

Deep Learning in Agent-Based Models: A Prospectus

Sander van der Hoog

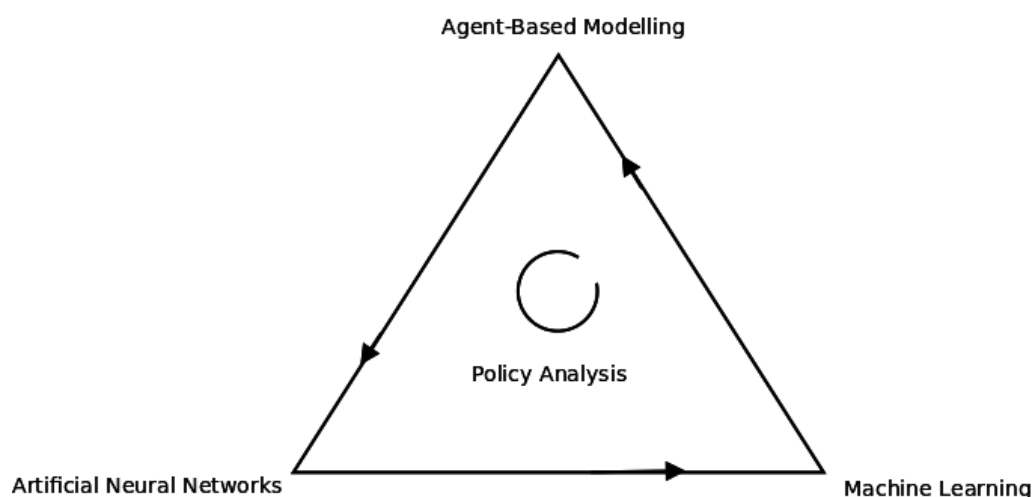
Deep Learning in Agent-Based Models: A Prospectus

Sander van der Hoog^a

January 4, 2016

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. (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-layered neural networks to agent-based models in economics.



Key words: Deep Learning, Agent-Based Models, Estimation, Meta-modelling.

^aComputational Economics Group, Dept. of Business Administration and Economics, Bielefeld University, Germany.

Contents

1	Introduction	3
2	Economics as a Complexity Science	5
2.1	Applying machine learning methods to economic problems	5
2.2	Machine learning for time series forecasting in Economics	6
2.3	The usefulness of ANNs to study complexity science	7
3	Detailed description of the proposed research	8
3.1	Task 1: Micro-emulation of the behaviour of economic agents	9
3.2	Task 2: Macro-emulation of an entire ABM simulation model	11
3.3	Task 3: Reduction of the complexity to design ANNs	12
3.4	Task 4: Reinforcement learning in economic policy design	13
4	Conclusions and final remarks	14

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 networks (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 ultimate aim of the research programme described here is to obtain ANNs that can emulate the time series dynamics of a pre-existing ABM simulation model. In other words, it is a meta-model, or a simulation of a simulation. 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 finally could also be used for empirical validation and estimation. There are currently several research efforts under way to construct agent-based macroeconomic models (Dawid et al., 2014; Dosi et al., 2014; Grazzini and Delli Gatti, 2013). These models aim to compete with standard Dynamic Stochastic General Equilibrium (DSGE) models that are currently in use by ECFIN and most Central Banks around the world. Using such models for policy analysis requires that they are calibrated and estimated on empirical data. For this, we need new methods and techniques to estimate such policy-oriented ABMs.

This paper deals with computer simulation models in the field of Agent-based Computational Economics (Epstein and Axtell, 1996; Arthur et al., 1997). This is a relatively young field of research that lies on the boundary of several traditional subjects: economics, computer science, mathematics, and physics. Hence, the project draws on expertise from all these fields.

In many cases, we do not know the correct equations of the economic model, and we might only know the behaviour of the artificial economic agents approximately through observations of the empirical behaviour of their real-world 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. Once we have constructed a computational model that satisfies certain requirements (e.g., stock-flow consistency of accounting relationships or dynamic completeness of behavioural repertoires) we usually find that the model is realistic enough to reproduce several empirical stylized facts of macrovariables, such as GDP growth rates, inflation rates, and unemployment rates, but all too often it is computationally heavy.

