Functional Network Construction and Approximation Algorithm for Polynomial Functions

Dapeng Zhou

School of Mathematics and Quantitative Economics, Shandong University of Finance and Economics, Jinan, 250014, China.

Keywords: Polynomial function; Functional network; Approximation algorithm; Network construction

Abstract: As for the optimization of polynomial function algorithm, artificial neural network is an important method to solve the problem of function approximation. However, the traditional learning neural network has some defects, such as being very sensitive to the initial weight and easy to converge to the local minimum; Slow convergence or even no convergence; Over fitting and over training; The number of hidden nodes in the network is uncertain. To solve the above problems, a three-layer functional network and approximation algorithm of polynomial function are proposed, and how to determine the number of calculation units in the middle hidden layer is given. The proposed algorithm can approximate polynomial function with arbitrary accuracy, and has fast convergence speed and good performance, which overcomes the shortcomings of artificial neural network. The example analysis shows that the algorithm is very effective, fast convergence speed and high calculation accuracy. The proposed polynomial functional regression functional network model and learning algorithm have important guiding significance for the research of computer algebra.

1. Introduction

The functional network deals with general functional models. Its connections between neurons have no weights, and the neuron functions are not fixed, but can be learned. It is often a combination of a given function cluster. specific problems to select different function clusters [1]. Functional networks have been applied to nonlinear system identification, stochastic analysis, computer algebra and other related fields [2]. This method shows good performance in solving the above problems.

Although functional networks have been successfully applied in many aspects, as functional networks are a new extension of neural networks proposed in recent years, some theoretical and application foundations are not perfect, so it is necessary to constantly put forward new network structures that are more suitable for the problems to be solved, improve the basic theories and propose new approximation algorithms [3]. Based on these ideas, the focus of this paper is to theoretically study the function approximation performance of functional networks. Function approximation is an important means to evaluate the computing power of functional networks. An important basic problem in the application of functional networks is how to design the structure of a functional network and have the ability to learn a given function [4]. Recently, more work has focused on a special model of functional networks called separable functional networks, which theoretically reflects the relationship between network approximation performance and the structure of intermediate computing units [5]. However, according to the general model structure of functional networks, there is no specific algorithm to achieve functional approximation in theory, so it is difficult to apply it to practical problems [6]. Therefore, the application range of functional networks is limited. Polynomial function is a relatively simple and easy-to-calculate function, and it is also one of the most widely used functions in practical problems. It also plays an important role in expressing general functions, constructing splines and fitting surfaces. Through the research and discussion of polynomial functions, we can further study the approximation of general functions, thus revealing the essence and the mechanism of functional networks to approximate general

  DOI: 10.25236/icmmct.2022.052

functions [7]. In this paper, the approximation problem of three-layer functional networks with polynomial function as the objective function is studied.

The function approximation problem can be described as: how to determine a function set Φ, so that any function f on a specific space A can be approximated by the functions in Φ. The basic problems are: (1) How to choose Φ; (2) How to base on The function in Φ constructs an approximation function so that it has as little deviation as possible from the objective function. Function approximation technology has a wide range of applications in system identification, optimal control and other fields. Therefore, it is of great practical significance to explore the theory and technology of function approximation. The research on function approximation theory has produced many mature function approximation methods, such as Algebraic polynomial approximation, trigonometric polynomial approximation, interpolation method approximation, spline function approximation, etc [8]. The classic method is usually the regression analysis method, that is, it is known to use the sample data set $\{(x_i, y_i) | x_i \in R^n, y_i \in R, i \in N\}$ to construct a function that satisfies the constraints such as sampling data point error, but there are other related but different problems, such as control and other fields, it is necessary to solve specific problems. model, and whether the model is appropriate can only be judged according to its suitability when the model is executed in the simulation. Regression analysis is not suitable for solving such problems, and the fundamental reason is that the source of sample data used is often limited by the domain knowledge of experts, which leads to deviations between the specific model solved and the actual model, or even wrong results. The development of machine learning methods provides a new idea to solve this problem. They can be used not only for data regression, but also for adaptive modeling, which expands the application field of approximation technology.

