcher. After the model has passed the necessary verification, it can be used independently so that Mathematical Model and Computations replace the Experiment.

The scientific method has been using for many centuries and will undoubtedly be successfully used further. If we talk about complex and highly intellectual problems, then at present and in the near future no competitors are foreseen for this *modus operandi*. Nevertheless, there are many other important and interesting tasks for the study (and use) of which a different approach can be applied.

2 Genetic Algorithms

Classical methods of scientific analysis (including mathematics, physics, and other natural sciences) emerged in the process of technological development, which can be viewed as a unique way of adaptation invented by humans. All other Earth species use different adaptation methods focused on a strongly limited amount of problems essential for survival and expansion. Nervous systems of lower animals are networks trained to effectively address these basic issues: nutrition and reproduction. Despite a relatively small size, they cope with the tasks quite effectively.

It is not surprising that the idea of transferring the technology developed by life-forms to manufactured objects came to many scientists and engineers. A science called *bionics* (or *biomimetics*) studies possible applications of "technological solutions" encompassed in biological objects to various engineering problems (see, e.g., Ball [1]). A form of this idea in application to computational sciences is as follows: *Combine methods of computational mathematics with principles of natural selection*.

First works in this direction have appeared in the mid 60s. They were mostly concerned with relatively simple optimization problems solved by the so–called *evolutionary algorithms* (EA) (also called genetic algorithms). These methods use selection principles analogous to those regulate successful evolution of spices (see, e.g., Beyer et al. [3], Schwefel [19], Fogel et al. [7], Holland [9]). A genetic algorithm operates with "populations" and generates a new population by modification of the current population and selection of those "individuals" that are most acceptable with respect to a certain selection principle. Each successive population is considered a "new generation". The selection principle is defined such that only the "individuals" that satisfy it can be considered as solutions of the problem studied. Hence, the core of this method is *an artificial system of competitors, which consists of mathematical objects*.

There are many different modifications of genetic algorithms that differ in how a "generation" is changed (from simple stochastic disturbance to crossbreeding of most successful individuals) and how a new set of individuals is formed. One of the first areas of application was the problem of minimizing functions, where the criterion for selection is the value of this function. Despite the large number of different studies of a complete theory of evolutionary algorithms has not yet been created. In the majority of cases, the algorithms are purely heuristic and the results related to convergence and its rate are very rare. Moreover, EA are rather slow and cannot compete with well-known deterministic algorithms (if for the corresponding optimization problem such an algorithm is applicable).

Sometimes, EA are considered as a special class of probabilistic optimization methods, which is worth trying only if no other method can be used. This viewpoint seems to be too limited. There is no doubt that EA have a much wider scope than optimization. From a mathematical point of view, they are closer to discrete dynamical systems and have some typical features, e.g., a sequence of "populations" may not converge to the desired solution (as in a deterministic algorithm) but fluctuates around it. The latter behavior is typical for a dynamic system at the vicinity of an attractor.

Invention of EA have made an important step towards AI technologies because they solve complicated problems of very different origin without using deep mathematical models. Not being very effective for well-studied classes of optimization problems, they form a basis for new methods commonly used in machine learning of networks.

3 AI and New Methods of Scientific Analysis

A strict and commonly accepted definition of *artificial intelligence* (AI) does not yet exist. Sometimes it is understood that AI arises in systems that are able to find acceptable solutions to complex problems in conditions of incomplete information. Such type problems arise in the recognition of images, classification of objects, decision making, and many other areas. The ability to analyze and solve such problems determines the *cognitive properties* of the system. Therefore, sometimes the question of the presence or absence of AP in a a system is associated with the level of its cognitive abilities. Biological systems often have a high level of recognition in areas where it is essential for survival.

Artificial neural networks (ANN) are inspired by ideas of biomimetics. (These issues are examined in depth and in detail in Kriesel [10], Sutton and Barto [20]). However, the question of where the border between intelligent and non-intelligent systems lies remains open. If we talk about technical systems, then we can offer the following feature, which allows us to select a system with AI: it is impossible (or very difficult) to establish why such a system gave a concrete answer. Moreover, often when reprocessing exactly the same initial data, the system can give a somewhat different answer. Computing systems without intellectual properties have completely different properties: they act according to completely defined rules and always produce the same result for the same input data.

In modern literature, methods for creating artificial systems with some intelligence (in the above sense) are called *machine learning* (ML). Image recognition and inverse problems were the first classes of real life problems where these methods have demonstrated high efficiency (see, e.g., Bubba et al. [4], Kutyniok et al. [11] and the references cited in these publications). Recently, similar approaches have been used as new tools of scientific computing and mathematical modelling, in particular, for getting approximations of differential equations (see E et al. [6], He et al. [8], Regazzoni et al. [15], Raissi and Karniadakis [14], Rudd and Ferrari [16], Ruthotto and Haber [17], Trehan and Durlofsky [22], Wu et al. [23]), quantitative analysis of energy type mathematical models in mechanics (Samaniego et al. [18], E and Yu [5], Teichert et al. [21]), automatic differentiation (Baydin et al. [2]), and optimization of a robotic system (Zohdi [24]).

