

Surrogate Modelling in (and of) Agent-Based Models: A Prospectus

Sander van der Hoog¹ 

Accepted: 8 February 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract A very timely issue for economic agent-based models (ABMs) is their empirical estimation. This paper describes a line of research that could resolve the issue by using machine learning techniques, using multi-layer artificial neural networks (ANNs), or so called Deep Nets. The seminal contribution by Hinton et al. (Neural Comput 18(7):1527–1554, 2006) introduced a fast and efficient training algorithm called Deep Learning, and there have been major breakthroughs in machine learning ever since. Economics has not yet benefited from these developments, and therefore we believe that now is the right time to apply multi-layered ANNs and Deep Learning to ABMs in economics.

Keywords Surrogate modelling · Agent-based models · Estimation

JEL Classification C63 · E03 · E27

1 Introduction

Agent-Based Models (ABMs) are becoming a powerful new paradigm for describing complex socio-economic systems. A very timely issue for such models is their empirical estimation. The research programme described in this paper will use machine learning techniques to approach the problem, using multi-layer artificial neural net-

Financial support from the Horizon 2020 ISIGrowth Project (Innovation-fuelled, Sustainable, Inclusive Growth), under Grant No. 649186, is gratefully acknowledged.

✉ Sander van der Hoog
svdhoog@gmail.com

¹ Economic Theory and Computational Economics (ETACE), Department of Business Administration and Economics, Bielefeld University, Bielefeld, Germany

works (ANNs), such as Deep Belief Networks and Restricted Boltzmann Machines. The seminal contribution by Hinton et al. (2006) introduced a fast and efficient training algorithm called Deep Learning, and there have been major breakthroughs in machine learning ever since. Economics has not yet benefited from these developments, and therefore we believe that now is the right time to apply Deep Learning and multi-layer neural nets to agent-based models in economics.

The agenda of this paper is to briefly sketch the current state-of-the-art in Artificial Intelligence (AI) and Machine Learning (ML), and apply them to economic decision-making problems. We outline a research programme that encompasses co-evolutionary learning, learning from experience, and artificial neural networks (ANN), and connect these to agent-based modelling in economics and policy-making.

We begin by tracing back the common heritage of complexity economics and evolutionary economics. A rich body of work by evolutionary economists deals with decision-making by real economic agents, in an attempt to capture their routines by automatizing their decision-making processes in computer algorithms. This starts with Herbert Simon's "A Behavioral Model of Rational Choice" (Simon 1955), it continues with Cyert and March's "A Behavioral Theory of the Firm" (Cyert and March 1963), and culminates in Simon's seminal work on "The Sciences of the Artificial" (Simon 1996 [1969]).

Around the same time, a computer science conference on "The Mechanization of Thought Processes" was held at the National Physical Laboratory (NPL) in 1958 (National Physical Laboratory 1959). The conference proceedings contain many of today's hot topics in Machine Learning: automatic pattern recognition, automatic speech recognition, and automatic language translation. At the time, the developments in AI, machine learning and theories of human decision-making were strongly intertwined.

After describing this common heritage, we take stock of the current state-of-the-art in Machine Learning and extrapolate into the not too distant future.

As a first topic, we propose to emulate artificial agent behaviour by so called surrogate modelling inside the model (this is the "in" part of the title). Such an emulator is an approximation or an imitation of the original system, that behaves in the same way but is computationally more tractable. In an ABM this could be thought of as a doppelgänger approach, in which one agent is observing another agent's behavioural pattern and their performance. It then tries to mimic that agent's behavioral pattern, and eventually replaces it in the simulation. In addition, the concept of an ANN-Policy-Agent is introduced who learns from observations of successful policy actions through a reinforcement learning mechanism. Reinforcement learning (RL) is a machine learning method in which an agent is situated in an environment and must select actions that result in outcomes, for which the agent receives a reward (or a penalty). The objective is to maximize the expected life-time rewards. Over time, the agent should learn what is the best course of action over a set of observed states of the environment, and this learned strategy is called the agent's policy. Since there may be some delay between taking an action and observing the outcome and receiving the reward, it is common in RL to discuss the credit assignment problem, which is the problem of correctly associating the current reward to a particular action. For economic policies, such as monetary policy, this can be of considerable importance since a considerable amount

of time could pass by between taking an action (e.g., increasing the policy interest rate) and the outcome, with many other economic processes happening at the same time.

A second topic of the paper is that we propose to use ANNs as computational emulators of entire ABMs (this is the “...(and of)” part in the title). In this context, the idea of an emulator or surrogate model is that the ANN functions as a computational approximation of the non-linear, multivariate time series generated by the ABM. There are various advantages to having such an emulator. It allows for a computationally tractable solution to the issue of parameter sensitivity analysis, robustness analysis, and could also be used for empirical validation and estimation. This is particularly appealing for large-scale ABMs that are computationally costly to simulate.¹

The goal is to develop new computational methods to improve the applicability of macroeconomic ABMs to economic policy analysis. When successful, we would have drastically reduced the complexity and computational load of simulating ABMs, and come up with new methods to model economic agents behaviour. Linking the time series forecasting capabilities of ANNs to ABMs also allows us to envision the possibility of docking experiments between different ABMs: the time series output from one ABM can be fed into a deep-layered ANN as input. This ANN can then be used as an agent inside another, larger-scale ABM. This notion leads to a hierarchical modelling scheme, in which ABMs of ABMs would become feasible.

Each agent in the larger ABM can have an internal mental model? of the world it inhabits, and those mental models may differ to any degree due to different experiences. On the longer term, this approach would allow the inclusion of computational cognitive models into economic ABMs, allowing the agents to be fully aware of their environment, and to consider the social embedding of their interactions.

1.1 Related Literature

There are currently several research efforts under way to construct agent-based macroeconomic models (Dawid et al. 2018a; Dosi et al. 2017; Grazzini and Gatti 2013). These models aim to compete with standard Dynamic Stochastic General Equilibrium (DSGE) models that are currently in use by most Central Banks around the world. Using DSGE models for policy analysis requires that they are calibrated and estimated on empirical data. For the ABMs this is no different, but we need new methods and techniques to estimate such policy-oriented ABMs.