Current large-scale agent-based simulation models (e.g., Dawid et al., 2014) 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).

The issue of computational intractability is ubiquitous. It 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. In this project we intend to approach the problem by taking advantage of machine learning techniques, in particular recent developments in artificial neural networks, such as Deep Learning, Deep Belief Networks, Recursive Networks and Restricted Boltzmann Machines.

The scientific relevance and innovativeness of the project is that it tries to solve the generic problem of computational tractability of computer simulation models (and more specifically, agent-based economic models), not by resorting to technological solutions (e.g., parallel computing, or GPU grids), but by using machine learning algorithms in order to reduce the computer simulation to a lighter form, by emulating them using artificial neural networks, and by then adopting that simulation model to obtain results.

In order for agent-based models to be useful for policy analysis in so called "what-if scenarios" that are used to test counter-factuals in would-be worlds, it is a necessary condition to use models with a sufficiently high resolution in terms of behavioural and institutional details. The target models considered in this proposal are large-scale agent-based models where large-scale here means on the order of millions of agents. High-resolution may refer to the resolution of time-scales, geographical scales, decision-making scales, 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 policies (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 include: What happens when the biggest banks go bankrupt? Or: What happens when Greece leaves the Euro? Obviously, these are not things we want to simply test in the real world, considering the detrimental consequences and the ethical objections. The disadvantage is that such large-scale ABMs are quite heavy from a computational perspective. 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., to test whether the empirical stylized facts are still reproduced for many parameter settings. This involves performing a parameter sensitivity analysis and a robustness analysis against small changes in economic mechanisms, or with respect to changes in the behavioural repertoire 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 those social sciences where computer simulation models are being actively pursued (e.g., economics, sociology, econophysics) there are many discussions surrounding the empirical estimation and validation of such models (e.g., Werker and Brenner, 2004; Brenner and Werker, 2006; Fagiolo et al., 2007; Grazzini et al., 2012; Grazzini and Richiardi, 2013; Grazzini et al., 2013; Yildizoglu and Salle, 2012). However, until now, no clear consensus has appeared on how to solve this problem. In econometric applications, some advances have been made on the estimation of ABMs. Noteworthy are two approaches, one using non-parametric bootstrap methods (Boswijk et al., 2007) and the other using estimation of a master equation derived from the Focker-Planck equation (Alfarano et al., 2005; Aoki and Yoshikawa, 2007; Di Guilmi et al., 2008). Currently there are several projects under way to construct agent-based macroeconomic models: the Eurace@Unibi model (Dawid et al., 2014), the Crisis Project (Grazzini and Delli Gatti, 2013), and the "Keynes meeting Schumpeter" models (K+S models, Dosi et al., 2010, 2013, 2014). These models consider it a feature, not a vice, to model the agents and their behavioural repertoires in great detail, by taking care that all the behavioural assumptions are supported by empirical evidence.

2 Economics as a Complexity Science

Several research groups spread out across Europe are pursuing the broad research themes of complexity science, and try to apply them to economics, such as the theory of networks, research on innovation, growth and financial fragility, and using models for policy analysis. Over the past decade several EU-funded research projects have made some headway in this overarching theme of 'Economics as a Complexity Science'. Examples of such research projects are: ComplexMarkets, Eurace, FOC-II, FinNov, Crisis, and SimPol. We are well aware of the progress that was made in each of these projects. At the same time, we are also convinced that much remains to be done in the field of agent-based macroeconomics, linking the real- and financial sectors, input-output models, supply chains, stock-flow consistent modelling, and many more subfields of agent-based economic modelling.

2.1 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 being also one of the founding fathers of modern-day Artificial Intelligence (AI), it seems only appropriate that in applying artificial neural networks to economic problems, we rely on various aspects of Simon's 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, and are able to compute with full accuracy the outcome of their actions. However, a substantial amount of time of real decision makers is being spent 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 man: "It is better to be approximately right, rather than being 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: (i) Imperfect information gathering, or incomplete observation of outcomes. (ii) Limitations in storage capacity or faulty interpretation of those observations (imperfect recall, bad documentation of results). (iii) Limits in processing abilities. (iv) 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; Dawid et al., 2014). 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 Machine learning for time series forecasting in Economics

Applications of ANNs to time series forecasting problems in economics include: financial market forecasting (Trippi, 1993; Azoff, 1994; Refenes, 1995; Gately, 1996), foreign exchange rates (Weigend, 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, 1994). A review of applications of ANNs in the field of Management Science and Operations Research is given by Wilson (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.

