Scientific article
UDC 330.47
DOI: https://doi.org/10.57809/2024.3.3.10.6

POLYNOMIAL NETWORKS INSTEAD OF NEURAL NETWORKS

Sergey Svetunkov ✉

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ sergey@svetunkov.ru

Abstract. Neural networks are widely used in various scientific fields and practical research. They are sometimes implemented in the modeling of nonlinear economic dynamics. However, neural networks are often not suitable for modeling nonlinear economics. An effective alternative to neural networks in economics is the Elementary image of the Kolmogorov-Gabor polynomial. It has proven to have a more powerful ability to model nonlinearity than the artificial neural network. At the same time, the coefficients of this polynomial are estimated much simpler and faster than the coefficients of the artificial neural network. This observation provides grounds for the idea to replace neurons in the network by the Elementary images of the Kolmogorov-Gabor polynomial, thus creating an alternative polynomial network. This network is trained in just a few steps, while a neural network is trained over several tens of thousands of steps. Additionally, a Bayesian approach can be applied to polynomial networks, while it is not possible with neural networks. What is more, polynomial networks describe nonlinear processes no worse, and some-times even better, than neural networks. Therefore, when modeling nonlinear economic processes, polynomial networks not only prove to be simpler and faster in calculations, but also are capable of Bayesian parameter re-estimation with significant accuracy.

Keywords: neural networks, polynomial networks, Kolmogorov-Gabor polynomial, elementary image KGp, nonlinear economic dynamics

Citation: Svetunkov S. Polynomial networks instead of neural networks. Technoeconomics. 2024. 3. 3 (10). 57—71. DOI: https://doi.org/10.57809/2024.3.3.10.6

Научная статья
УДК 330.47
DOI: https://doi.org/10.57809/2024.3.3.10.6

ПОЛИНОМИАЛЬНЫЕ СЕТИ ВМЕСТО НЕЙРОННЫХ СЕТЕЙ

Сергей Светуньков ✉

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия

✉ sergey@svetunkov.ru

Аннотация. Нейронные сети активно используются в самых разных областях науки и в практических исследованиях. Встречаются случаи использования нейронных сетей в моделировании нелинейной экономической динамики. Но чаще всего нейронные сети оказываются малопригодными для моделирования нелинейной экономики. Эффективной альтернативой применению нейронных сетей в экономике может служить элементарный образ полинома Колмогорова-Габора. Показано, что элементарный образ полинома Колмогорова-Габора обладает более мощной способностью моделирования нелинейности, нежели модель искусственного нейрона. При этом коэффициенты этого полинома оцениваются значительно проще и быстрее, чем коэффициенты искусственного нейрона. Данное утверждение позволяет предложить замену нейронным сетям — вместо нейронов в сеть подставляются элементарные образы полинома Колмогорова-Габора и получается альтернативная полиномиальная сеть. Эта сеть обучается за несколько шагов в то время как нейронная сеть обучается за несколько десятков тысяч шагов. К тому же к полиномиальной сети применим байесовский подход, в то время как к нейронным сетям его использовать не удаётся. Показано также, что полиномиальные сети описывают нелинейные процессы не хуже, а иногда даже лучше, чем нейронные сети. В этой связи, при моделировании нелинейных экономических процессов предлагается использовать полиномиальные сети как более простые и быстродействующие в вычислениях, способные к байесовской переоценке параметров и не менее точные, чем нейронные сети.

Ключевые слова: нейронные сети, полиномиальные сети, полином Колмогорова-Габора, элементарный образ KGp, нелинейная экономическая динамика

Для цитирования: Светуньков С. Полиномиальные сети вместо нейронных сетей // Техноэкономика. 2024. Т. 3, № 3 (10). С. 57–71. DOI: https://doi.org/10.57809/2024.3.3.10.6

Introduction

The successful use of neural networks in natural and engineering sciences has prompted economists to search for opportunities to apply them to solve various tasks, including economic forecasting. However, attempts to use them for this purpose have not been very successful so far; after all, neural networks were created to solve image recognition tasks, and their application to modelling dynamic systems has been unsuccessful yet.

Since objects of economic forecasting often possess some inertia in their dynamics, it became possible to take this property into account in neural networks by using feedback connections. Such connections in neural networks are called “recurrent”. Therefore, neural networks with such connections are also called recurrent (RNN). If in simple neural networks the input data is considered unordered, in recurrent networks, the presence of feedback connections models the order of the data sequence, making them more suitable for modelling dynamic processes.

Materials and Methods

Currently, the scientific community is increasingly publishing results of successful applications of RNN in economic modelling and forecasting. There are some successful examples of economic forecasting using RNN. For instance, Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara (Hewamalage, Bergmeir, and Bandara, 2021) showed that RNN can be more accurate than such popular forecasting models as exponential smoothing (ETS) and autoregressive integrated moving average (ARIMA). However, unlike RNN, ETS and ARIMA models are reliable, efficient, convenient, and easily formalized. RNN, on the other hand, still represents poorly formalized models that need to be tailored for each case, with the success or failure depending on the qualifications of the researcher using them.

When considering the possibility of applying artificial intelligence and machine learning to economic forecasting, Stephan Kolassa wrote about the three major issues: scarce, opaque, and brittle data (Kolassa, 2020). And yet, the main tool of artificial intelligence and machine learning is neural networks! If we also consider that building RNN requires not only advanced programming skills but also knowledge of complex mathematical optimization methods that are a significant part of machine learning, it becomes clear why RNN in modelling nonlinear economic dynamics and economic forecasting are still relatively rare.