The empirical estimation and validation of this type of model is an active field of research (see e.g., Werker and Brenner 2004; Brenner and Werker 2006; Fagiolo et al. 2007; Grazzini et al. 2012, 2013; Grazzini and Richiardi 2013; Yildizoglu and Salle 2012; Barde 2016, 2017; Lamperti 2015; Lamperti et al. 2017; Lux and Zwinkels

¹ Even though we have to generate quite a lot of training data from the computationally heavy simulation model, this is necessary only once to train the surrogate model. Once the surrogate is in place, it can be used to replace the heavy model, precisely as the term “surrogate” would seem to imply. Note that the computational complexity of training the surrogate model on the training data might turn out to be quite high. It therefore seems reasonable to assume that this method is only applicable to a subclass of ABMs that benefit from using such emulators. I thank a referee for this observation.

2018; Barde and van der Hoog 2017). However, until now no clear consensus has emerged on how to resolve the empirical validation problem.

In order for agent-based models to be useful for economic policy analysis so called “what-if scenarios” are used to test counter-factuals in would-be worlds. It is therefore necessary to use models with a sufficiently high resolution in terms of behavioural patterns and institutional details. For the large-scale agent-based models that we consider in this paper, ‘large-scale’ means on the order of thousands of agents. High-resolution may refer to the resolution of time-scales, geographical scales, decision-making scales (number of options to consider), or other dimensionality of the agents’ characteristics.

The advantage of such large-scale, high-resolution, high-fidelity agent-based models is that they can be used as virtual laboratories, or as laboratory “in silico” (Tesfatsion and Judd 2006). The model can be used for testing various economic policy scenarios (Dawid and Fagiolo 2008; Dawid and Neugart 2011; Fagiolo and Roventini 2012b, a), that may not be feasible to test in the real world (e.g., due to ethical objections). Examples of such scenarios include: What happens when the biggest banks go bankrupt? Or: What happens when a Euro member leaves the Euro? Obviously, these are not things we want to simply test in the real world, considering the detrimental social consequences and ethical objections.

In many cases, we do not know the correct equations of the economic model, and we might only know how to model the behaviour of the artificial economic agents approximately by observing the behaviour of their empirical counterparts (e.g., through direct observation of market participants, or through laboratory experiments). Therefore, we do not have access to the mathematical description of the economic system, and have to resort to computational modelling.

The disadvantage of such large-scale ABMs is that they are quite heavy computationally. It is easy to generate overwhelming amounts of data, and reach the boundaries of what is commonly accepted to be computationally tractable, in terms of simulation time, number of processors used, and data storage requirements. If we then want to apply such models to perform policy analyses, we have to test the robustness of the model, i.e., test whether the empirical stylized facts are still reproduced for many parameter settings. This involves performing a global parameter sensitivity analysis and a robustness analysis against small changes in the economic mechanisms, or with respect to changes in the individual behavioural repertoires of the agents. This usually requires a large number of simulations (on the order of thousands), in order to obtain a large enough sampling of the phase space, and to be able to ascertain whether the model is sensitive, stable, robust, or fragile.

In combination with the computational burden of running these simulation models, the current generation of large-scale agent-based simulation models require large computing systems, such as multi-processor servers, HPCs, or grids of GPUs, in order to run sufficiently many simulations. This not only involves running large numbers of simulations for producing results for publications, but also to perform rigorous robustness testing, parameter sensitivity analyses, and general verification and validation (V&V) procedures to ensure the correctness and validity of the computer simulations (cf. Sargent 2011; Kleijnen 1995, 2015).

The issue of computational intractability is therefore looming and ubiquitous. But this problem has been around for a long time in physics and climate science, where

research using many-particle systems and large-scale climate models is constantly pushing the frontier of what is feasible from a computational point of view.

For economics, the scientific relevance and innovativeness of this line of research is that it tries to solve the generic problem of computational tractability of computer simulation models (and more specifically, of agent-based economic models) from an algorithmic point of view, and not by resorting to technological solutions (e.g., parallel computing, or GPU grids). The innovation is to use machine learning algorithms in order to reduce the computer simulation to a lighter form, by emulating the models. One such method is to use artificial neural networks as the surrogate modelling approach, but other methods exist as well. Note however that in order to use the machine learning algorithms a large amount of training data has to be generated.

2 Applying Machine Learning Methods to Economic Problems

A primary motivation for applying machine learning techniques to economic decision making problems is the work by Herbert Simon on bounded rationality and satisficing in “*Administrative Behavior*” and “*Sciences of the Artificial*” (Simon 1997 [1947], 1955, 1959, 1996 [1969]). Simon was also one of the founders of modern-day Artificial Intelligence (AI), so it seems only appropriate that in applying artificial neural networks to economic problems, we rely on various aspects of his path-breaking work.

The first aspect we adopt is goal-oriented, adaptive behaviour. In a perfect world agents are not required to spend time on planning and learning. They already have all the relevant information available, and are able to compute with full accuracy the outcome of their actions. However, a substantial amount of time is spent by real decision makers on planning and learning about new information. Time constraints are important for making decisions, hence satisficing with threshold aspiration levels rather than optimizing would be the preferred methodology.

Open, complex systems make it essential to behave in a flexible, adaptive manner, rather than using rigid, predetermined rules that prescribe an exact course of action for every contingency. This naturally leads to the use of heuristics, routines, and rules of thumb. Satisfying aspiration levels instead of optimizing appears to be more appropriate as a model of human decision makers, as in the adage ‘It’s better to be approximately right than exactly wrong.’ Such an approach would lead to decision makers who realize they are fallible, and in order to achieve their goals they must do the best they can given the circumstances. They would aim for robust decision making routines, rather than precise prescriptions.

Such considerations point into the direction that human decision makers are not always able to make perfect decisions, due to various limitations in their decision making capabilities: (1) Imperfect information gathering, or incomplete observation of outcomes. (2) Limitations in storage capacity or faulty interpretation of those observations (imperfect recall, bad documentation of results). (3) Limits in processing abilities. (4) Imperfections in foreseeing the exact consequences of their actions. Even when acting in perfect isolation or when they act in the belief that they have precise control over their actions, unintended consequences of deliberate, decisive human action may result from a noisy environment. All such imperfections in gathering, storing and

processing of information and in foreseeing events are a fact of life for the human decision maker.

A second motivation for applying machine learning techniques to economic problems is the seminal book “A Behavioral Theory of the Firm” by Cyert and March (1963). This book describes many operating procedures related to real firm decision making. Besides an emphasis on organizational processes and decision making routines, a further aim of the theory was to link empirical data to the models by considering the results of case studies of real firms.

