Neuroevolutionary Approach
to Metamodel-Based Optimization
in Production and Logistics

Ilya Jackson
Transport and Telecommunication Institute (TTI), Lomonosova Iela 1,
Riga, Latvia
jackson.i@tsi.lv

Abstract. Considering recent success of neuroevolutionary approaches in automated machine learning, their application to metamodeling seems to be promising. This paper examines, whether it is feasible and efficient to combine artificial neural network with genetic algorithm for metamodeling automation of logistic and production systems. Besides, the possibility of using the proposed approach to derive optimal control parameters is discussed.

Keywords: Metamodeling · Optimization · Production · Logistics · Inventory control · Post-pandemic development

1 Introduction

The recent COVID-19 pandemic has tested sustainability of supply chains to its limits and revealed serious vulnerabilities of logistic and production systems across the globe. As a result, evolution of traditional ways of creating and distributing value has been notably accelerated [1].

Problems in production and logistics are industry-specific and contain non-standard features. This fact along with complexity, dimensionality and stochasticity frequently leads to analytic intractability. That is why, approximate solutions, simulation-based optimization and derivative-free “black-box” optimization are common choice for solving real-world problems in production and logistics. Nevertheless, realistic and detailed models require sheer amount of computational resources. In the light of this fact, it is quite reasonable to take advantage on an alternative model, which is specifically built in order to approximate an original one with a tolerable degree of accuracy. Such an approach is widely known as metamodeling [2].

Nowadays such machine learning (ML) algorithms as artificial neural networks (ANN), support-vector machines, gradient boosting algorithms and many others demonstrate state-of-the-art performance in such nontrivial tasks as image recognition, machine translation, speech recognition, and fitness approximation. This undisputable success could not bypass metamodeling. Smarter solutions that take advantage on ML-based metamodels significantly accelerate optimization search, thus lead to faster decision-making allowing companies to maintain business continuity in the face of disruptions and uncertainties, which is especially important in post-pandemic times.
Despite the promising potential of ML algorithms in metamodeling, there is a serious challenge that cannot be ignored. Although ML algorithms are universal and task-unspecific, they are sensitive to a plethora of design decisions. For example, data preparation and pre-processing, feature selection and extraction, model selection and hyperparameter optimization [3]. That is why, data scientists, artificial intelligence developers and other experts are deeply involved into development, fine-tuning and customization of ML models. This procedure is labor-intense and requires additional time of highly qualified specialists along with financial resources. Automation of machine learning (AutoML) clearly has potential to make advanced ML-based metamodels accessible to domain scientists working in the field of logistics and production, which can be considered as “democratization” of ML. However, recent publications emphasize that widespread industrial applications require AutoML solutions to be tailored for solving a particular class of problems within the domain [3–5].

2 Related Work

There are dozens of ML-based solutions for metamodeling of production and logistic systems. Jarosz et al. compared such metamodeling approaches as ANN, support-vector machines and random forest based on a two-stage production process of blister copper [6]. Yang et al. have demonstrated advantages of ML-based metamodeling techniques for optimization of additive manufacturing systems [7]. The recent paper reports on the computational framework for industrial lot-sizing. The framework incorporates the concepts of parallel computing, metaheuristics and ML (gradient tree boosting regressor) for solving an economic lot-size problem [8].

Notwithstanding the astonishing performance of ML metamodels there is a serpent in this Eden. Although the vast majority of ML models are universal function approximators, human experts still play a significant role. Taking into consideration recent success of neuroevolutionary approaches in neural architecture search and hyperparameter optimization, their application to metamodeling in logistics and production becomes promising.

3 Methodology

The neuroevolution comprises neural and evolutionary components. In the proposed computational framework GA is chosen as the evolutionary component and multilayer perceptron (MLP) as the neural one.

3.1 Neural Component

The computational framework for automated metamodeling relies on the extension of the universal approximation theorem. The theorem states that any feedforward fully connected ANN with at least one hidden layer, finite number of neurons and non-linear activation function can approximate nonlinear input-output maps [9]. MLP is the most general feed-forward ANN, and other feedforward networks can be considered as its
specific case. For this reason, MLP is used as a baseline for a metamodel. MLP derives its computational efficiency and robustness from distributed structure and capability to generalize. The generalization refers to the ability of producing correct output for inputs that have not been encountered during the learning phase.