What needs to be done to make neural networks a usable tool in economic modelling and economic forecasting? How can we make them simpler so that any researcher applying mathematical methods in economic modelling but not possessing perfect forecasting skills could use them? How can we make the process of building and evaluating such a model simple and universal?

The answers to these questions can be found by turning our attention to alternative models. Pursuing this goal, the research focuses specifically on a mathematical model of an artificial neuron, the Kolmogorov-Gabor polynomial and the Wiener series, an elementary image of the Kolmogorov-Gabor polynomial, neural networks, and polynomial networks.

Results and Discussion

Mathematical Model of an Artificial Neuron

A neural network represents a set of j interconnected neurons. Each individual neuron has one or more inputs and one (and only one) output. Its mathematical model is a superposition of a linear multifactorial function and a non-linear function:

$$\hat{y}_j = f(a_0 + \sum_{i=1}^{n} a_i x_i) = f(y') \tag{1}$$

where:

$\hat{y}_j$ – output signal of the j-th neuron;

f – transfer function;

a_i – weight of the i-th signal (factor);

x_i – i- th component of the input signal (factor);

i = 1, ..., n – neuron input number;

n - number of neuron inputs;

a_0 – free coefficient;

y' – the result of the sum of the weighted input signals.

To avoid problems that may arise with data scales when working with neural networks, all variables are pre-normalized.

Artificial neuron models (1) differ from each other in the type of transfer function f(y'). Depending on the tasks that the researcher sets when forming a neural network, this transfer function can be a simple activation function, where the output signal takes a value of 0 or 1, or a more complex function that converts the sum of weighted input signals into a numerical output. This conversion can be performed using a linear or nonlinear function.

One of the simplest variants of an artificial neuron model is when the transfer function is written as:

$$\begin{cases} f(y') = 0, & y'_b \geq y', \\ f(y') = by', & y'_b < y' < y'_e, \\ f(y') = 1, & y'_e \leq y'. \end{cases} \tag{2}$$

Here, y'_b and y'_e are some predetermined constants of the minimum (base) and maximum (final) values of the output changes, and b is the proportionality coefficient.

At the output of the neuron, a signal is obtained that represents a superposition of two linear functions, and the coefficients ai and b can be easily estimated using statistical methods. But how can a nonlinear dependence between input factors and output be described using a linear neuron model? In order to do this, it is necessary to connect many linear neuron models to each other. And the more complex the configuration of such a neural network, the more accurately it will describe non-linearity. This process is reminiscent of the well-known piecewise-linear approximation method (Leenaerts, van Bokhoven, 2013). Since such networks represent a superposition of linear functions, their coefficients can be estimated using standard statistical methods, for example, the least squares method (LSM). However, since there are many coefficients in the neural network model, it is easier to solve this learning task using one of the numerical methods. Such simple neural networks are suitable for describing weak nonlinearities. The more complex the nonlinear phenomena being described, the more cumbersome the neural networks with piecewise-linear transfer functions become. Therefore, neural networks of this type are rarely used in practice.

Most often, an S-shaped nonlinear function, called a sigmoid, is used as a transfer function. Networks with such transfer functions excellently describe nonlinearities and are less cumbersome than networks with piecewise-linear transfer functions.

Among many possible functions with S-shaped forms, the most convenient ones are logistic:

$$\hat{y}_j = \frac{b_0}{b_1 + e^{-b_2 y'}} \tag{3}$$

and the hyperbolic tangent

$$\hat{y}_j = \frac{e^{y'} - b_0 e^{-y'}}{b_1 e^{y'} + b_2 e^{-y'}} \tag{4}$$

However, in practice, these transfer functions are often simplified by setting all their coefficients bi equal to one. Direct application of well-known statistical methods, such as the method of least squares, to estimate the coefficients of an artificial neuron (1) turns out to be impossible, since the mathematical model of an artificial neural network in this case represents a superposition of many nonlinear functions of parameters. Therefore, to estimate the coefficients of an artificial neuron and a neural network, one of the numerical methods is used, most often the gradient method. In the gradient method, as is known, the value of the first derivative (gradient in the multifactor case) is calculated. Functions (3) and (4) differ from many other sigmoidal functions in their derivatives that are expressed through the function itself, with the gradient method easily applicable to them. It is precisely for this reason that the logistic function and the hyperbolic tangent have become the most popular types of transfer functions in neural networks.

Hereinafter, when considering an artificial neuron model, we will assume that its transfer function is represented in the form of (3) or (4). We will not consider the transfer function in linear form.

Kolmogorov-Gabor polynomial and Wiener series

In the work "Theory of Functionals, Integral, and Integro-Differential Equations" from 1930, V. Volterra derived series that allow studying systems with soft inertial nonlinearities (Volterra, 1930). In 1958, N. Wiener, in his monograph "Nonlinear Problems in the Theory of Random Processes", presented a modification of Volterra's series for the discrete case (Wiener, 1958). This same problem for continuous processes was solved in 1956 by A.N. Kolmogorov (Kolmogorov, 1956), and in 1961, D. Gabor proposed a discrete variant of the "extended prediction operator" based on it (Gabor, Wilby, Woodcock, 1961).

It turns out that the same scientific research tool was developed independently by two groups of researchers: V. Volterra and N. Wiener, as well as A.N. Kolmogorov and D. Gabor. This mathematical tool can be referred to as "Wiener series" or the "Kolmogorov-Gabor polynomial".