A clear assessment of the impact of *A Behavioral Theory of the Firm* and its methodological stance was given by Argote and Greve (2007, p. 339):

The general methodological point was that theory should model organizational processes, and should be generated through systematic observation of processes in actual organizations. One component of this point is that organizational theory should not oversimplify. Although parsimony is needed in theory building, parsimony that throws out basic insights – like replacing a process model with a maximization assumption – can be harmful.

In the context of agent-based economic models, this idea has been developed further into the Management Science Approach (Dawid and Harting 2012). In this approach the economic agents are assumed to use decision making routines that are empirically-grounded in the Management Science literature. The underlying assumption is that managers of a firm apply the methods and techniques that they have been taught whilst doing their MBA at Management School. This method can for example be applied to model the pricing behaviour of firms, the inventory management problem, the interest rate setting for loans by bank managers, or the hiring and firing practices of a Human Resource Management department.

In our approach, we use a combination of the artificial intelligence methods proposed by Simon (learning appropriate heuristics in order to satisfy certain goals), and the empirically-grounded behavioural rules as proposed by Cyert and March (actual organizational processes).

Another exciting field of research is to include a rich cognitive structure into the agents’ behavioral repertoires. The decision making routines, although adaptive, are often still too rigid from a cognitive science point of view. A lack of meta-rules to update the behavioral rules is often seen as a serious drawback of these models, especially when it comes to addressing the Lucas Critique which states that economic policy has to take into account the adaptive behavioral response by the agents that are subject to the policy.

This perceived lack of cognitive modelling in the behavioral routines of economic agents can be alleviated if we would allow each agent in the ABM to have an internal “mental model” of the world it inhabits, and those mental models can differ to any degree. On the longer term, this approach would allow the inclusion of computational cognitive models into economic agent-based models, allowing the agents to be fully

aware of their environment, and possibly also to consider the social embedding of their interactions.²

2.1 Machine Learning for Time Series Forecasting in Economics

Applications of ANNs to time series forecasting problems in economics include: financial market forecasting (Trippi and Turban 1993; Azoff 1994; Refenes 1995; Gately 1996), foreign exchange rates (Weigend et al. 1992; Refenes 1993; Kuan and Liu 1995), load demand forecasts on electricity markets (Bacha and Meyer 1992; Srinivasan et al. 1994), commodity prices (Kohzadi et al. 1996), and macroeconomic indices (Maasoumi et al. 1994). A review of applications of ANNs in the field of Management Science and Operations Research is given by Wilson and Sharda (1992) and Sharda (1994). The M-competition (Makridakis et al. 1982) provides a widely cited data base for comparing the forecasting performance of ANNs in comparison to traditional statistical methods. The data for the M-competition are mostly from business, economics, and finance, see Kang (1991), Sharda (1994), Tang and Fishwick (1993) for examples. Another comparison is provided by the Santa Fe forecasting competition (Weigend and Gershenfeld 1993) which includes very long time series coming from various fields.

2.2 The Usefulness of ANNs to Study Complexity Science

According to Gorr (1994), ANNs are very appropriate in the following situations: (1) large data sets; (2) problems with nonlinear structure; (3) multivariate time series forecasting problems. Below we discuss several important research questions.

A first issue is how ANNs model the autocorrelated time series data and produce better results than traditional linear and non-linear statistical methods. According to Bengio et al. (2013), sequential statistical data (time series) suffer from the “difficulty of learning long-term dependencies”. If the past is coded linearly (regardless of how many observations in the past) then the effect of the past of the *previous* step is diminishing exponentially. If the past is modelled non-linearly, then the non-linearity is “composed many times”, leading to a highly non-linear relationship of past to present. According to the paper cited, recurrent neural networks (RNN) are better in modelling such relationships. However, RNNs suffer from the problem of “diminishing gradients” when using back-propagation for training the weights with Stochastic Gradient Descent. A solution to this problem is to use information about the second-order derivatives. In such cases Momentum Methods such as Nesterov’s Accelerated Gradient (NAG) seem more promising (see Sect. 3 for more details).

All of the methods proposed in the literature for modelling time series data with ANNs are gradient-tracking in one way or the other, and are conservative about sudden changes. Hessian-free optimization (Martens and Sutskever 2011) and the Ph.D.-thesis

² A possible candidate for such environmentally-aware, cognitive algorithms would be Hutter’s AIXI, see Hutter (2000).

by Sutskever (2013) show the applicability of such methods in the multivariate time series domain.

A second issue is how to systematically build an appropriate network given a specific forecasting problem. To follow the current best-practices as outlined above we can start from the simplest RNN representation, and try state-of-the-art approaches. The design of good initializations of the networks is a good starting point. If we have domain knowledge about the units that operate in the system and their qualities, we can estimate the relative size of each input node and the long term effect that actions should have. We then use state-of-the-art parameter estimation techniques, as described in Bengio et al. (2013) in order to fix the weights on the input nodes.

A third issue is the question what is the best training method for time series forecasting problems. This is discussed extensively in Sutskever (2013), noting that optimized Stochastic Gradient Descent (SGD) may be adequate, if one considers proper initialization of the network. Momentum methods are another option. As described above, we could start with the simplest method first (SGD), or consider the best practice for problems similar to ours. We should keep in mind, however, that we may have various problems that are different in structure.

A fourth and final question is how we should go about designing the sampling scheme, and the pre- and post-processing of the data? What are the effects of these factors on the predictive performance of ANNs?

One of the advantages of ANNs is that they alleviate the need for *feature engineering* which is the art and science of traditional machine learning. Instead, any real number goes through a squashing function (logistic or tanh), resulting in a number between 0 and 1 (or -1 and 1). In case of categorical values, one can have a ‘softmax layer’, that assigns a probability distribution over the states. Alternatively, one can have “ON/OFF” nodes with binary values. The fine art then becomes how to design the structure of the network itself: how many layers, and how many nodes per layer.

3 Applying Surrogate Modelling to Economic ABMs

Due to the fact that macroeconomic agent-based models are computationally heavy simulators, if we want to apply them to policy analysis, we need to reduce the complexity of the ABM simulation to a less complex, more computationally tractable system. This means that we need to develop surrogate models that allow us to approximate or emulate the multi-dimensional nonlinear dynamics of the original system. The entire process of finding such an approximate emulator for an ABM consists of four steps.

