

How Modelling and Simulation Can Benefit the Most from Machine Learning

Matthias Wastian^{*}, Dominik Brunmeir

dwh Simulation Services, Neustiftgasse 57–59, 1070 Vienna, Austria; ^{*}matthias.wastian@dwh.at

SNE 27(4), 2017, 209–212, DOI: 10.11128/sne.27.sn.10397
 Received: December 2, 2017, Revised: December 18, 2017,
 Accepted: December 22, 2017
 SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
 ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This article is about possible intersections between modelling and simulation and machine learning. It introduces some basic ideas of machine learning and shows how they can be applied onto traditional modelling and simulation processes in such a way that both approaches benefit as much as possible from each other. New developments and advancements in various areas allow for new techniques which may help in understanding complex dynamic systems.

Introduction

Before we talk about what we believe are the most interesting applications of machine learning methods in modelling and simulation nowadays, we shortly introduce some basic definitions. Then we give an overview about how machine learning methods can help to optimize or approximate models as well as how machine learning can be used within a model to improve it, e.g. by making agents in an agent-based model more intelligent.

1 Fields of Interest

1.1 Machine Learning and Data Science

Tom M. Mitchell provided a widely quoted, formal definition of the algorithms studied in the machine learning field: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."* [1]

Machine learning and statistics are closely related fields. According to Michael I. Jordan, the ideas of machine learning, from methodological principles to theoretical tools, have had a long prehistory in statistics. He also suggested the term data science as a placeholder to call the overall field.

Typical machine learning tasks include:

1. Mostly supervised learning tasks:

- Classification: building models to predict class labels for new observations
- Regression: building predictive models for continuous observations

2. Mostly unsupervised learning tasks:

- Clustering: segmenting data into natural subgroups
- Dimensionality reduction: mapping high-dimensional data into a lower-dimensional space, e.g. word embeddings.

Commonly used machine learning algorithms are listed below:

- Nearest neighbors
- Decision trees
- Artificial neural networks
- Support vector machines
- Bayesian networks
- Hierarchical clustering
- Genetic algorithms

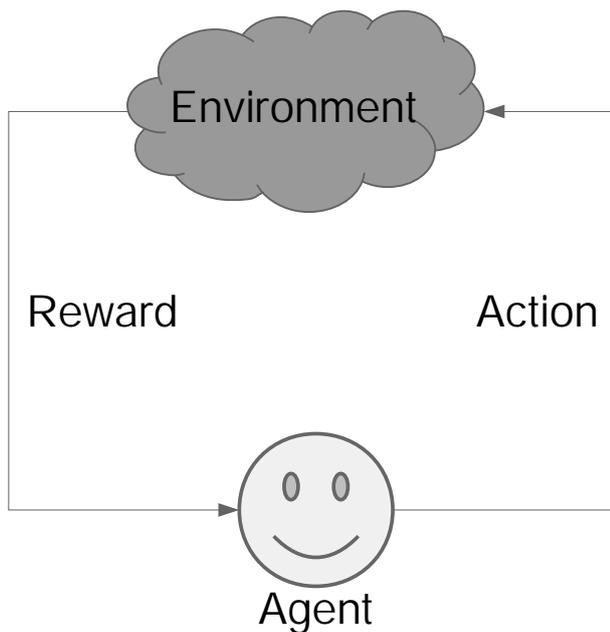


Figure 1: Reinforcement Learning Loop

1.2 Reinforcement Learning

Reinforcement learning is a part of machine learning that focuses on learning optimal control in a partially observable markov decision process. This means teaching a software agent an optimal decision policy using rewards and penalties as signals. Richard S. Sutton [8] classifies it as neither supervised nor unsupervised but as an own branch of machine learning. The main difficulty is to balance the need for exploration and exploitation. Exploration focuses on finding new ways to gain rewards, while exploitation is necessary to utilize known ways to gain rewards. A reinforcement learning problem generally consists of the following parts:

- policy
- reward signal
- value function
- model (optional)

The policy defines the action an agent takes in a given state. This is what an agent learns in a classical reinforcement learning problem. The reward signal gives feedback to the agent after each time step. It rewards or punishes the action taken by the agent according to its policy. The agent's goal is to maximize the rewards. The value function may be described as the long-term

goal of the agent. It is the expected reward an agent may achieve in the future. This is in contrast to the reward signal, which only gives the immediate motivation for an action. As a fourth element we can add the model, which is used for planning. It describes the environment and the change of state. This is entirely optional, but allows the use a different set of algorithms.