2.3 The usefulness of ANNs to study complexity science

According to Gorr (1994), ANNs are very appropriate in the following situations: (i) large data sets; (ii) problems with nonlinear structure; (iii) multivariate time series forecasting problems. Important issues that can be addressed include:

- (1) How do 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 (a.k.a. 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. In such cases Hessian-Free (HF) optimization methods or Momentum Methods such as Nesterov's Accelerated Gradient (NAG) seem more promising (see Section 3, Task 3 for more details).

The paper suggest a number of optimizations for RNNs. We believe one of the most relevant for our problem is that of filtered, low-pass filter inputs. These are nodes with self-weights close to 1, (similar to exponential filters) that allow non-linearities to persist longer and not disappear at the next step. This is coupled with non-linearities modeled e.g. as $out = \max(0, in)$ rather than a sigmoid or tanh function. There is justification for the approaches and some promising results (although this is by no means a solved problem) in that the output of the error function is "rough" and requires some form of control for local cliffs that lead to local minima. All of the methods proposed in the literature are gradient-tracking in one way or the other, and are conservative about sudden changes. Hessian-free optimization (Martens and Sutskever, 2011) and the PhD-thesis by Sutskever (2013) show the applicability of such methods in the multivariate time series domain.

- (2) Given a specific forecasting problem, how do we systematically build an appropriate network that is best suited for the problem?

We follow 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 point of entry. 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), for example, in order to fix the weights on the input nodes.

- (3) What is the best training method or algorithm for forecasting problems, particularly 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. The doppelganger ANN described in task 1 (micro-emulation) is not an RNN, but

is rather an actuator based on the inputs. It *can* have memory of its own actions, but it is still distinctively different from an RNN that models a time series. Hence, we should find different best practices for each of our sub-tasks. In the task descriptions in Section 3 we make initial propositions for each of the task descriptions.

- (4) How should we 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.

All these questions are addressed in more detail below.

3 Detailed description of the proposed research

In order to use macroeconomic agent-based models for policy, we need to reduce the complexity of the ABM simulation to a less complex, more computationally tractable system. In other words, surrogate models or meta-modelling approaches need to be developed, 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 a four-step procedure:

1. Construct an ABM and generate synthetic time series data.
2. Train a multi-layered, deep neural network on the synthetic data.
3. Reduce the complexity of the neural network to allow for efficient computation.
4. Apply the meta-model to economic policy analysis.

According to these four steps, the work in the research project can be structured into four broad research themes (a full description of each task is given below):

Task 1: Micro-emulation of the behaviour of single agents.

Task 2: Macro-emulation of an entire ABM simulation.

Task 3: Reduction of the complexity to design ANNs.

Task 4: Reinforcement learning in economic policy design.

In the end, the application of the Deep Learning algorithms to economic ABMs allows us to perform the following specific tasks:

1. Emulation of agent-based models using artificial neural networks (ANN-emulation).
2. Estimation with empirical data using the ANN-emulation.
3. Parameter sensitivity analysis using the ANN-emulation.
4. Policy analysis using the ANN-emulation.

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

3.1 Task 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. In the end, the original agents may be replaced by their Doppelgänger, 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 task 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.

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 to hold simulated purchase surveys among its consumers (Nagle and Hogan, 2006). Once a year, the firm considers a sample of households to present them 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 c.q. 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 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 way of 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-parameteric 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. 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. 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 firm, but also the default on individual loans. In order to make such an assessment, the banks 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 agent-based macroeconomic model (Eurace@Unibi), the bank uses a highly simplified model to determine the probability of default of the firms to which they have outstanding loans, or of new firms that make credit requests. The essential aspect of the model is that the bank's assessment of the firm's probability of default is based on balance-sheet data of the firm, and derived from the firm's equity and financial ratios such as the debt-equity-ratio, an indicator of financial fragility.