First, we construct an ABM and generate synthetic time series data. Then, a multi-layered, deep neural network is designed and trained on the synthetic data. Third, the trained neural network should be empirically validated using real-world data. And fourth, we apply the trained, empirically validated deep neural network to economic policy analysis.

According to these four steps, the question how to construct efficient emulators of ABMs could be structured along four broad research questions, or research themes:

1. Micro-emulation of the behaviour of individual agents inside the model. This involves creating a so called “doppelgänger” of each agent, that mimics the agent’s behavior.
2. Macro-emulation of an entire ABM simulation, using the multivariate time series data that is produced by the model.
3. Reduction of the complexity to design ANNs, setting the number of input nodes, number of hidden layers, and number of nodes in each hidden layer.
4. Reinforcement learning in economic policy design, generating an ANN-policy agent that can be used for policy analysis.

In the end, the emulation of agent-based models using artificial neural networks (ANN-emulation) allows us to perform specific types of analysis that appear quite complicated with large-scale ABMs.³ For example, the empirical validation of the ANN-emulator could be an efficient, indirect method to calibrate the parameters of the ABM itself. Also, a global parameter sensitivity analysis could be performed using the ANN-emulator instead of the computationally heavy ABM. And finally, after a careful internal validation procedure to check that the ANN can in fact emulate the ABM to a sufficient degree of accuracy, the ANN-emulator could also be used as a policy analysis tool.

In the following sections we describe each of these research questions in more detail.

3.1 Research Theme 1: Micro-emulation of the Behaviour of Economic Agents

This is a local approach, in the sense of modelling the behaviour of each of the rule-based agents by ANNs. A neural network is trained to predict the actions of a particular agent in the model, i.e. the ANN acts as a doppelgänger of that agent. Due to the multitude of instances and agent types, each with their own set of instructions and constraints, and because of the dynamically changing environment of the ABM, such networks can help us model the behaviour of our agents and reduce the complexity of the model at the local level.

The ANNs may need extensive training, but are cheap when they run.⁴ In the end, the original agents may be replaced by their emulator code, or we may run the hybrid model with both types of agents.

Various decision making problems in standard macroeconomics models are formulated as optimization problems. This theme is dedicated to show that this could be dealt with equally well using machine learning methods. In each example below, we replace a standard optimization problem with a heuristic.

The examples all refer to the Eurace@Unibi model (Dawid et al. 2018a, b).

³ But not impossible in principle. For example, the global sensitivity analysis of a large-scale ABM such as the Eurace@Unibi Model was already performed using HPC clusters (Barde and van der Hoog 2017). Also, empirical validation is currently being done for medium-sized ABMs, and given the exponential increase in computing power is expected to yield results in the coming years.

⁴ A side-remark must be made here on the computational complexity of the neural network training. Since it is not uncommon that ANNs have millions of weights that must be trained, this is by no means a simple feat.

1. The firm's consumer demand estimation problem In the current model, demand is estimated by two methods, one is backward-looking, the other is forward-looking. In the backward-looking method, the firm only relies on past observations and uses a simple OLS regression on the previous months' sales revenues to estimate the future demand for its product. It estimates the parameters of a linear, quadratic, or cubic fit for the perceived demand function that the firm believes to be facing (for more details, see Harting 2014, Ch. 1, pp. 13–14).

In the forward-looking method, the firm uses a market research routine that is commonly used in practice, namely by doing simulated purchase surveys among its consumers (Nagle and Hogan 2006). Once a year, the firm considers a sample of households and presents these with a set of products at differing prices. The survey contains questions regarding consumers' preferences and price sensitivities, and asks them how they would react and how they would alter their consumption pattern when the firm would change its price (assuming the prices of all its competitors stay the same). In this way, the firm tries to gauge the overall price sensitivity of its consumers, and to estimate its future market share. (see Harting 2014, Ch. 3, pp. 79–81 for more details on the market research method, and references therein).

The firm's consumer demand estimation problem using artificial neural networks The idea of replacing the linear, quadratic, or cubic fitting of the data by a neural network is relatively straightforward. The ANN-firm would try to estimate the local slope of its demand function by using an ANN, and adjust its arc weights while the simulation is ongoing. Since neural networks are a data-driven approach, there is no need to assume any particular statistical model. Also it is not necessary to rely on linear statistical methods such as OLS, since ANNs are non-linear, non-parametric time series methods.

2. The consumers' demand problem In the current model, the consumers' decision to buy a product from a specific firm is derived from a discrete choice model using a multinomial logit function. The selection probability to select a firm is an increasing function of the consumer's utility for that firm's product. The utility value is decreasing in price, so a firm with a higher price will have a lower selection probability, but it is bounded away from zero. By replacing the logit function with a neural network formulation of the same problem, the ANN-consumer can learn how to best achieve its target value. Of course, for the ANN to successfully learn the optimal behavior, it has to be given rewards which makes this a form of supervised learning. In this way, the consumers' choice problem is redefined as goal-oriented behaviour, rather than as a stochastic model of choice.

3. The banks' internal risk model Banks' decision making process involves the problem of setting interest rates for individual loans to the private sector (corporate loans to firms and mortgage loans to households). In order to make a profit, banks should assess their debtors' credit worthiness, and the likelihood that the borrower will not repay the loan in the near future (i.e., the probability that they will default on the credit). This includes an evaluation of the probability of default of the debtor, but also should take into account the bank's own funding costs. In order to make such an assessment,

the bank can either use an internal risk model, or rely on information provided by external rating agencies, or both. Whichever method is being used, they have to assess the various types of risk associated to their assets, including market risk, credit risk, and systemic risk (Duffie and Singleton 2003). Market risk refers to changes in the value of the assets on the balance sheet of the bank. Typically, these are fluctuating due to the mark-to-market asset valuation and the volatility of prices on the financial asset markets. Credit risk refers to the risk the bank is facing due to the changing values of assets on the balance sheets of its debtors.

In the current version of the Eurace@Unibi model, the bank uses a highly simplified backward-looking model to determine the probability of default of loans in their portfolio, by considering the aggregate default flow of the previous quarter. For new credit requests on the other hand, the bank uses a forward-looking model in which the probability of default is based on the balance sheet data of the firm. E.g. it is based on the firm's equity and financial ratios such as the debt to equity ratio which is an indicator of financial fragility. The bank's in the model believe these indicators are a good measure for the firm's future probability of default.