2. Functional network

2.1. Functional neuron

Neuron is a kind of mathematical model which is abstracted, generalized and simplified by referring to the main functions of neurons in the real nervous system [9]. However, it is different from the general study of abstract mathematics [10]. It has a clear purpose, and its purpose has two different aspects that are both related and different from each other. On the one hand, the purpose is to study the mechanism and function of the real nervous system, which is the long-term goal of human beings. The progress of this research provides a source for the research of artificial intelligence. Therefore, neurons and neural network models can't be arbitrary, but must be restricted by disciplines such as neurology, physiology and psychology. On the other hand, the study of artificial neural network is based on the real nervous system, but it is not restricted by "authenticity". Just as humans and animals move by walking and crawling, manual movement is more effective by rolling. Therefore, there is a certain degree of freedom in establishing the mathematical model of neurons. However, it must be restricted, have a clear purpose, and have actual or potential application value. Leaving this point is just a pure mathematical game.

Studies by neuroscientists show that the biological nervous system is composed of numerous neurons, which include cell body, axons, dendrites and synapses. When synapses receive chemical information from other axons, they stimulate or simulate the corresponding dendrites and input signals to the cell body. The adjustment of the connection weight in the neuron model is modeled after the setting of this biological mechanism. Cell body is the core of neurons, and its function is to process and transform the signals from dendrites, and then output signals from axons. The function of neural network model is realized by activation function $f(\cdot)$. The activation function of traditional neuron model adopts Sigmoid, RBF, Gaussian function, etc. Once activation function $f(\cdot)$ is selected, it will not change. In fact, this assumption is not reasonable. We know that in daily life, some people understand problems quickly while studying, while others don't. This shows that the processing function of the same type of neuron cell body to external signals varies from person to person.

2.2. Construction of functional network

The functional network is composed of the following elements: ① input unit layer: This is a layer of unit of input data, and the input unit is represented by a solid circle with corresponding name; ② Output unit layer: This is the last layer unit, which outputs the result data of the network. The output unit is also represented by a solid circle with corresponding name; ③ One or more layers of neurons or computing units: each neuron is a computing unit, which calculates a set of input values from the previous layer of neurons or input units, and provides a set of input data to the next layer of neurons or output units. The calculation units are interconnected, and the output of each neuron can be used as a part of the data of another neuron or output unit. Once the input value is given, the output value is determined by the type of neuron, which is defined by a function. For example, suppose a neuron has s inputs $(x_1, x_2, \cdots, x_s)$ and k functions $F_j, j = 1,2,\cdots,k$, so that the $y_j = F_j(x_1, x_2, \cdots, x_s); j = 1,2,\cdots,k$ function F_j is determined by the network structure, and the neuron is represented by a circle with the corresponding F_j function name; ④ Directed connecting lines: they connect the input layer, the first layer neuron, the second layer neuron, the last layer neuron and the output unit. The arrow indicates the direction of information flow. All these together form a network structure, which determines the functional ability of the network. The structure of a network refers to the organization of neurons and the connections they contain. Figure 1 and Figure 2 show the structure of a neural network and functional network respectively.