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 research project. But 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, but it 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 its 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 research should involve both:

- Offline learning on the time series data coming from an agent-based model that is detached from the agent. One can think of this as an outside-observer-approach.
- Online learning on time series data being generated by the model while the learning agent itself is part of this model.
- If multiple agents are simultaneously using online learning in this sense, we can speak of **co-evolutionary learning** by heterogeneous, artificial neural network agents.

The aim of this task is to focus on the appropriate network structure to emulate the multivariate time series generated by the target system (in this case, the particular agents to emulate).

The final results of Task 1 are:

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 Task 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 in Task 1, this is a global approach. 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 been trained just on the empirical, historic data, since this is just a single 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 forecasting, with the difference that instead of estimating the future values based on the past values, the input to the ABM are the actions of the agents in the model. A recurrent neural network (RNN) is the type of network that is the obvious choice for such a task. 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.

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 task will be informed by other tasks, e.g. the modelling in Task 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 task 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 results of Task 2 are:

1. A deep layered ANN that is trained on data generated by an ABM.
2. A deep layered ANN that can be used for empirical estimation, and for policy analysis.

3.3 Task 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 goals of this task are:

1. Focus on Hessian-Free Optimization and Momentum Methods, and possibly adapt these methods to our specific problems.
2. Optimize the choice of network parameters: the number of input nodes, and the number of hidden layers.

The final results from Task 3 are:

1. Designing good initializations of the network parameters for the Deep Learning algorithms.
2. Develop insights to inform the optimization methods and the Deep Learning algorithms.

3.4 Task 4: Reinforcement learning in economic policy design

The final task 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, the 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.

We distinguish between two cases, namely "offline training" versus "online learning".

Offline training of the ANN-policy-agent.

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.

Online learning by the ANN-policy-agent.

In this 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 goals of this task are:

1. Focus on which reward functions and what network structures are most appropriate for the ANN-policy-agent.
2. Design the endogenous policy setting behaviour for the ANN-policy-agent: behavioral heuristics, rules to adapt these heuristics, as well as the parameters for the ANN.

The final result of Task 4 is 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 Conclusions and final remarks

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 statistical 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 of the research project is to develop new methods and techniques to improve the applicability of macroeconomic ABMs for economic policy analysis.

For the practical implementation of this goal, we need to make advances in two domains:

- Deep Learning: developing new machine learning techniques to represent ABMs by ANNs (Tasks 1-2).
- Complexity Reduction: developing new complexity-reduction techniques to guide the design of ANNs (Task 3).

The work to be done consists of the following broad research tasks:

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, tasks 1 through 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 project connects both micro- and macroeconomics, and joins both estimation and emulation in machine learning.

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.

References

- Alfarano, S., Lux, T., Wagner, F., 2005. Estimation of Agent-Based Models: The Case of an Asymmetric Herding Model. *Computational Economics* 26 (1), 19–49.
- Aoki, M., Yoshikawa, H., 2007. *Reconstructing Macroeconomics*. Cambridge Books. Cambridge University Press.
- 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.