In summary, such a "structural model" approach may or may not be in accordance with the actual behavior of real banks, which would be a matter of empirical study that is beyond the scope of our current paper. And in fact many alternative models for evaluating credit default risk exist, as illustrated by the rich overview given by Duffie and Singleton (2003).

One such an alternative approach is the "first-passage model" (Duffie and Singleton 2003, pp. 53), which uses empirical time series collected over a certain time window, to determine the actual default probabilities for a population of firms that have similar risk profiles. Such a time series approach differs substantially from the more theoretical "reduced-form" approaches, and they would fit quite nicely with the neural network approach.

The artificial neural network approach to model the banks' decision making problem will thus provide us with a nonlinear, nonparametric, multivariate time series forecasting method. The bank can be modelled as a goal-oriented entity, that tries to set interest rates based on forecasted default probabilities, which are derived from time series that are being generated online, i.e. during an ongoing simulation. In the end, this could yield an agent-based model of the credit market in which the credit risk models proposed in Duffie and Singleton (2003) have been internalized into our agents' behavioural repertoires.

This line of research can distinguish between "offline training" and "online learning". Offline training makes use of time series data that are being generated by an agent-based model that is "detached" from the agent. One can think of this as an outside-observer-approach, where the agent is not part of the simulation environment, but can observe the actions and outcomes of other agents. This is similar to how children learn how to behave appropriately in a social environment, before they are accepted as full members of that environment.

Online learning, on the other hand, occurs while the agent is itself part of the simulation environment, and is observing the time series being generated online. If multiple agents are simultaneously using online learning in this sense, we can speak of

co-evolutionary learning by a population of heterogeneous, artificial neural network agents.

The main aim of this particular research theme is to focus on the appropriate network structure to emulate the multivariate time series data being generated by the target system (in this case, the particular agents to emulate). The final goals of this research strain are to obtain: (1) a model of firm behaviour replaced by an ANN for the firm's demand estimation routine; (2) a model of consumer behaviour replaced by an ANN for the consumers' choice problem; (3) a model of bank behaviour replaced by an ANN for the bank's internal risk evaluation and risk-assessment problem.

3.2 Research Theme 2: Macro-emulation of an Entire ABM Simulation Model

Due to the recent breakthrough of Deep Learning techniques for multi-layered networks to model non-linearities, it becomes possible to emulate an entire ABM simulation by an ANN generating time-series. Contrary to the local approach described above this is a global approach in which a neural network is trained to predict the probabilistic structure of the macro-level variables of the model.

This is useful for robustness and parameter sensitivity analysis, since it allows a much larger exploration of the parameter space. A second advantage is that by training the ANN on many counter-factual scenarios, it is expected to perform better than an ANN that has just been trained on empirical historic data since this is just one realization of the empirical data generating process.

In our aim to make the problem of tuning the parameters of an ABM more tractable, we try to emulate the input/output function of the entire ABM by an (ultimately) less complex and more tractable Deep Neural Network. Our starting point is that, in an ABM, a multitude of autonomous agents react to changes in their (economic) environment and, in doing so, alter that very environment. In a Deep Neural Net, a multitude of nodes at different layers can encode different information structures and decision processes, such that the network as a whole can serve specific functions. Bringing the two together, we aim to train a neural network that produces the same output (in terms of time series of macro-economic variables) as the ABM.

This problem is similar to multi-variate time series prediction, with the difference that instead of estimating the future values based on the past values, the actions of the agents in the model are inputs to the ANN. For such tasks a recurrent neural network (RNN) is the obvious choice. A key part of the design is a *feedback* property, namely that the output of the model (the values of the measured macro-economic variables) are fed back to the input. A second key part is to split the network input layer to represent the 'present', and the 'past'. This is how the RNN design captures 'history': at each step t the network receives inputs at time t , but *also* of time $t - 1$, and possibly further time lags to account for long time-dependencies.

In terms of integrating the decision processes of the individual agents, a first approximation could be a multi-layered structure in which the nodes of the first input layer are entire ANNs that model each individual agent in the agent-based model. Of course, this is not expected to be any more tractable than the ABM itself. However, it is expected that the ANN will be able to emulate the ABM *with a much smaller number*

of agents, as the multi-layer structure allows the ANN to model increasingly more complex functions of the modelled economy without the need to fully emulate it. Instead of representing the individual agents one by one, the ANN is a representation of the entire ABM at increasing “layers of abstraction”. Most importantly, this theme will be informed by other themes, e.g. the modelling in theme 1 for the individual agent ANNs can inform the initial design of the macro-emulation ANN.

Training data for the macro-emulation ABM will be provided by already collected (synthetic) data from ABM simulations, as well as from new simulations with agents that are themselves ANNs, rather than the current fixed behavioral routines. The big advantage of training the ANN on simulated data, in addition to the abundance of such data, is that such a network will not just *learn* to forecast a specific realization of some ABM emulation, but it will learn the more general underlying data generating mechanism that is common to all such simulations which are seen during the training phase, and, ideally, also for new previously unseen simulations. This provides for an out-of-sample validation stage by using a subset of the synthetic data that was previously unseen by the ANN, and can be used to test the performance of the macro-emulation Deep Neural Network.

The main aim of this particular research theme is to focus on the appropriate network structure to emulate the multivariate time series generated by the macro-ABM as a whole. A second aim is to investigate what is the most appropriate learning/optimization technique for this problem. The final goal is to obtain a deep layered ANN that is trained on data generated by an ABM, and that can be usefully applied for empirical validation, and for policy analysis.

3.3 Research Theme 3: Reduction of the Complexity to Design ANNs

The design and training of deep ANNs is a complex task. To guide the design of the ANN in terms of the number of nodes and hidden layers, and in order to improve the efficiency of the Deep Learning algorithm, the complexity of the ANN must be reduced.

The problem of training deep neural networks is an unconstrained global minimization problem, i.e. to find the arc weights such that the training error of the ANN is minimized (the training error is the difference between the ANNs performance on the training set and on the test set).

This problem is NP-hard, so the computational costs will increase exponentially with problem size (given by the number of input nodes and the number of hidden layers). Therefore smart heuristics are needed to approximate the global minimum. Many such heuristics have been developed, but most of these assume that the objective function (the loss function or error function) is differentiable in its arguments. Hence the algorithms make use of the gradient and the Hessian of the objective function. Example methods include Gradient Descent (GD), Stochastic Gradient Descent (SGD), Momentum methods and Nesterov’s Accelerated Gradient (see Sutskever 2013 for an overview, and references therein).