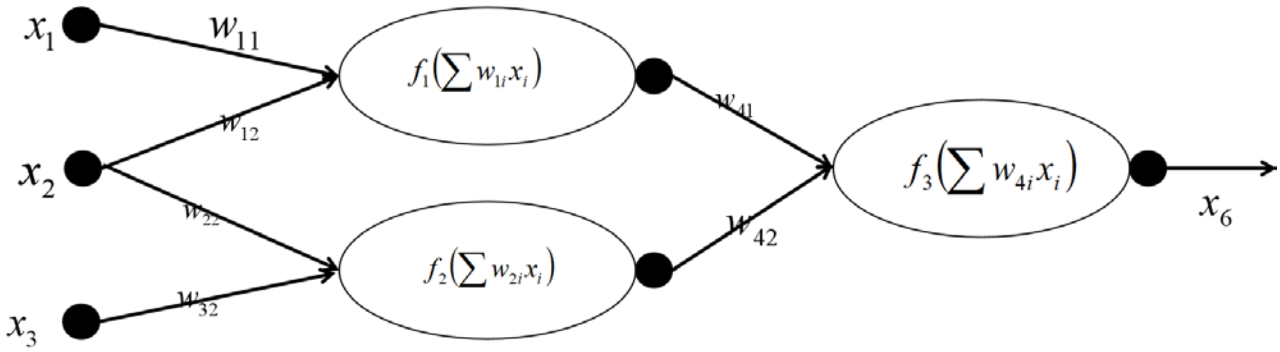

Figure 1 Neural network structure

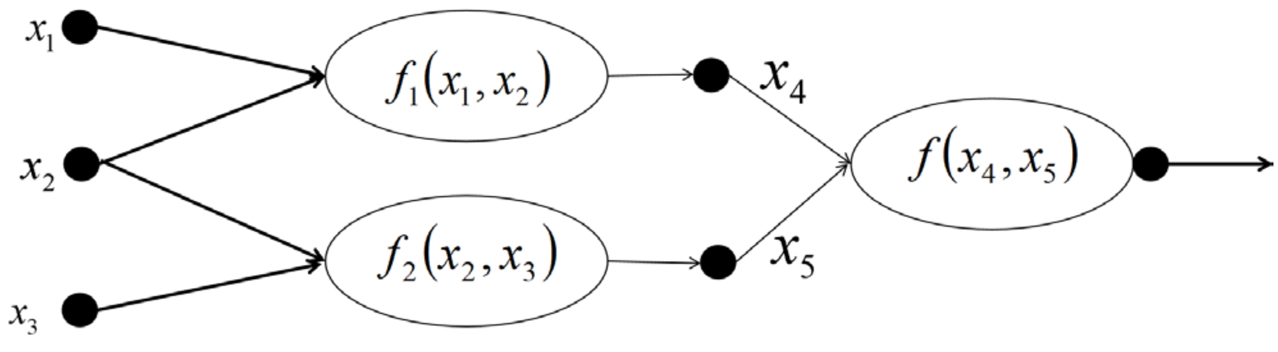

Figure 2 Functional network structure

The most obvious difference between the standard neural network and the functional network is that in the standard neural network system, the activation function of neurons is given, and the activated connection weights are trained. In the functional network system, there is no weight. The functional network trains the function F, that is, trains the activation function. In a standard neural network system, the output unit is the output of a single neuron (the last unit), while in a functional network system, the output unit can output one or more neurons, which implies a compatibility, that is, the more neurons connected to the output unit, the less the degree of freedom of the output neurons.

3. Functional network based on function approximation

3.1. Functional Network Approximation Algorithm

Functional networks, like neural networks, also have various structures. It is impossible to describe all functional networks with a unified general structure, and it is impossible to represent all functional networks with a unified function. In the application research of functional networks, separable functional networks are the most widely used. Its functional expression is the combination of the separation effect of input variables Compared with the current neural network approximation method, the advantage of function approximation based on functional network is that the calculation principle and mechanism are different. The use of functional network with parallel structure can make the solution of the problem trained once and obtained in parallel. When solving practical problems, the traditional least squares fitting method often takes a certain randomness in the sampling data. It is closely related to intuitive experience and the physical background of the problem. It selects the appropriate basis function class based on experience and mathematical knowledge. The functional parameters and basis function classes in functional networks are completed by learning. Functional networks have the characteristics of self adaptation and self-learning. If the data and basis function from the problem change, it can be seen from the above example that only the basis function needs to be adjusted slightly, and there is no need to start from scratch. The traditional fitting method has the disadvantages that if the sampling data is insufficient, the error of the objective function will be too large; If there is too much sampling data, there will be over matching. Functional networks are adaptive, which can be avoided if combined with the empirical knowledge of practical problems. If the basis function is selected properly and reasonably, the approximation effect will be better.

In the sense of functional network approximation, the problem of function approximation can be described as: how to determine a set of basis functions Φ through learning, so that any function f in a specific space A can be approximately constructed from the functions in Φ. The basic problems are: ① How to choose the basis function set Φ; ② How to construct an approximate function based on the functions in the basis function set Φ, so that its error with the objective function is as small as possible. This can be done by functional network learning to measure the error between the approximate function and the objective function, so it has the following definition.

1: definition: square root error

$$RMSE = \sqrt{\sum_{p=1}^{r} \left\| b_p - \hat{b}_p \right\|^2 / r} \tag{1}$$

Where: $\hat{b}_p$ refers to the output of the functional network and $\|\cdot\|$ refers to the normal form of the function.