In published scientific works, results of forecasting obtained via the "Kolmogorov-Gabor polynomial" are encountered. For example, Hamidreza Marateb and other authors use this polynomial to predict COVID-19 hospital stays (Marateb, Norouzirad, Tavakolian, et. al., 2023), and Wei Liu and colleagues use this polynomial to forecast electrical load (Liu, Dou, Wang, 2018). The term "Wiener Series" is not found in applied works but is used in published articles and monographs of mathematicians. For example, in the work of Wim van Drongelen, where the differences between the Wiener and Volterra series are discussed, which is helpful to derive the expressions for the zero-, first-, and second-order Wiener kernels (Wim, 2010). There are many other publications on solving practical problems where authors use not the concept of "Wiener series" but "Kolmogorov-Gabor polynomial" (Anjorin, Ricks, 2023; McElroy, Ghosh, Lahiri, 2024; Nelles, 2020; Razif, Shabri, 2023).

Since we are not considering the theoretical properties of this mathematical tool but are studying its practical applicability, we will adhere to the term that is commonly used in practical research, namely "Kolmogorov-Gabor polynomial" (hereinafter, KGp).

This polynomial can be represented in general form as:

$$y = a_0 + \sum_{i=1}^{m} a_i x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} x_i x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} a_{ijk} x_i x_j x_k + \ldots \quad (5)$$

For example, for two factors, it will take the following form:

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_{11} x_1^2 + a_{12} x_1 x_2 + a_{22} x_2^2 \quad (6)$$

It is obvious that estimating the values of 6 coefficients of such a model from statistical data does not present any difficulties. However, for three factors, the KGp will become significantly more complex and cumbersome, containing 20 unknown coefficients:

$$\begin{aligned} \hat{y} &= a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_1^2 + a_5 x_2^2 + a_6 x_3^2 + a_7 x_1 x_2 + a_8 x_1 x_3 + \\ & a_9 x_2 x_3 + a_{10} x_1^3 + a_{11} x_2^3 + a_{12} x_3^3 + a_{13} x_1^2 x_2 + a_{14} x_1^2 x_3 + a_{15} x_1 x_2^2 + a_{16} x_2^2 x_3 + \\ & a_{17} x_1 x_3^2 + a_{18} x_2 x_3^2 + a_{19} x_1 x_2 x_3 \end{aligned} \quad (7)$$

If the number of factors increases to i = 4, then the number of KGp coefficients will grow to N = 70, and for i = 5, the number of coefficients to be estimated becomes equal to N = 252.

In general, the number of KGp coefficients grows nonlinearly with the increase in the number of factors i. This number is calculated using a well-known formula from combinatorics.

The enormous dimensionality of the KGp construction task with many original variables limits the practical application of this tool for modeling nonlinear dependencies. Therefore, "It may be mentioned that parameter-saving approximations to KG polynomials have interested researchers for a long time" (Terasvirta, Kock, 2010). The prominent Ukrainian scientist A.G. Ivakhnenko proposed a method of step-by-step decomposition of the KGp model construction process, which he called a “multi-level system” (Ivakhnenko, 1963). He repeatedly used this method to solve practical problems and tried to popularize it (Ivakhnenko, 1971;1975). Frankly speaking, most scientific research using KGp employs the approach proposed by Ivakhnenko. Although it simplifies the method of estimating KGp coefficients, it remains cumbersome and is only suitable for cases with a small number of variables xi. The work (Svetunkov, 2024) shows that A.G. Ivakhnenko's method leads to the construction of a different polynomial, not KGp. For example, when i = 3, the polynomial constructed by Ivakhnenko's method contains 80 terms, while KGp in this case should consist of 20 terms. This means that Ivakhnenko's “multi-level system” represents a different model than the KGp model, and its scientific significance becomes unclear.

Elementary image of the Kolmogorov-Gabor polynomial

The fact that scientists have not been able to propose a method for constructing KGp that could overcome the “curse of dimensionality” (Ivakhnenko, 1963) has deprived science of such a powerful modeling tool as KGp for many years. However, instead of the full KGp, its simplified analog (Svetunkov, 2024) can be used, which excellently handles modeling nonlinearities. Let's consider this possibility in more detail. For any number of variables xi, i=1, 2, ..., m, affecting the variable y, a simple linear multifactorial model can be easily constructed:

$$\widehat{y'} = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_m x_m \tag{8}$$

The coefficients of this model are estimated by any statistical method, for example, LSM.

Then, the calculated values of the modeled variable need to be used in a polynomial of degree m:

$$\hat{y} = b_0 + b_1 \widehat{y'} + b_2 (\widehat{y'})^2 + \ldots + b_m (\widehat{y'})^m \tag{9}$$

The coefficients of such a polynomial can also be estimated using one of the statistical methods. As can be seen, it is necessary to estimate 2·(m+1) unknown coefficients, where (m+1) coefficients from model (8) are estimated first, and then, based on these estimates and the calculated values of the variable y, the other (m+1) coefficients from the model (9) are estimated. Both models (8) and (9) are linear in parameters, and the coefficients of these models can be easily found for sufficiently large m.

If we now substitute model (8) into equation (9) and expand the brackets, we can group the terms to obtain a structure and several terms that fully correspond to the structure and number of terms in KGp, thus deriving the desired polynomial model.