1.3 Deep Learning

Deep learning is an established approach to learn data representations with the help of deep neural networks. In contrast to "shallow" neural networks deep ones consist of many layers between the input and the output layers, called hidden layers. Deep learning is dealing with learning multiple levels of representation and abstraction that can be discovered in structured data as well as in unstructured data [10]. Advancements in hardware and learning algorithms have made it feasible to employ deep learning architectures in a multitude of machine learning tasks, for example computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation or other classification tasks. Modern architectures achieve a superhuman performance in many of those.

1.4 Deep Reinforcement Learning

The idea of using deep neural networks with reinforcement learning is not new, but advancements in computing power allow a more liberal use of these techniques. We may employ deep neural networks to approximate each part of a reinforcement learning Problem (policy, reward function, value function) as needed. With naive approaches this is highly unstable but recent approaches (e.g. [9], [4]) have made many improvements and achieved a more stable learning without exploding or vanishing gradients. Systems using deep reinforcement learning show promising ability to generalize learned behavior and once trained can be used in similar models and achieve good results.

2 Macroscopic Usage of Machine Learning: Simulation Optimization

The term simulation optimization refers to the search for specific hyperparameter settings to a stochastic simulation such that a target objective, which is a func-

tion of the simulation output, is minimized without loss of generality [11]. As complex models that are build nowadays might include an extremely large number of possible hyperparameter settings, machine learning methods can be very useful to find an adequately good hyperparameter setting for the model of interest.

3 Usage of Machine Learning within a Model

3.1 Agent-Based Models

Agents are defined as autonomous entities that interact with themselves and the environment. This interaction may be defined through simple rules and may lead to interesting emergent behavior. The rules often have a physical, biological or sociological background.

A well-known example model are the Lotka-Volterra equations. They describe the population of interacting species in a dynamic system. Let x be the number of prey and y the number of predators, then we may describe the change of population as follows:

$$\frac{dx}{dt} = \alpha x - \beta xy \quad (1)$$

$$\frac{dy}{dt} = \delta xy - \gamma y \quad (2)$$

α , δ are reproduction parameters of the prey and predators, while β is the contact rate (can be interpreted as prey consumption rate) and γ is the loss of population of the predator species.

If we want to describe this as an agent-based model, we first define two different classes of agents with differing behavior. While both agents move around randomly, the predator agent eats a prey if it comes within a certain radius $\hat{\beta}$ and the predator dies with a certain probability $\hat{\gamma}$. Within fluctuations due to stochastic effects, this model behaves almost equally to the classical model.

From this point of view it is easy to track and analyze the behavior of a single agent. It is also easy to add certain constraints to the environment where the agents live and so generate new insights.

3.2 Prediction

Many models include some predictive element and may benefit from machine learning for doing so. If we consider the model of an icehockey game, the agent im-

proves his performance if he skates where the puck is going to be, because he is able to accurately predict the trajectory of the puck, based on data from real pucks.

3.3 Reinforcement Learning

As reinforcement learning deals with agents learning a certain behavior, it fits naturally in improving agent-based models [7].

Behavioral Approximation. Many behavioral rules are based on data, collected in real world environments. These rules can be very complex and often cannot be sufficiently simplified to be summarized in a rule system. Deep reinforcement learning can be used to approximate the behavior of the agents from data without any prior knowledge [2] [3].

Behavioral Optimization. Another aspect of employing deep reinforcement learning to agent based modeling is training the agents to utilize an optimal behavior. If we consider the predator-prey model, a single prey agent may survive longer if it evades the predator. If we let a prey learn, such a behavior this can lead to swarm or herd building as emergent behavior [6] [5].