For convex objective functions, to find the global minimum the gradient methods are globally converging, i.e. they will always find the global minimum, but they will

just take longer to converge for worse initializations of the parameters. However, for deep and recurrent networks, the initialization does matter since the objective function of such networks cannot be assumed to be convex. Hence, it is important to design good initializations for the algorithms.

The greedy unsupervised pre-training algorithm of Hinton et al. (2006) and Hinton and Salakhutdinov (2006) is a good starting point since it greedily trains the parameters of each layer sequentially. Such greedy layerwise pre-training is then followed by a “fine-tuning” algorithm such as the standard Stochastic Gradient Descent method.

Another method is the Hessian-Free (HF) Optimization (Martens and Sutskever 2010, 2012) that is able to train very deep feed-forward networks even without such a pre-training step. HF is a second-order method and therefore rather slow, but it is very powerful. It is the preferred method of optimization if there is no idea about good initializations of the network.

The most recent innovations in this field, described by Sutskever (2013, Ch. 7), are able to train very deep neural networks (up to 17 hidden layers) by using aggressive Momentum Methods. Such methods use gradient information to update parameters in a direction that is more effective than steepest descent by accumulating speed in directions that consistently reduce the objective function. The most promising method of this type is Nesterov’s Accelerated Gradient (NAG) method, which is a first-order optimization algorithm that has better convergence properties than Gradient Descent. It has two parameters: a learning rate ε and a momentum constant μ , where $(1 - \mu)$ can be thought of as the friction of the error surface. High values of μ implies the algorithm retains gradient information and leads to fast convergence, while low values imply high friction and a loss of gradient information, leading to slower convergence to the global minimum.

Using NAG with very aggressive momentum values (μ close to 0.99) leads to excellent results for problems that previously were deemed unsolvable, such as data sets exhibiting very long time-dependencies (50–200 time steps).

The main aim of this research theme is to focus on Hessian-Free Optimization and Momentum Methods, and possibly adapt these methods to our specific problems. A second aim is to optimize the choice of network parameters: the number of input nodes, the number of hidden layers, and how many nodes in each hidden layer.

The final goal of this strain of research is to design good initializations of the network parameters for the Deep Learning algorithms, and to develop insights to inform the optimization methods and the Deep Learning algorithms.

3.4 Research Theme 4: Reinforcement Learning in Economic Policy Design

The final theme is to apply a surrogate, or meta-modelling approach to policy decision-making. A government or central bank agent may be given certain goals (e.g., maintaining a stable price level, or a low unemployment rate, or macrofinancial stability) rather than using hand-crafted rules to serve those goals (such as a Taylor rule for monetary policy). Using reinforcement learning techniques an agent starts with little knowledge of the world, but given a reward function the agent learns to perform better over time, during a simulation run. The ABM allows us to evolve successful policies not only by using empirical data, but also by learning from online-generated

streaming data. The idea is to have a neural network policy agent (ANN-policy-agent), and this is again a local approach.

The objective is to develop a model with an endogenous policy-maker, an ANN-policy-agent, who evolves its decision-making routines endogenously. This agent should adapt its policy in response to the behavioural changes of the other agents in the model. Similar to research theme 1, we can again distinguish between “offline training” and “online learning”.

In offline training we train the ANN-policy-agent using pre-generated, historical data from the original ABM simulation. Its input are (rule-based) policy decisions made during that simulation and time series of economic variables, and we use unsupervised layer-by-layer training. Thus, the ANN-policy-agent learns (unsupervised) the outcome of policy decisions under specific circumstances, and it is possible to re-enforce this training with data from multiple ABM simulations by using a Monte Carlo approach. After this unsupervised pre-training, we perform an additional supervised training phase, in which we reward policy decisions that have desired outcomes. The trained ANN-policy-agent is then used in ABM simulations as the policy-making authority. Depending on the properties and coverage of the training data, this type of ANN-policy-agent is expected to fare well in the test simulations.

In the online learning setting, the ANN-policy-agent learns from online streaming data. It has to train its weights while taking actual policy actions during a running ABM economy. This situation is bad from an AI point of view, since training normally takes a long time and occurs in isolation of the actual environment. Therefore the analogy to our setup is a ANN-policy-agent that has not been trained before, or has inappropriate weights for the situation. It has to adapt as best it can, similar to a learning child. However, this setting seems more close to what actual real-world policy-makers are facing, especially in times of a changing economic environment. In times of crisis, policy-makers have to adjust quickly to changing circumstances, possibly making choices that appear suboptimal, but satisfying certain target levels.

The main aim of this theme is to focus on which reward functions and what network structures are most appropriate for the ANN-policy-agent. A second aim is to design the endogenous policy setting behaviour for the ANN-policy-agent: which behavioral heuristics are used, what meta-rules adapt these heuristics, and what are the parameters for the ANN. The final goal of this strain of research is to develop a model with a ANN-policy-agent that can set policy endogenously, and is able to adjust its policy response to the behavioral changes of the other agents in the model economy.

4 Conclusion

The purpose of this paper is to sketch a line of research in which artificial neural networks (ANNs) are used as computational approximations or as emulators of the nonlinear, multivariate time series dynamics of a pre-existing agent-based model (ABM). In other words, it is a meta-modelling approach using machine learning techniques. There are various advantages to having such an emulator. For instance, it allows for a computationally tractable solution to the issue of parameter sensitivity analysis, robustness analysis, and could also be used for empirical validation and estimation.

The overall goal is to develop new methods and techniques to improve the applicability of macroeconomic ABMs to economic policy analysis. For the practical implementation of this goal, we need to make advances in two domains. First, to develop new machine learning techniques to represent ABMs by ANNs (research themes 1 and 2 in the main text). Second, to develop new complexity reduction techniques to guide the design of such emulator ANNs (research theme 3). The work to be done consists of the following broad strains of research:

1. Micro-emulation of the behaviour of agents. A neural network is trained to predict the actions of a particular agent in the model, i.e. the ANN acts as a doppelgänger of that agent.
2. Macro-emulation of an entire ABM simulation. This is a global approach. A neural network is trained to predict the probabilistic structure on the macro-level, of variables in the ABM model, based on the initialization parameters. ANNs have proved to be very successful for multivariate time series forecasting. They are much more flexible than traditional statistical methods since they are nonlinear, nonparametric time series approximation techniques.
3. Reduction of complexity. The design of the structure of the neural network in terms of numbers of input- and output nodes and the number of hidden layers is a complicated problem. In order to improve the efficiency of the Deep Learning algorithm, the complexity of the ANN must therefore be reduced. This can be done by modelling it in terms of a Hamiltonian system, and proceed with describing the time-evolution of the Hamiltonian.
4. Reinforcement learning in policy design. A government or central bank agent may be given certain goals (such as a stable price level, low unemployment rates, or macrofinancial stability), rather than hand-crafted rules. Using reinforcement learning techniques, an agent starts with little knowledge of the world, but given a reward function that models those goals, the agent learns to perform better over time.