The system (8) – (9) can be represented in a more compact mathematical form:

$$\hat{y} = b_0 + \sum_{j=1}^{m} b_j (a_0 + \sum_{i=1}^{m} a_i x_i)^j \tag{10}$$

Models (8) – (9) or (10) do not represent KGp, but rather an approximate model of it. Indeed, it can be observed that a significantly smaller number of coefficients is estimated than would be necessary if constructing a KGp. For example, for m=3, the proposed method estimates 4+4=8 unknown coefficients, whereas the complete KGp, as indicated in (7), consists of 20 terms, requiring the estimation of 20 unknown coefficients. The smaller number of coefficients implies that model (10) provides approximate estimates of the KGp coefficients. Essentially, this means that a complete KGp is not constructed, but rather its simplified model, which has been proposed to be referred to as the “Elementary image of KGp” (Svetunkov, 2024).

The advantages of the elementary image of KGp over the KGp itself rest on the fact that the elementary image can be constructed for any number of variables, while the original KGp can only be constructed for a small number of variables. However, this does mean that the simplified representation of KGp captures nonlinearities less effectively than the original KGp. Nevertheless, previous studies have shown that, even though the elementary image of KGp is a simplified model, it possesses excellent approximation properties and describes various nonlinear relationships very well (Svetunkov, 2024).

Comparison of the Artificial Neuron Model and the Elementary image of KGp

If we compare the artificial neuron model (1) with the elementary KGp model (10), certain similarities can be observed. Firstly, both the neuron model and the polynomial model are unstructured statistical models. In regression analysis within mathematical statistics, it is assumed that the form of the relationship between input and output factors is explicitly represented by some function. It is presumed that the identified regression relationship is the model of the expected value of the modelled process, which evolves according to the properties of this model. The diversity of modelled processes significantly exceeds the forms of regression relationships used in mathematical statistics (Izonin et. al., 2024). Therefore, in econometrics, when selecting an econometric model, the researcher aims not to choose the best model but to select an acceptable model based on the principle: "Since I do not have other forms of functions for the econometric model, I will use this one as the best of a bad lot." Both the neuron model and the polynomial model do not require knowledge of the practical use since the form is automatically selected during the estimation of the neuron and polynomial coefficients, and the researcher is not familiar with this form. Therefore, both models are "blackbox" models, where the structure of the model is of no interest. The second similarity between the neuron and polynomial models will be even more apparent if we represent the artificial neuron model (1) as follows:

$$\begin{cases} y' = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_m x_m, \\ \hat{y} = f(y'), \end{cases} \tag{11}$$

and the model of the elementary image KGp can be represented as follows:

$$\begin{aligned} &\widehat{y'} = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_m x_m, \\ &\hat{y} = f(\widehat{y'}) \end{aligned} \tag{12}$$

It is evident that in both cases (11) and (12), the first equation in the system represents a multi-factor linear model of the same type, while the second equation in both (11) and (12) represents a transformation of the obtained result. Thus, the structural similarity between the artificial neuron model and the elementary KGp model is apparent.

However, in the artificial neuron model, the coefficients of the linear transformation and the transfer function are estimated simultaneously. Therefore, (11) is presented as a system of two equations. In contrast, in the elementary KGp model, the coefficients of the first equation are estimated first, and then the obtained results are substituted into the second equation. Thus, this is not a system of two equations but rather two sequential equations.

Aside from the obvious similarities between these two models, there are also several significant differences. The first obvious difference lies in the methods of estimating the coefficients of the neuron and the elementary KGp model. In the artificial neuron model (11), all its coefficients (for both the first and second equations) are estimated simultaneously, and due to the nonlinearity of the transfer function parameters, one of the numerical methods is used for this purpose. In the elementary KGp model (12), the coefficients are estimated in two stages: first, the coefficients of the first linear equation are estimated, and then the second polynomial equation is evaluated. Both equations are linear in their parameters, and coefficient estimation can be performed using straightforward statistical methods (Chen et. al., 2021).

Considering the difference in estimation methods, it should be noted that the process of estimating the coefficients in the elementary KGp model is significantly simpler and faster than estimating the coefficients of the artificial neuron model.

The second difference between them is that the artificial neuron is consistently nonlinear regardless of the number of input variables (signals), whereas the elementary KGp model increases its level of nonlinearity and its approximation power with the number of input variables. As shown in (Liu, Lu, Luo, 2020), artificial neurons with a sigmoid transfer function are well described by third-degree polynomials. This means that the elementary KGp model may describe nonlinear processes somewhat worse than the artificial neuron if the number of inputs $m < 3$. However, if the number of input factors $m > 3$, the elementary KGp model may be more accurate than the artificial neuron model. For $m = 3$, both models are expected to provide approximately the same level of approximation accuracy.

Example. We will use data on the Gross Domestic Product (GDP) of the United Kingdom (yt), gross capital accumulation (x1), and the economically active population of the country (x2) from 1990 to 2016. We aim to find the dependence of GDP on these two other factors. Based on this data, we will construct both an artificial neuron model and an elementary KGp model, first normalizing all data. We will minimize the sum of the squared deviations between the actual and calculated values.

As expected, the artificial neuron model approximated the original data slightly better. The mean squared error of approximation for the neuron model was 0.0189, while for the elementary KGp model, it was 0.0228.

Now, we will add a third factor (x3) to this data, namely, the expenditure on research and development in the UK for the same period. For the artificial neuron model, the mean squared error of approximation is 0.01753, and for the elementary KGp model, it is 0.01734. Apparently, the elementary KGp model provided a slightly more accurate description of the data compared to the artificial neuron model.