In the proposed computational framework, the search space defines the plethora of attainable structures. Since this paper is rather focused on a proof of concept and does not attempt to develop the all-embracing solution, the search space contains only the most vital components of MLP, such as depth, width, activation functions, optimizer and learning rate.

3.2 Evolutionary Component

Mimicking Darwinian natural selection, GA appears to be the driving force behind the neuroevolution. GA samples, crosses over, mutates and resamples the population of highly fit MLP-based metamodels in order to form a new even fitter population. As the result, instead of exhaustive search for nearly optimal topologies, GA improves the solution step by step from the most promising topologies. The implemented GA is distinguished by Gray-code chromosome encoding, uniform crossover and tournament selection. Components that define the search strategy.

An appropriate chromosome representation for the particular problem domain can drastically accelerate the search. Space-efficiency makes binary representation an apparent choice, however, chromosome representations vary noticeably even among binary schemes. The choice falls on the reflected Gray-code, because such encoding scheme can guarantee gradualism for the reason that exactly one bit has to be flipped in order to reach the nearest value [10]. This fact prevents problems related to large Hamming clefts (Fig. 1).

![Fig. 1. Components of the MLP are encoded into the Gray’s code.](image)

Uniform crossover compensates a plethora of serious drawbacks of more traditional crossovers. Since the number of crossover points is not constant, the decision of flipping is made independently for every single bit, thus, the operator will not tend to split up the fit sections of a chromosome. Additionally, the uniform crossover makes
premature convergence less likely to happen [11]. Tournament selection is a standard selection operator in contemporary neuroevolutionary techniques. It is distinguished by several significant benefits over alternatives, for instance it is simple to implement and robust to noisy fitness functions [12].

3.3 Metamodeling via Neuroevolution

In order to formalize automated metamodeling via neuroevolution, let’s consider a nonlinear input-output map represented by the “black-box” function \( y = f(x) \), where \( x \) stands for the vector of input parameters and \( y \) is the output. Despite \( f(x) \) is a “black-box”, the set of observations \( \tau = \{ x_i, y_i \}_{i=1}^{N} \) is given. Since MLP-based metamodel can be built from such components as layers, activation functions, optimizers and respective hyperparameters, this information is encoded into a single chromosome \( a \in A \). After that GA searches for such \( a^* \in A \) that allows MLP-based metamodel \( F(a^*, x) \) to approximate the original model. Such that \( \|F(a, x) - f(x)\| < \varepsilon \), where \( \varepsilon \) is a positive value, small enough for the problem under consideration (Fig. 2).

![Fig. 2. Metamodelling automation with neuroevolution.](image-url)

In the following numerical studies, the search space encompasses the number of hidden layers from 0 to 10 and the number of neurons in each layer from 10 to 200 with the discretization step of 1. Additionally, the search space includes learning rates from 0.01 to 0.3 with the discretization step of 0.01 and all the activation functions and optimizers available in Keras 2.3.0 [13]. So that GA orchestrates evolutionary morphism to find a structure in feasible region that produces the metamodel with the highest coefficient of determination (\( R^2 \)). The total number of trainable parameters that
include both neurons and weights is restricted to 20000, because the obtained metamodel has to be more computationally efficient than the original one.

3.4 Metamodel-Based Optimization

An accurate metamodel can significantly ease computational burden. It can be trained once based on a dataset derived from experiments with the original model. After that optimization search can be conducted iteratively. There is a plethora of suitable optimization algorithms, but GA is a convenient choice for demonstration. The computational framework for metamodeling is already equipped with GA, which is universal can be applied for both neuroevolution and the following optimization. The logic behind metamodel-based optimization with GA is the same as for the network morphism. To be more specific, by applying genetic operators to the parental solutions, a set of new candidate solutions is obtained. After that, the produced offspring compete for a place in the next generation based on its fitness. This process continues until termination criteria is met (Fig. 3).

![Fig. 3. The logic behind GA-driven metamodel-based optimization.](image)

4 Numerical Study

This research takes into consideration the ongoing reproducibility crisis. Since the reproducibility of experiments is the cornerstone of the scientific method, all the source-code of models, metamodels and computational framework are available in the GitHub repository [14].

4.1 Systems Under Consideration

The models of the considered systems are implemented in the form of a discrete-event simulation. The first model simulates an inventory control system and referred to as the
model 1. It is mainly based on the recent work [15]. The model can be characterized as multiproduct and stochastic with lost-sales, discounts and perishability. The second model simulates production-inventory system and referred to as the model 2. The model adopts Markov-modulated demand, in which the demand process is related to an underlying Markov chain [16]. For both models averaged net-profit is considered as an output of interest (dependent variable).