Here, the principal scheme of data processing and generation of a "model" differs essentially from the scheme in Fig. 1. In essence, human participation is limited to three things. They are:

1. *network structure* and its size, i.e., the researcher defines a certain class of models among which a suitable model must be found (e.g., in many cases the class of *deep neural networks* (DNN) having multiple layers of neurones is used);
2. the *quality criterion* used for comparing the actual (correct) output data with the data generated by a network. Typically, the quality criterion is a version of the least squares principle (see, e.g., Bubba et al. [4], He et al. [8], Sutton and Barto [20]);

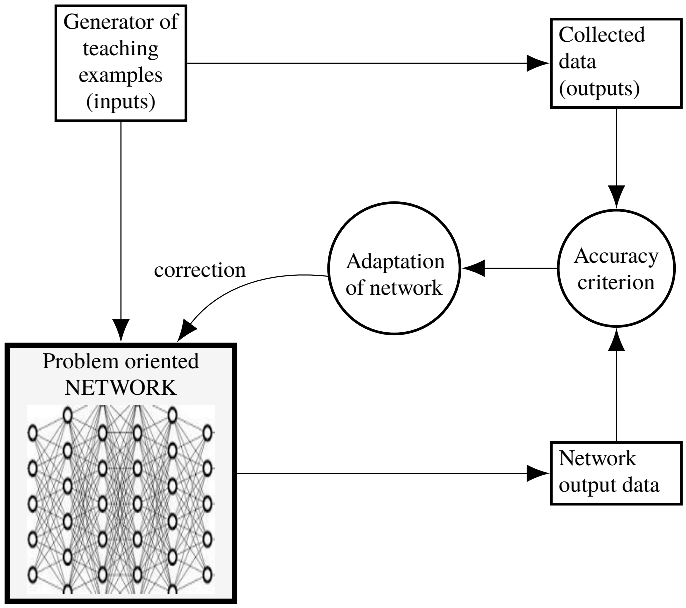

Fig. 2 Network modeling

3. the *adaptation or optimization algorithm* used in the iterative procedure of network changing (methods of nonlinear programming and structural optimisation are often used as the basic tools).

In the case of supervised machine learning method, we also need a sufficiently wide set of actual data that can be used in the *teaching process*. Therefore, an important component of this technology is the formation of such a set.

The scheme in Fig. 2 presents one cycle of the learning process: generator of input data. Usually the learning process occurs fully automatically without human intervention. As the result, we obtain a network, which structure and weights are adapted to solve a particular problem. Thus, machine learning creates a model presented in terms of the network structure.

A comparison of Figs 1 and 2 shows the fundamental differences between the two methods of analysis. The main idea of the Mathematical Modelling (and its part Scientific Computing) is: *Correct mathematical model supplied with a proper computational method provides the required information about a process or phenomenon*. The main concept of ML is rather different: *Using a sufficiently representative set of correct input–output data create a network type model of a process or phenomenon.*

4 Open Problems and Development Prospect

AI methods are at the very beginning of their development and a unified theory of them is yet to be created. We believe that such a theory will appear on the crossroad of the discrete mathematics, group theory, representation theory, and computational mathematics. Below we briefly discuss three open problems, which have a fundamental meaning for understanding of what is AI and how to create AI systems. To the best of our knowledge they are not solved. New results related to the problems could form a basis of the forthcoming theory of AI.

Problem of Teaching
The learning process is usually related to the optimization of a non-linear functional (set of functionals) having large dimension. Usually this functional is nonconvex and has a complex structure with numerous local extrema and stationary points. Therefore, known minimization algorithms (based on additional assumptions such as, e.g., convexity or unimodality) may be not enough efficient and we do not know how to guarantee that the teaching process used is indeed efficient and generates a network with the structure close to the best possible (among other networks of the same size).

Selection of a Suitable Class of Networks
One of the major tasks to be solved first of all is to set the structure and size of the network, which we are going to train. The problem is how to adjust topological structure/parameters of a network to the complexity of the problem in question. It would be nice to have some quantitative a priori criteria able to define the amount of layers and neurones depending on the amount of output parameters, variability of the data, and desired accuracy.

Why a Neural Network Model Works?
If a network has a simple structure and consists of a small amount of elements, then it is usually possible to understand why it is functioning correctly. However, simple networks can match only simple problems. Future development of AI technologies will inevitably lead to very complicated networks that will be created for analysis of serious scientific (technological, medical, social) problems. It is quite probable that some advanced AI systems will work more effectively than standard methods of mathematical modelling. In this case, scientists will be interested to understand why a particular AI system works better than their models. To answer this question it is necessary to "decode" the information encompassed in the "black box" structure and translate it into notions and terms available for humans. This problem does not attract much attention now, but may become one of the fundamental problems in the close future. We must admit that there is no guarantee that it will always be solvable.