Of course, different results may be observed in various cases, but generally, it should be assumed that as the number of input factors increases in these two models, the elementary KGp model has a greater capability to describe nonlinearity compared to the neuron model.

Neural networks and polynomial networks

In practice, no one uses a mathematical neuron model as a standalone model for describing some nonlinear dependency. The true power of this modelling tool becomes apparent when connecting elementary neurons together, where the outputs from preceding neurons serve as inputs to subsequent neurons. Such an interconnected network of artificial neurons can describe very complex nonlinearities between input and output variables. Therefore, well-constructed neural networks prove to be so accurate in approximation that other known modelling methods cannot compete with them, including regression-correlation analysis methods.

Since we have just seen that the elementary KGp model can successfully compete with the neuron model, it is natural to assume that a network connecting such polynomials with each other might also serve as an alternative to neural networks.

Let us compare two networks of the same structure: a neural network and a polynomial network. But first, let's revisit the methods of estimating the coefficients of these networks. In neural network theory, this process is called “training,” and we will use this term along with “estimation.”

If a multilayer neural network is subjected to one of the numerical training methods (most commonly, the gradient method), all coefficients of the network are estimated simultaneously. In this process, the coefficients of the output layer play a major role — they are estimated more extensively, while the coefficients of preceding layers are either not trained at all or are poorly estimated. This happens because the error at the output is completely addressed during the training of the output layer. To mitigate this issue, Rumelhart, Hinton, and Williams proposed the backpropagation algorithm in 1986. This procedure involves “distributing” the training error across all estimated coefficients of the neural network and adjusting the gradient method so that the training of the coefficients in the last layer proceeds more slowly than the training of the coefficients in preceding layers. This is achieved using both the model's coefficient values and the gradient method parameters, which are adjusted depending on how far the coefficient is from the network output.

As the simple explanation of neural network construction shows, using them in applied research requires a good understanding of mathematics and programming skills, as training a neural network is an iterative process with many simultaneously estimated parameters. Often, in practice, researchers use standard template networks and software products with pre-embedded training procedures for neural networks of a given structure. In other words, researchers do not design the structure of the neural network for their specific tasks but use a ready-made template developed by someone else for different tasks. In such cases, the advantages of neural networks are not fully realized, and their application becomes less effective.

The complexity of training neural networks is the main drawback hindering their widespread practical application in modelling complex economic processes. In contrast, polynomial networks are straightforward to train. In polynomial networks, not all coefficients are estimated simultaneously but sequentially — from the coefficients of the input layer to those of the output layer. The results from one layer's estimation serve as the basis for estimating the subsequent layer. Once all the coefficients of the polynomial network have been estimated in stages, the training is complete. There is no need to re-estimate the model coefficients in search of a better solution — the best solution has already been found. This means that training a polynomial network does not require recurrent methods; in general, the well-known least squares method is quite sufficient for this task.

Thus, training neural networks requires significant time and the use of complex computational methods. In contrast, training polynomial networks takes negligible time and employs basic statistical methods. However, it should be clarified whether the simplicity of training polynomial networks leads to a loss of accuracy in modeling.

To address this question, let us build a two-layer neural network and an equivalent two-layer polynomial network using data on the UK's GDP (y) in relation to gross capital accumulation (x1), the size of the economically active population (x3), spending on research and development (x3), and the size of social benefits in the UK (x4).

We will test how these two networks perform in three cases using the given data:

1. From 1990 to 2020,
2. From 1990 to 2021,
3. From 1990 to 2022.

Since the influence of the first three factors on the GDP is approximately the same — they are factors that generate GDP — and the influence of social benefits (x4) is somewhat different — they consume funds from the state budget and social funds — we will construct the neural network and the polynomial network in the following manner:

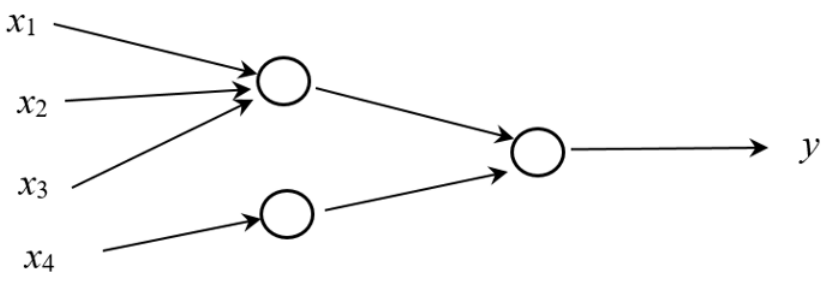

Fig. 1. Graphical model of a neural network and a polynomial network

For the neural network, each circle represents an artificial neuron, while for the polynomial network, each circle represents an elementary Kolmogorov-Gabor polynomial. A total of three neurons and three polynomials are used.

Since, after normalizing the initial data, it becomes both negative and positive, the transfer functions of the first two neurons are represented as logistic functions, while the transfer function of the last neuron is represented as a hyperbolic tangent function.

Alongside these two networks, we will also assess the accuracy of modelling this dependency separately with an artificial neuron model (using a hyperbolic tangent transfer function) and a Kolmogorov-Gabor polynomial model. The comparison results are presented in Table 1.