This may lead to more flexible policies and more adaptive behaviour on the part of the policy agent, as it allows for a more flexible, discretionary policy setting behavior, rather than using a fixed, rule-based policy. As the policy agent learns how to set policies optimally, it must adapt to the behavioural changes of the other agents, who might change their behaviour in response to the policy. Hence, this policy-feedback-loop addresses in a very natural way the Lucas Critique.

In summary, points 1–4 not only help us to design a strategy how to emulate and estimate agent-based models using artificial neural networks, but it may also contribute to the burgeoning literature on learning in macroeconomics and optimal policy design. Hence, the research programme connects both micro- and macroeconomics, and joins both estimation and emulation in machine learning.

4.1 Vision and Outlook for the Future

When successful, we could apply the new methods to a plethora of problems. We would have drastically reduced the complexity and computational load of simulating agent-based models, and come up with new methods to model economic agents'

behaviour. Furthermore, linking the time series forecasting capabilities of the Deep Learning algorithm to agent-based models also allows us to envision the possibility of docking experiments between different ABMs: the time series output from one ABM can be fed into the Deep Learning algorithm, resulting in an artificial neural network. This artificial neural network can then be used as an agent inside another, larger-scale ABM. This notion leads to a hierarchical modelling scheme, in which ABMs of ABMs would become feasible. Each agent in the larger ABM can have an internal “mental model” of the world it inhabits, and those mental models can differ to any degree. On the longer term, this approach would allow the inclusion of computational cognitive models into economic agent-based models, allowing the agents to be fully aware of their environment, and to consider the social embedding of their interactions.

Acknowledgements This paper has benefited from discussions with Spyros Kousides, Nan Su and Herbert Dawid. Any remaining errors or omissions are the sole responsibility of the author.

References

- Argote, L., & Greve, H. R. (2007). A behavioral theory of the firm—40 years and counting: Introduction and impact. *Organization Science*, 18(3), 337–349.
- Azoff, E. (1994). *Neural network time series forecasting of financial markets*. Chichester: Wiley.
- Bacha, H., & Meyer, W. (1992). A neural network architecture for load forecasting. *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2, 442–447.
- Barde, S. (2016). Direct comparison of agent-based models of herding in financial markets. *Journal of Economic Dynamics and Control*, 73, 329–353.
- Barde, S. (2017). A practical, accurate, information criterion for Nth order Markov processes. *Computational Economics*, 50(2), 281–324.
- Barde, S., & van der Hoog, S. (2017). An empirical validation protocol for large-scale agent-based models. Studies in Economics 1712, School of Economics, University of Kent. <https://ideas.repec.org/p/ukc/ukcedp/1712.html>
- Bengio, Y., Boulanger-Lewandowski, N., & Pascanu, R. (2013). Advances in optimizing recurrent networks. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 8624–8628).
- Brenner, T., & Werker, C. (2006). A practical guide to inference in simulation models. Papers on Economics and Evolution 2006-02, Philipps University Marburg, Department of Geography.
- Cyert, R. M., & March, J. G. (1963). *A behavioral theory of the firm*. Englewood Cliffs: Prentice Hall.
- Dawid, H., & Fagiolo, G. (2008). Agent-based models for economic policy design: Introduction to the special issue. *Journal of Economic Behavior & Organization*, 67(2), 351–354. (**special issue on Agent-based models for economic policy design**).
- Dawid, H. & Harting, P. (2012). Capturing firm behavior in agent-based models of industry evolution and macroeconomic dynamics. In G. Bünsdorf (Ed.), *Applied evolutionary economics, behavior and organizations* (Ch. 6, pp. 103–130). Cheltenham: Edward Elgar.
- Dawid, H., & Neugart, M. (2011). Agent-based models for economic policy design. *Eastern Economic Journal*, 37, 44–50.
- Dawid, H., Harting, P., van der Hoog, S., & Neugart, M. (2018a). Macroeconomics with heterogeneous agent models: Fostering transparency, reproducibility and replication. *Journal of Evolutionary Economics* (**forthcoming**).
- Dawid, H., Gemkow, S., Harting, P., van der Hoog, S., & Neugart, M. (2018b). Agent-based macroeconomic modeling and policy analysis: The Eurace@Unibi Model. In S.-H. Chen, M. Kaboudan, & Y.-R. Du (Eds.), *The Oxford handbook on computational economics and finance* (Ch. 17, pp. 490–519). Oxford: Oxford University Press.
- Dosi, G., Napoletano, M., Roventini, A., & Treibich, T. (2017). Micro and macro policies in Keynes+Schumpeter evolutionary models. *Journal of Evolutionary Economics*, 27, 63–90.