5 Conclusions

Machine learning uses various methods of scientific computing, but as technical appliances only. In principle, "learning" and generation of an intellectual system can be performed without SC (as, e.g., in the world of nature). Therefore, AI is principally not reducible to scientific computing and cannot be viewed as its part. At the same time, the development of AP methods cannot in any way lead to the fading of interest in classical methods of computational mathematics.

Newly emerging methods of scientific analysis will use certain combination of the classical mathematical modelling (for the parts related to well-defined and well-studied subproblems) and artificial neural networks trained to simulate processes or effects where there is no complete information or suitable theory. Therefore, complications of AI networks will stimulate the interest to certain classes of computational methods. Moreover, they will generate new interesting problems for mathematical and numerical analysis. We believe that hybridised AI-SC technologies (combining the giant scientific potential of applied mathematics with huge potential of genetic algorithms and machine learning methods) will shortly form the mainstream of the technological development.

References

[1] P. Ball. Life's lessons in design. *Nature*, 409:413–416, 2011.

[2] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: A survey. *J. Mach. Learning Res.*, 18: 1–43, 2018.

[3] H.-G. Beyer, H.-P. Schwefel, and I. Wegener. How to analyze evolutionary algorithms. *Theoret. Comput. Sci.*, 287(1):101–130, 2002.

[4] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen, and V. Srinivasan. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Problems*, 35(6):064002, 2019. https://doi.org/10.1088/1361-6420/ab10ca.

[5] W. E and B. Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat.*, 6:1–12, 2018.

[6] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.*, 5:349–380, 2017.

[7] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence Through Simulated Evolution*. Wiley, New York, 1966.

[8] J. He, L. Li, J. Xu, and C. Zheng. ReLU deep neural networks and linear finite elements. *J. Comput. Math.*, 38(3):502–527, 2020.

[9] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Univ. Michigan Press, Ann Arbor, MI, 1975.

[10] D. Kriesel. A brief introduction to neural networks. Available at http://www.dkriesel.com, 2007.

[11] G. Kutyniok, V. Mehrmann, and P. C. Petersen. Regularization and numerical solution of the inverse scattering problem using shearlet frames. *J. Inverse Ill-Posed Problems*, 25(3):287–309, 2017.

[12] O. Mali, P. Neittaanmäki, and S. Repin. *Accuracy Verification Methods: Theory and Algorithms*. Springer, Berlin, 2014.

[13] P. Neittaanmäki and S. Repin. *Reliable Methods for Computer Simulation: Error Control and A Posteriori Estimates*. Elsevier, New York, 2004.

[14] M. Raissi and G. E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *J. Comput. Phys.*, 357:125–141, 2018.

[15] F. Regazzoni, L. Dedè, and A. Quarteroni. Machine learning for fast and reliable solution of time-dependent differential equations. *J. Comput. Phys.*, 397:108852, 2019.

[16] K. Rudd and S. Ferrari. A constrained integration (CINT) approach to solving partial differential equations using artificial neural networks. *Neurocomputing*, 155:277–285, 2015.

[17] L. Ruthotto and E. Haber. Deep neural networks motivated by partial differential equations. *J. Math. Imaging Vis.*, 62:352–364, 2020. https://doi.org/10.1007/s10851-019-00903-1.

[18] E. Samaniego, C. Anitescu, S. Goswami, V. M. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, and T. Rabczuk. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Comput. Methods Appl. Mech. Engrg.*, 362:112790, 2020.

[19] H.-P. Schwefel. *Numerical Optimization of Computer Models*. Wiley, Chichester, 1981.

[20] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

[21] G. H. Teichert, A. R. Natarajan, A. Van der Ven, and K. Garikipati. Machine learning materials physics: Integrable deep neural networks enable scale bridging by learning free energy functions. *Comput. Methods Appl. Mech. Engrg.*, 353:201–216, 2019.

[22] S. Trehan and L. J. Durlofsky. Machine-learning-based modeling of coarse-scale error, with application to uncertainty quantification. *Comput. Geosci.*, 22:1093–1113, 2018.

[23] J.-L. Wu, R. Sun, S. Laizet, and H. Xiao. Representation of stress tensor perturbations with application in machine-learning-assisted turbulence modeling. *Comput. Methods Appl. Mech. Engrg.*, 346:707–726, 2019.

[24] T. I. Zohdi. Dynamic thermomechanical modeling and simulation of the design of rapid free-form 3D printing processes with evolutionary machine learning. *Comput. Methods Appl. Mech. Engrg.*, 331:343–362, 2018.