Table 1. Results of approximation of UK data by different models for specific periods of time

Model type	Neural network	Polynomial network	Artificial neuron model	Elementary image of the Kolmogorov-Gabor polynomial
1. 1990 – 2020				
Average sum of squares of approximation error	0.0077	0.0062	0.0310	0.0057
Standard deviation in %	11.14	10.02	15.47	9.60
Number of passes when evaluating coefficients	85 864	5	3 729	2
2. 1990 – 2021				
Average sum of squares of approximation error	0.0087	0.0075	0.0416	0.0068
Standard deviation in %	9.18	8.09	17.12	6.86
Number of passes when evaluating coefficients	34 252	5	15 011	2
3. 1990 – 2022				
Average sum of squares of approximation error	0.0216	0.0237	0.0745	0.0275
Standard deviation in %	17.31	18.12	32.15	19.55
Number of passes when evaluating coefficients	36 111	5	19 136	2

It is crucial to compare two models: the neural network model and the polynomial network model. The results show that in two out of three cases, the polynomial network was more accurate in approximation than the neural network. This does not necessarily mean that this ratio will hold in all cases; it only indicates that the polynomial network used in this example performed as well as the neural network. Both networks work with similar accuracy — their approximation standard errors differ by about one percent. However, the advantages of the polynomial network over the neural network become evident when considering the time required for training each network.

It is also worth noting that the elementary Kolmogorov-Gabor polynomial (the last column in Table 1) was the best model in terms of approximation accuracy in the first two cases and only 2.24% worse than the leader in the third case. This further confirms that the elementary Kolmogorov-Gabor polynomial can serve as an alternative to neural networks in modelling economic dynamics (Svetunkov, 2024).

Bayesian approach

Neural networks have a significant drawback. During the training process, the computational algorithm adjusts the network parameters to achieve the best output. However, it is impossible to understand how each neuron and each input factor influence the result based on the computed parameters. This limitation severely restricts the application of neural networks in modelling economic processes and, particularly, economic dynamics, as the meaning of the evaluated model parameters remains unclear.

Today, one of the actively developing approaches in economic modelling is the Bayesian approach. Generally, the Bayesian approach involves updating prior (a priori) conclusions based on new (posterior) data. In a narrower sense, Bayesian methods refer to statistical methods that use Bayes' theorem to compute conditional prior and posterior probabilities based on probabilistic distributions.

The Bayesian approach, both in its broad and narrow senses, is used by researchers in solving various econometric problems and economic forecasting tasks. One would expect the emergence of Bayesian methods applied to neural networks, but "neuro-Bayesian methods" currently exist only in theory. Applying the Bayesian approach to neural networks has proven to be impossible due to the neural network being a "blackbox" with an unknown structure to the researcher. Attempts to mathematically describe this network using multi-stage superposition of functions have been unsuccessful due to the complexity of the final mathematical model. Of course, the researcher knows the number of neurons, the connections between them, the number of "synapses," etc. But what is still impossible to trace is how the transformation of the input signal into the output signal occurs and how the factors and coefficients of the neural network influence the result.

Dynamic processes in economic systems can be classified into reversible and irreversible processes. Reversible processes are such that when returned to their initial condition, they will proceed in the same manner as before. In contrast, in economic systems where irreversible processes occur, there are not only quantitative but also qualitative changes. The latter means that the relationships between elements, the set of elements, and even the structure of the systems change — new elements emerge during evolution, and some old elements disappear. If a statistical model is constructed for such processes that adequately describes the past on average, then for it to be used as a forecasting model, its parameters need to be updated based on new posterior information for the model, i.e., the Bayesian approach must be employed.

Methods of model adaptation and adaptive models developed in forecasting serve as one of the tools for the Bayesian approach in its broad sense. Practice shows that their application significantly improves the accuracy of economic forecasts made using regression models. The widely known exponential smoothing method in short-term forecasting is one such method.

Unlike neural networks, in polynomial networks, the influence of each coefficient and factor on the result is known. This means that the Bayesian approach can be applied to polynomial networks, making them suitable for forecasting economic dynamics. To demonstrate this possibility, we will use one of the methods for adapting econometric models based on stochastic approximation.

We will use the same neural network and polynomial network models that were constructed using GDP data from the UK with four factors for the years 1990 to 2020. The subsequent years, 2021 and 2022, will be used as a validation period to test the suitability of the models for forecasting.

Polynomial network adaptation was performed on this data, while the neural network, as mentioned earlier, is not suitable for Bayesian evaluation and thus remained unchanged. In contrast, the adapted polynomial network, the original polynomial network, and the neural network were used. The results of forecasting the UK GDP using each of these three models for 2021 and 2022 are presented in Table 2.

Table 2. Comparative accuracy results, the UK GDP forecast by three models

Year	Factual data, y_t	Adapted polynomial network		Original polynomial network		Neural network	
		forecast	% error	forecast	% error	forecast	% error
2021	1.580226	1.39129	12.7	1.376194	13.80	1.375751	13.83
2022	2.001742	1.89455	5.5	1.559506	24.84	1.375753	37.07

As expected, the Bayesian approach to adapting the polynomial network improved its forecasting capabilities — the adapted polynomial network predicted the UK GDP more accurately than both the non-adapted original polynomial network and the neural network.

Conclusion

When modelling many economic processes, scientists and practitioners often face situations where none of the statistical models provide the necessary accuracy. In such cases, there is a desire to use some unstructured model like neural networks, but they are complex to apply, and examples of their successful use in economic practice are few.

The Kolmogorov-Gabor polynomial is a powerful tool for modelling nonlinearities, but as the number of influencing factors increases, the polynomial's structure becomes significantly more complex, which limits its applicability. Results of using KGp are limited to polynomials with a small number of explanatory variables — between three and five. The multi-stage procedure for constructing KGp proposed by A.G. Ivakhnenko in the 1970s turned out to be quite inefficient.