4.2 Metamodeling as a Regression Problem

The procedure begins with generation of random inputs in the feasible range. The model 1 comprises 151 independent variables and replicated 35 times. The model 2 comprises 81 independent variables and replicated 44 times. Storing the vector of inputs and the average output as an observation, the datasets $\tau_1$ and $\tau_2$ of 600 observations are generated. Both datasets are split by training and test subsets using 10-fold cross validation. As soon as $\tau$ is generated, metamodeling can be formulated as multivariate regression problem. Therefore, the metamodeling procedure can be reduced to minimization of the mean squared error (MSE) between the actual output $y$ and prediction made by the metamodel $F(a, x)$.

The data is standardized prior to training. AutoML procedure is driven by GA with the empirically adjusted parameters. Namely, tournament size of 5, crossover probability of 0.30 and mutation probability of 0.05. It is worth to point out that slight variations of hyperparameters have not notably affected either the convergence speed or the likelihood of premature convergence. The evolution lasts 10 generations, and each generation is populated with 40 MLPs. In 10 generations the metamodels with the average coefficients of determination of 0.81 and 0.82 have been trained (Fig. 4).

![Fig. 4. Evolution of the MLP-based metamodels (vertical arrows represent the standard deviation of the fitness function within the generation).](image)

Coefficients of determination calculated based on all the cross-validation slices are insignificant, which means that the MLPs have been capable to generalize nonlinear relations between the inputs and output (Fig. 5). Furthermore, violin plots demonstrate
that the original samples generated by the simulation and the samples recreated using the trained metamodels belong to the same population (Fig. 6).

The Table 1 contains several accuracy-related statistics. Besides, ordinary least squares regression is used in order to demonstrate the presence of nonlinearity.

**Table 1.** Accuracy of the fittest MLPs compared to ordinary least squares.

| Models | Training sample | Test sample | Ordinary least squares |
|--------|-----------------|-------------|------------------------|
|        | MSE $R^2$ | $\tilde{R}^2$ | MSE $R^2$ | $\tilde{R}^2$ | MSE $R^2$ | $\tilde{R}^2$ |
| 1      | 0.07 0.88 0.86 | 0.20 0.81 0.79 | 0.42 0.55 0.52 |
| 2      | 0.09 0.86 0.84 | 0.18 0.82 0.80 | 0.67 0.39 0.38 |

Fig. 5. Box-plot of coefficients of $R^2$ calculated in 10-fold cross-validation.

Fig. 6. Violin plots of the net profit generated with the model and recreated using the metamodel.
In the context of regression analysis, the role of residuals should not be disparaged. Residuals provide a general approach to indicate the quality of the metamodels and diagnose possible problems. The expected value of residuals for both metamodels is close to 0, namely $-0.011$ and $0.013$ for the model 1 and 2 respectively. It implies that both metamodels are not systematically biased toward over-prediction or under-prediction. After the distributions of residuals are plotted against theoretical standard normal, the possible issue related to heavy tails immediately catches the eye (Fig. 7).

![Quantile-quantile plots of the residuals.](image)

Nevertheless, the Anderson–Darling normality test is confidently passed in both cases (Table 2). The null hypothesis that a sample is taken from a normally distributed population is tested. Since the calculated statistic is smaller than the critical value for both samples, the null hypothesis that the data is drawn from normal distribution can be accepted for the significance level 0.05. It is also worth to test, whether residuals are homoscedastic. In practical sense, if residuals are heteroscedastic, this implies that the predictive power of the model is different for different sections of the data. Applying Goldfeld–Quandt test for heteroscedasticity, the null hypothesis that the two subsamples of the dataset have the same variance is not rejected for residuals of both metamodels (Table 3).