- Arthur, W. B., Durlauf, S., Lane, D., 1997. *The Economy as an Evolving Complex System II*. Reading, MA, Reading, MA.
- Azoff, E., 1994. *Neural Network Time Series Forecasting of Financial Markets*. John Wiley and Sons, Chichester.
- Bacha, H., Meyer, W., 1992. A neural network architecture for load forecasting. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*. Vol. 2. pp. 442–447.
- Bengio, Y., Boulanger-Lewandowski, N., Pascanu, R., 2013. Advances in optimizing recurrent networks. Vol. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 8624–8628.
- Boswijk, H. P., Hommes, C. H., Manzan, S., 2007. Behavioral heterogeneity in stock prices. *Journal of Economic Dynamics & Control* 31 (6), 1938–1970.
- 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*. Prentice Hall, Englewood Cliffs.
- 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., Gemkow, S., Harting, P., van der Hoog, S., Neugart, M., 2014. *Agent-Based Macroeconomic Modeling and Policy Analysis: The Eurace@Unibi Model*. Handbook on Computational Economics and Finance. Oxford University Press.
- Dawid, H., Harting, P., 2012. Capturing firm behavior in agent-based models of industry evolution and macroeconomic dynamics. In: Bünsdorf, G. (Ed.), *Evolution, Organization and Economic Behavior*. Edward Elgar, Ch. 6.
- Dawid, H., Neugart, M., 2011. Agent-based models for economic policy design. *Eastern Economic Journal* 37.
- Di Guilmi, C., Gallegati, M., Landini, S., 2008. Modeling Maximum Entropy and Mean-Field Interaction in Macroeconomics. *Economics Discussion Papers 2008-36*, Kiel Institute for the World Economy.
- Dosi, G., Fagiolo, G., Napoletano, M., Roventini, A., 2013. Income distribution, credit and fiscal policies in an agent-based Keynesian model. *Journal of Economic Dynamics & Control* 37, 1598–1625.
- Dosi, G., Fagiolo, G., Roventini, A., 2010. Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles. *Journal of Economic Dynamics & Control* 34, 1748–1767.
- Dosi, G., Napoletano, M., Roventini, A., Treibich, T., 2014. Micro and Macro Policies in Keynes+Schumpeter Evolutionary Models. LEM Papers Series 2014/21, Laboratory of Economics and Management (LEM), Sant'Anna School of Advanced Studies, Pisa, Italy.
- Duffie, D., Singleton, K. J., 2003. *Credit Risk: Pricing, Measurement, and Management*. Princeton University Press, Princeton.
- Epstein, J. M., Axtell, R. L., 1996. *Growing Artificial Societies: Social Science from the Bottom Up*. MIT Press.
- Fagiolo, G., Moneta, A., Windrum, P., October 2007. A Critical Guide to Empirical Validation of Agent-Based Models in Economics: Methodologies, Procedures, and Open Problems. *Computational Economics* 30 (3), 195–226.
URL <http://ideas.repec.org/a/kap/compec/v30y2007i3p195-226.html>
- 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.
- Gately, E., 1996. *Neural Networks for Financial Forecasting*. John Wiley, New York.
- Gorr, W. L., 1994. Editorial: Research prospective on neural network forecasting. *International Journal of Forecasting* 10 (1), 1–4.
- Grazzini, J., Delli Gatti, 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.
- 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., 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. 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.
- Kang, S., 1991. An Investigation of the Use of Feedforward Neural Networks for Forecasting. Ph.D. thesis, Kent State University.
- 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), 347–64.
- Maasoumi, E. ; Khotanzad, A. . A. 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. *Artificial Intelligence and Statistics*, 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)*.
- Martens, J., Sutskever, I., 2012. Training deep and recurrent networks with Hessian-Free optimization. In: Montavon, G., Orr, G. B., Müller, K.-R. (Eds.), *Neural Networks: Tricks of the Trade*. Vol. 7700 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 479–535.

- Nagle, T., Hogan, J., 2006. *The Strategy and Tactics of Pricing: A Guide to Growing More Profitably*. Pearson Prentice Hall, New Jersey.
- Refenes, A., 1993. Constructive learning and its application to currency exchange rate forecasting. Probus Publishing Company, Chicago.
- Refenes, A., 1995. *Neural Networks in the Capital Markets*. John Wiley, Chichester.
- Sargent, R. G., 2011. Verification and validation of simulation models. In: *Proceedings of the Winter Simulation Conference. WSC '11. Winter Simulation Conference*, 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*. The MIT Press, Cambridge, MA, (3rd ed.).
- Simon, H. A., 1997 [1947]. *Administrative Behavior*. The Free Press, New York, NY, (4th ed.).
- Srinivasan, D., Liew, A., Chang, C., 1994. A neural network short-term load forecaster. Vol. 28. pp. 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*. North-Holland.
- Trippi, R.R., T. E., 1993. *Neural Networks in Finance and Investment: Using Artificial Intelligence to Improve Real-world Performance*. Probus, Chicago.
- Weigend, A., Gershenfeld, N., 1993. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, Reading, MA.
- Weigend, A.S., H. B. R. D., 1992. Predicting sunspots and exchange rates with connectionist networks. Addison-Wesley, Redwood City, CA, pp. 395–432.
- Werker, C., Brenner, T., May 2004. Empirical Calibration of Simulation Models. *Papers on Economics and Evolution 2004-10*, Philipps University Marburg, Department of Geography.
- Wilson, R., S. R., August 1992. Neural networks. *OR/MS Today*, 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.