The method for constructing a simplified KGp model suggested in this research avoids these drawbacks — it is easy to construct, its parameters are easily estimated, and its size can be arbitrary. This method provides an approximate estimate of the actual KGp, which is why the model is referred to as the "elementary image of KGp."

Research has shown that the elementary image of KGp describes nonlinear economic processes well and can itself serve as an important tool for economic-mathematical modelling.

Structurally and functionally, the elementary image of KGp is very similar to an artificial neuron model. This means it can be used as a basis for developing another type of unstructured (sometimes referred to as non-parametric) models that describe complex nonlinear processes — polynomial networks.

Polynomial networks can have the same structure as neural networks. They can be single-layered or multi-layered, feedforward or recurrent — essentially, they can be like neural networks, but instead of artificial neuron models, they use elementary images of KGp.

It should be noted that specialists in neural networks have already used power polynomials instead of sigmoids in the transfer function of neural network models. These networks are reported to train faster and have very good approximate properties. However, it is not clear what degree these polynomials should be. Since there is no answer to this question, researchers use polynomials of small degrees or linear forms, referring to KGp. Moreover, unlike the polynomial networks discussed in this article, all coefficients of such neural networks with polynomial transfer functions are trained simultaneously, which implies a multi-iterative evaluation procedure. In the proposed polynomial networks, each elementary image of KGp is trained separately, which ensures quick and efficient estimation of all polynomial network coefficients.

As demonstrated with simple examples, training a simple two-layer feedforward neural network required several tens of thousands of passes, while a polynomial network of the same structure needed only a few simple iterations. The accuracy of describing economic nonlinearity is approximately the same for both neural networks and polynomial networks.

A significant advantage of polynomial networks is the possibility of applying the Bayesian approach — reassessing the parameters of the polynomial network based on new posterior data. This possibility was demonstrated through the adaptation of a polynomial network using the stochastic approximation method. The Bayesian approach to neural networks is currently impossible, and neuro-Bayesian methods are still in the stage of unsuccessful development.

Of course, the proposed polynomial networks based on the elementary image of KGp require additional and extensive research. However, it is already clear that they can serve as an alternative to neural networks in modelling complex economic processes.

REFERENCES

Anjorin A., Ricks M. 2023. Forecasting Single Family House Prices in the US using GMDH. SSRN Electronic Journal, 31. doi:10.2139/ssrn.4723624

Chen J., Yang S., Zhang D., Nanehkaran Y.A. 2021. A turning point prediction method of stock price based on RVFL-GMDH and chaotic time series analysis. Knowledge and Information Systems, 2693–2718. doi: 10.1007/s10115-021-01602-3

Gabor D., Wilby W. R., Woodcock R.A. 1961. A universal nonlinear filter, predictor and simulator which optimizes itself by a learning process. Proc. Inst. Electr. Engrs. 108 B (40), 85—98.

Hewamalage H., Bergmeir C., Bandara K. 2021. Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. International Journal of Forecasting 37, 388–427. doi:10.1016/j.ijforecast.2020.06.008

Izonin I., Muzyka R., Tkachenko R., Dronyuk I., Yemets K., Mitoulis S.A. 2024. A Method for Reducing Training Time of ML-Based Cascade Scheme for Large-Volume Data Analysis. Sensors 24(15), 4762. doi:10.3390/s24154762

Ivakhnenko A.G. 1963. Self-learning systems with positive feedback, 328.

Ivakhnenko A.G. 1971. Systems of heuristic self-organization in technical cybernetics, 372.

Ivakhnenko A.G. 1975. Long-term forecasting and control of complex systems, 312.

Kolassa S. 2020. Will deep and machine learning solve our forecasting problems? Foresight: The International Journal of Applied Forecasting, 57, 13-18.

Kolmogorov A.N. 1956. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. Reports of the USSR Academy of Sciences 108 (2), 179-182.

Leenaerts D., van Bokhoven W.M. 2013. Piecewise Linear Modeling and Analysis. Springer US, 198.

Liu Q., Lu X., Luo F., Zhou S., He J., Wang K. 2020. Secure BP from Homomorphic Encryption. Security and Communication Networks 1. doi:10.1155/2020/5328059

Liu W., Dou Z., Wang W., Liu Y., Zou H., Zhang B., Hou S. 2018. Short-Term Load Forecasting Based on Elastic Net Improved GMDH and Difference Degree Weighting Optimization. Appl. Sci. 8(9), 1603. doi:10.3390/app8091603

Marateb H., Norouzirad N., Tavakolian K., Aminorroaya F., Mohebbian M., Macanas M.A., Lafuente S.R, Sami R., Mansourian M. 2023. Predicting COVID-19 Hospital Stays with Kolmogorov–Gabor Polynomials: Charting the Future of Care. Information 14 (11), 590. doi:10.3390/info14110590

McElroy T., Ghosh D., Lahiri S. 2024. Quadratic Prediction of Time Series via Auto-Cumulants. Sankhya 86, 431–463. doi:10.1007/s13171-023-00326-6

Nelles O. 2020. Classical Polynomial Approaches. Nonlinear System Identification. doi:10.1007/978-3-030-47439-3_20

Razif N., Shabri A. 2023. Fourier Residual Modified Approach in Group Method of Data Handling for Electricity Load Forecasting. Advances in Intelligent Computing Techniques and Applications 210. doi:10.1007/978-3-031-59711-4_12

Svetunkov S. 1993. Econometric methods of demand forecasting,183.