4 Model Approximation

Machine learning can also be very helpful to modelers for approximating already existing models. The universal approximation theorem [12] states that a feed-forward neural network with a single hidden layer containing a finite number of neurons can approximate any continuous function with arbitrary accuracy on compact subsets of \mathbb{R}^n under very mild assumptions on the activation functions used. One of the first versions of the theorem was proven by George Cybenko for the case of sigmoid activation function [13]. In an extension by Kurt Hornik this was shown for a larger set of activation functions [14].

Universal Approximation Theorem. Let ϕ be a nonconstant, bounden and monotonically increasing continuous function, I_m the m -dimensional unit hypercube and $C(I_m)$ the space of continuous functions on I_m . For any $\varepsilon > 0$ and any function $f \in C(I_m)$ there are an integer N , real constants $v_i, b_i \in \mathbb{R}$ and real vectors

$w_i \in \mathbb{R}^n$ with $i = 1, \dots, N$ such that:

$$\left| \sum_{i=1}^N v_i \phi(w_i^T x + b_i) - f(x) \right| \quad (3)$$

This still holds when replacing I_m with any compact subset of \mathbb{R}^m .

How this approach can be implemented to enlarge an already existing model is shown by the following example.

4.1 City Quarters with Optimised Solar Hybrid Heating and Cooling Systems

In order to be able to share potential energy surpluses between buildings, a good starting point for a city quarter model like in this project (FFG number 845168) was to build physical models for certain building types with specified configurations (e.g. regarding size, the domestic heat water demand, a potential photovoltaic unit, a potential cooling system). Simulation runs for one building are computationally feasible, but the computation times for combining a lot of buildings to a city quarter model or maybe even a model of a whole large city really skyrocket. Thus the approach was to approximate the physical building models with computationally cheaper data models based on machine learning algorithms like extremely randomized trees [15], a special kind of random forests.

Machine learning based model approximation allowed to transfer the original model to larger scales.

References

- [1] Mitchell, Tom M., *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA. 1997.
- [2] Wang et al, *Deep Reinforcement Learning of Cell Movement in the Early Stage of C. elegans Embryogenesis*. arXiv:1801.04600
- [3] Bloembergen et al, *Evolutionary Dynamics of Multi-agent Learning: A Survey*. *J. Artif. Int. Res.* May 2015; 53: 659–697.
- [4] Mnih et al, *Playing Atari with Deep Reinforcement Learning*. arXiv: 1312.5602
- [5] Yang et al, *An Empirical Study of AI Population Dynamics with Million-agent Reinforcement Learning*. arXiv: 1709.04511
- [6] Olsen, Megan M. and Fraczkowski, Rachel, *Co-evolution in predator prey through reinforcement learning*. *Journal of Computational Science*; 2015
- [7] Stankovic, M., *Multi-agent reinforcement learning*. Symposium on Neural Networks and Applications. 2016.
- [8] Sutton, RS, Barto AG *Reinforcement Learning: An Introduction*. 4th Edition. Publisher: MIT Press; 2018
- [9] Mnih et al, *Human-level control through deep reinforcement learning*. *Nature* 2015. 518: 529–533. doi: 10.1038/nature14236
- [10] Tolks, Andreas, *The Next Generation of Modeling & Simulation: Integrating Big Data and Deep Learning*. Summer Simulation Multiconference 2015, Chicago, IL.
- [11] Amaran, Satyajith et al., *Simulation Optimization: A Review of Algorithms and Applications*. *Annals of Operations Research*, May 2016, Volume 240, Issue 1, pp 351-380.
- [12] Balázs Csanád Csáji, *Approximation with Artificial Neural Networks*. Faculty of Sciences. Eötvös Loránd University, Hungary. 2001.
- [13] Cybenko, G., *Approximations by Superpositions of Sigmoidal Functions*. *Mathematics of Control, Signals, and Systems*, 2 (4), 303-314, 1989.
- [14] Hornik, Kurt, *Approximation Capabilities of Multilayer Feedforward Networks*, *o. Neural Networks*, 4(2), 251–257. doi:10.1016/0893-6080(91)90009-T, 1991.
- [15] Geurts, P. et al. *Extremely Randomized Trees*. *Machine Learning*, 63(1), 3-42, 2006.