![Quantile-quantile plots of the residuals.](image)

**Table 2.** Anderson–Darling normality test for residuals

| Models | Statistic | Critical value | Significance level |
|--------|-----------|----------------|--------------------|
| 1      | 0.523     | 0.787          | 0.05               |
| 2      | 0.619     | 0.787          | 0.05               |
4.3 Metamodel-Based Optimization

For the model 1 and model 2 all the inputs except control parameters are randomly generated producing 30 test scenarios. For each scenario, the output (net profit) is subject to maximization. Net-profit achieved by the metamodel-based optimization is compared with net-profit obtained by classical simulation-based optimization driven by the same GA. Metamodel-based optimization and simulation-based optimization last 31 generations with 100 candidate solutions in each generation. Percentage error is used for comparing the accuracy (Fig. 8).

| Models | Statistic | Critical F-value | p-value | Significance level |
|--------|-----------|------------------|---------|--------------------|
| 1      | 1.1       | 1.23             | 0.38    | 0.05               |
| 2      | 0.97      | 1.30             | 0.43    | 0.05               |

**Fig. 8.** Metammodel-based optimization is compared to simulation-based optimization.

Mean percentage errors of 5.21 and 3.23 can be interpreted as insignificant taking into account complexity, computational budget and presence of stochastic noise. It is important to keep in mind that MLPs are approximations and an error is inevitable.

5 Conclusions

MLP-based metamodels derived using neuroevolution have demonstrated capability to learn and generalize complex nonlinear relations between variables. Therefore, they can be applied for metamodeling of complex real-world systems in logistics and production. The experiments have been conducted to proof the concept, however, the accuracy of the metamodels can be improved further by performing more exhaustive search. However, exhaustive search is associated with an increase in the computational budget and the common trade-off between accuracy and computational efficiency.
Optimization using MLP-based metamodels has several distinct advantages over the classical approaches. Firstly, the output of the MLP-based metamodel is always deterministic. Secondly, MLPs, by definition, are computationally less expensive, which is especially notable for shallow architectures that use activation functions from the linear unit family.

References

1. Lee, J., Wright, J.: A COVID-19 and shattered supply chains: Reducing vulnerabilities through smarter supply chains. IBM institute for business value (2020)
2. Blanning, R.: The construction and implementation of metamodels. Simulation 24, 177–184 (1975)
3. Hutter, F., Kotthoff, L., Vanschoren, J.: Automated Machine Learning. Springer, New York, NY, USA (2019)
4. He, X., Zhao, K., Chu, X.: AutoML: A Survey of the State-of-the-Art. In: arXiv preprint arXiv:1908.00709 (2019)
5. Liang, J., Meyerson, E., Hodjat, B., Fink, D., Mutch, K., Miikkulainen, R.: Evolutionary neural AutoML for deep learning. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 401–409 (2019)
6. Jarosz, P., Kusiak, J., Malecki, S., Morkisz, P., Oprocha, P., Pietrucha, W., Sztangret, L.: Metamodeling and optimization of a blister copper two-stage production process. JOM 68 (6), 1535–1540 (2016)
7. Yang, Z., Hagedorn, T., Eddy, D., Krishnamurty, S., Grosse, I., Denno, P., Lu, Y., Witherell, P.: A domain-driven approach to metamodeling in additive manufacturing. In: ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers Digital Collection, pp. 1–10 (2017)
8. de Sousa Junior, W., Montevechi, J., Miranda, R., Rocha, F. and Vilela, F.: Economic lot-size using machine learning, parallelism, metaheuristic and simulation. Int. J. Simul. Model. 18(2), 205–216 (2019)
9. Hanin, B.: Approximating Continuous Functions by ReLU Nets of Minimal Width. arXiv preprint arXiv:1710.11278 (2018)
10. Caruana, R., Schaffer, J.: Representation and hidden bias: Gray vs. binary coding for genetic algorithms. In: Proceedings of the 5th International Conference on Machine Learning, pp. 153–161. San Mateo, CA: Morgan Kaufmann (1988)
11. Michalewicz, Z.: Evolution strategies and other methods. In: Genetic Algorithms + Data Structures = Evolution Programs, pp. 159–177. Springer, Berlin, Heidelberg (1996)
12. Miller, B., Goldberg, D.: Genetic algorithms, tournament selection, and the effects of noise. Complex Syst. 9(3), 193–212 (1995)
13. Chollet, F.: Keras: The python deep learning library. Astrophysics Source Code Library (2018)
14. GitHub repository “metainventory”, https://github.com/Jackil1993/metainventory Accessed 20 July 2020
15. Jackson, I.: Simulation-optimisation approach to stochastic inventory control with perishability. Inf. Technol. Manag. Sci. 22, 9–14 (2019)
16. Jackson, I.: Neuroevolutionary approach to metamodeling of production-inventory systems with lost-sales and markovian demand. In: International Conference on Reliability and Statistics in Transportation and Communication, pp. 90–99, Springer, Cham (2019)