- Duffie, D., & Singleton, K. J. (2003). *Credit risk: Pricing, measurement, and management*. Princeton: Princeton University Press.
- Fagiolo, G., & Roventini, A. (2012a). Macroeconomic policy in DSGE and agent-based models. *Revue de l'OFCE*, 124, 67–116.
- Fagiolo, G., & Roventini, A. (2012b). On the scientific status of economic policy: A tale of alternative paradigms. *The Knowledge Engineering Review*, 27, 163–185.
- Fagiolo, G., Moneta, A., & Windrum, P. (2007). A critical guide to empirical validation of agent-based models in economics: Methodologies, procedures, and open problems. *Computational Economics*, 30(3), 195–226.
- Gately, E. (1996). *Neural networks for financial forecasting*. New York: Wiley.
- Gorr, W. L. (1994). Editorial: Research prospective on neural network forecasting. *International Journal of Forecasting*, 10(1), 1–4.
- Grazzini, J., & Gatti, D. D. (2013). Paper on the development of MABM Mark II: The input–output network in the CRISIS macro agent-based model. CRISIS Project Deliverable D3.3, Universit Cattolica del Sacro Cuore, Milano.
- Grazzini, J., & Richiardi, M. G. (2013). Consistent estimation of agent-based models by simulated minimum distance. LABORatorio R. Revelli working papers series 130, LABORatorio R. Revelli, Centre for Employment Studies.
- Grazzini, J., Richiardi, M., & Sella, L. (2012). Indirect estimation of agent-based models: An application to a simple diffusion model. LABORatorio R. Revelli Working papers series 118, LABORatorio R. Revelli, Centre for Employment Studies.
- Grazzini, J., Richiardi, M. G., & Sella, L. (2013). Analysis of agent-based models. LABORatorio R. Revelli working papers series 135, LABORatorio R. Revelli, Centre for Employment Studies.
- Harting, P. (2014). Policy design in the presence of technological change: An agent-based approach. Ph.D. Thesis, University of Bielefeld.
- Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computing*, 18(7), 1527–1554.
- Hinton, G. E., & Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507.
- Hutter, M. (2000). A theory of universal artificial intelligence based on algorithmic complexity. CoRR [arXiv:cs.AI/0004001](https://arxiv.org/abs/cs.AI/0004001).
- Kang, S. (1991). An investigation of the use of feedforward neural networks for forecasting. Ph.D. thesis, Kent State University.
- Kleijnen, J. P. (2015). *Design and analysis of simulation experiments*. Heidelberg: Springer.
- Kleijnen, J. P. C. (1995). Verification and validation of simulation models. *European Journal of Operational Research*, 82(1), 145–162.
- Kohzadi, N., Boyd, M., Kermanshahi, B., & Kaastra, I. (1996). A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing*, 10, 169–181.
- Kuan, C. M., & Liu, T. (1995). Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics*, 10(4), 347–364.
- Lamperti, F. (2015). An information theoretic criterion for empirical validation of time series models. LEM papers series 2015/02, Laboratory of Economics and Management (LEM), Sant'Anna School of Advanced Studies, Pisa, Italy.
- Lamperti, F., Roventini, A., & Sani, A. (2017). Agent-based model calibration using machine learning surrogates. Papers [arXiv:1703.10639](https://arxiv.org/abs/1703.10639). <https://ideas.repec.org/p/arx/papers/1703.10639.html>.
- Lux, T., & Zwinkels, R. C. J. (2018). Empirical validation of heterogeneous agent models. In C. H. Hommes & B. LeBaron (Eds.), *Handbook of computational economics: Heterogeneous agent models* (Vol. 4). North Holland: Elsevier.
- Maasoumi, E., Khotanzad, A., & Abaye, A. (1994). Artificial neural networks for some macroeconomic series: A first report. *Econometric Reviews*, 13, 105–122.
- Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibdon, M., & Lewandowski, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1, 111–153.
- Martens, J., & Sutskever, I. (2010). Parallelizable sampling of Markov random fields. In *Artificial intelligence and statistics* (pp. 517–524).

- Martens, J., & Sutskever, I. (2011). Learning recurrent neural networks with Hessian-Free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Bellevue, Washington, USA (pp. 1033–1040).
- Martens, J., & Sutskever, I. (2012). Training deep and recurrent networks with Hessian-Free optimization. In G. Montavon, G. B. Orr, & K. R. Müller (Eds.), *Neural networks: Tricks of the trade. Lecture notes in computer science* (Vol. 7700, pp. 479–535). Berlin: Springer.
- Nagle, T., & Hogan, J. (2006). *The strategy and tactics of pricing: A guide to growing more profitably*. Upper Saddle River, NJ: Pearson Prentice Hall.
- National Physical Laboratory (Ed.). (1959). Mechanisation of thought processes. In *Proceedings of a symposium held at the national physical laboratory on the 24th, 25th, 26th and 27th November 1958*, Her Majesty's Stationary Office, 1959.
- Refenes, A. (1993). *Constructive learning and its application to currency exchange rate forecasting*, chap 39 (pp. 777–806). Chicago: Probus Publishing Company.
- Refenes, A. (1995). *Neural networks in the capital markets*. Chichester: Wiley.
- Sargent, R. G. (2011). Verification and validation of simulation models. In *Proceedings of the winter simulation conference, WSC '11* (pp. 183–198).
- Sharda, R. (1994). Neural networks for the ms/or analyst: An application bibliography. *Interfaces*, 24, 116–130.
- Simon, H. A. (1955). A behavioral model of rational choice. *Quarterly Journal of Economics*, 69(1), 99–118.
- Simon, H. A. (1959). Theories of decision-making in economics and behavioral science. *American Economic Review*, 49, 253–283.
- Simon, H. A. (1996 [1969]). *The sciences of the artificial* (3rd ed.). Cambridge, MA: The MIT Press.
- Simon, H. A. (1997 [1947]). *Administrative behavior* (4th ed.). New York, NY: The Free Press.
- Srinivasan, D., Liew, A., & Chang, C. (1994). A neural network short-term load forecaster. *Electric Power Systems Research*, 28, 227–234.
- Sutskever, I. (2013). Training recurrent neural networks. Ph.D. thesis, Department of Computer Science, University of Toronto.
- Tang, Z., & Fishwick, P. (1993). Feedforward neural nets as models for time series forecasting. *ORSA Journal on Computing*, 5, 374–385.
- Tesfatsion, L., & Judd, K. E. (2006). *Handbook of computational economics II: Agent-based computational economics*. Amsterdam: North-Holland.
- Trippi, R., & Turban, E. (1993). *Neural networks in finance and investment: Using artificial intelligence to improve real-world performance*. Chicago: Probus.
- Weigend, A., & Gershenfeld, N. (1993). *Time series prediction: Forecasting the future and understanding the past*. Reading, MA: Addison-Wesley.
- Weigend, A., Huberman, B., & Rumelhart, D. (1992). *Predicting sunspots and exchange rates with connectionist networks* (pp. 395–432). Redwood City, CA: Addison-Wesley.
- Werker, C., & Brenner, T. (2004). Empirical calibration of simulation models. Papers on economics and evolution 2004–10, Philipps University Marburg, Department of Geography.
- Wilson, R., & Sharda, R. (1992). Neural networks. *OR/MS Today*, pp. 36–42.
- Yildizoglu, M., & Salle, I. (2012). Efficient sampling and metamodeling for computational economic models. *Cahiers du GREThA 2012–18*, Groupe de Recherche en Economie Théorique et Appliquée.