Svetunkov S. 2024. Elementary image of the Kolmogorov-Gabor polynomial in economic modeling. Technoeconomics 3 (2), 4–21. doi:10.57809/2024.3.2.9.1

Terasvirta T., Kock A. 2010. Forecasting With Nonlinear Time Series Models. SSRN Electronic Journal, 34. doi:10.2139/ssrn.1531092

Volterra V. 1930. Theory of Functionals and of Integral and Integro-differential Equations, 226.

Wiener N. 1958. Nonlinear problems in random theory, 138.

Wim V. 2010. Signal Processing for Neuroscientists. Advanced Topics, Nonlinear Techniques and Multi-Channel Analysis, 61-90. doi:10.1016/B978-0-12-384915-1.00004-8.

СПИСОК ИСТОЧНИКОВ

Anjorin A., Ricks M. 2023. Forecasting Single Family House Prices in the US using GMDH. SSRN Electronic Journal, 31. doi:10.2139/ssrn.4723624

Chen J., Yang S., Zhang D., Nanehkaran Y.A. 2021. A turning point prediction method of stock price based on RVFL-GMDH and chaotic time series analysis. Knowledge and Information Systems, 2693–2718. doi: 10.1007/s10115-021-01602-3

Gabor D., Wilby W. R., Woodcock R.A. 1961. A universal nonlinear filter, predictor and simulator which optimizes itself by a learning process. Proc. Inst. Electr. Engrs. 108 B (40), 85—98.

Hewamalage H., Bergmeir C., Bandara K. 2021. Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. International Journal of Forecasting 37, 388–427. doi:10.1016/j.ijforecast.2020.06.008

Izonin I., Muzyka R., Tkachenko R., Dronyuk I., Yemets K., Mitoulis S.A. 2024. A Method for Reducing Training Time of ML-Based Cascade Scheme for Large-Volume Data Analysis. Sensors 24(15), 4762. doi:10.3390/s24154762

Ivakhnenko A.G. 1963. Self-learning systems with positive feedback, 328.

Ivakhnenko A.G. 1971. Systems of heuristic self-organization in technical cybernetics, 372.

Ivakhnenko A.G. 1975. Long-term forecasting and control of complex systems, 312.

Kolassa S. 2020. Will deep and machine learning solve our forecasting problems? Foresight: The International Journal of Applied Forecasting, 57, 13-18.

Kolmogorov A.N. 1956. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. Reports of the USSR Academy of Sciences 108 (2), 179-182.

Leenaerts D., van Bokhoven W.M. 2013. Piecewise Linear Modeling and Analysis. Springer US, 198.

Liu Q., Lu X., Luo F., Zhou S., He J., Wang K. 2020. Secure BP from Homomorphic Encryption. Security and Communication Networks 1. doi:10.1155/2020/5328059

Liu W., Dou Z., Wang W., Liu Y., Zou H., Zhang B., Hou S. 2018. Short-Term Load Forecasting Based on Elastic Net Improved GMDH and Difference Degree Weighting Optimization. Appl. Sci. 8(9), 1603. doi:10.3390/app8091603

Marateb H., Norouzirad N., Tavakolian K., Aminorroaya F., Mohebbian M., Macanas M.A., Lafuente S.R, Sami R., Mansourian M. 2023. Predicting COVID-19 Hospital Stays with Kolmogorov–Gabor Polynomials: Charting the Future of Care. Information 14 (11), 590. doi:10.3390/info14110590

McElroy T., Ghosh D., Lahiri S. 2024. Quadratic Prediction of Time Series via Auto-Cumulants. Sankhya 86, 431–463. doi:10.1007/s13171-023-00326-6

Nelles O. 2020. Classical Polynomial Approaches. Nonlinear System Identification. doi:10.1007/978-3-030-47439-3_20

Razif N., Shabri A. 2023. Fourier Residual Modified Approach in Group Method of Data Handling for Electricity Load Forecasting. Advances in Intelligent Computing Techniques and Applications 210. doi:10.1007/978-3-031-59711-4_12

Svetunkov S. 1993. Econometric methods of demand forecasting,183.

Svetunkov S. 2024. Elementary image of the Kolmogorov-Gabor polynomial in economic modeling. Technoeconomics 3 (2), 4–21. doi:10.57809/2024.3.2.9.1

Terasvirta T., Kock A. 2010. Forecasting With Nonlinear Time Series Models. SSRN Electronic Journal, 34. doi:10.2139/ssrn.1531092

Volterra V. 1930. Theory of Functionals and of Integral and Integro-differential Equations, 226.

Wiener N. 1958. Nonlinear problems in random theory, 138.
Wim V. 2010. Signal Processing for Neuroscientists. Advanced Topics, Nonlinear Techniques and Multi-Channel Analysis, 61-90. doi:10.1016/B978-0-12-384915-1.00004-8.

INFORMATION ABOUT AUTHOR / ИНФОРМАЦИЯ ОБ АВТОРЕ

SVETUNKOV Sergey G. – Professor, Doctor of Economic Sciences
E-mail: sergey@svetunkov.ru
СВЕТУНЬКОВ Сергей Геннадьевич – профессор, д.э.н.
E-mail: sergey@svetunkov.ru
ORCID: https://orcid.org/0000-0001-6251-7644

Статья поступила в редакцию 15.08.2024; одобрена после рецензирования 21.08.2024; принята к публикации 24.08.2024.
The article was submitted 15.08.2024; approved after reviewing 21.08.2024; accepted for publication 24.08.2024.