For the learning algorithm with teacher training, a set of learning sample data is given

$$\{((x1i, x2i), x0i) | i = 1,2,\cdots,n\} \tag{2}$$

The error cost function can be defined as:

$$Ei = x0i - \sum_{i=1}^{r} \sum_{r+1}^{n} Cij\,\phi i(x1i)\varphi_j(x2i)$$
$$i = 1,2,\cdots,n \tag{3}$$

In order to find the optimal network parameters, it is necessary to minimize the sum of squares of errors:

$$\frac{\partial E}{\partial cpq} = 2\sum_{k=1}^{n} Ek\phi p(x1k)\varphi_q(x2k) \equiv 0$$
$$p = 1,\cdots,r; q = r+1,\cdots,n \tag{4}$$

By solving the linear equations, the optimal network parameters Cpq can be obtained.

3.2. Polynomial functional regression functional network

$$E = \sum_{k=1}^{n} E_k^2 = \sum_{k=1}^{n}\left[x_{0k} - \sum_{i=1}^{r}\sum_{r+1}^{n} Cij\phi i(x_{1k})\varphi_j(x_{2k})\right]^2 \tag{5}$$

Derivation of formula (5), there are

Based on the characteristics of topological structure of regression functional network, if the function of functional neurons in the middle layer is set as an adjustable polynomial function sequence, a new model of regression functional network with polynomial function is proposed. It not only inherits the structural characteristics of regression functional network RFN, but also has strong recursive calculation ability and function approximation ability. Aiming at the ubiquitous recursive calculation problem, a learning algorithm based on polynomial functional regression functional network is designed. This learning algorithm is applied to the approximate decomposition of multivariate polynomials, which is a classical but not completely solved problem. The methods and ideas of this paper are expanded and extended to other related recursive calculation problems in other disciplines. Its purpose is to provide a new calculation model and algorithm for approximate calculation of computer algebra and related fields. Compared with the neural network calculation model in this paper, an example shows that the model structure and learning algorithm of polynomial functional regression functional network PFRFN proposed in this chapter are intuitive, clear, fast and stable.

Polynomial regression functional network PFRNN is still a functional functional network with both feed-forward and feedback paths. It is different from general RFN in that the output of its intermediate unit L layer mapped by a set of functional neurons whose neuron functions are adjustable polynomial functions is recursively sent to other functional neurons or itself after one or several steps. The function of feedback path enables functional networks to effectively handle the context-related computation of recursive sequences. In general, each output of L layer functional neuron of RFN represents the occurrence probability of a recognition unit. When necessary, all this information is transmitted to the subsequent frame signal. It only takes into account the characteristics of continuous speech recognition, rather than a specific functional neuron function. Dealing with practical problems is often limited by the domain knowledge of experts, which limits the application scope of RFN network to a great extent. Therefore, compared with RFN network, PFRFN network proposed in this paper has better advantages of self-organization, self-adaptability and self-learning, and its functional neuron function is adjustable, which makes the network converge quickly, calculate with high accuracy and have good stability.

4. Conclusions

The one-input one-output intermediate-layer computing unit proposed in this paper is a three-layer functional network determined by the number of times of the approximation function. The difference between this approximation algorithm and the neural network is that it does not require an activation function, nor does it require training parameters. It is simple and practical , which is easy to calculate. It can be seen from the example that it is easier to achieve this result. The obtained results show that: when the polynomial order is given, the number of computing units in the middle layer is the order plus one, and the given function can be approximated to a predetermined precision arbitrarily. It is of great significance to be further applied to function approximation.

References

[1] Moothedath S, Chaporkar P, Belur M. A Flow-Network-Based Polynomial-Time Approximation Algorithm for the Minimum Constrained Input Structural Controllability Problem[J]. IEEE Transactions on Automatic Control, 2018, 63(9):3151-3158.

[2] Lin T , Zhou Z . Robust network function virtualization[J]. Networks, 2020, 75(4):438-462.

[3] Kamiyama N . A Note on Submodular Function Minimization with Covering Type Linear Constraints[J]. Algorithmica, 2018, 80(10):2957-2971.

[4] Bhabak P , Harutyunyan H A . Approximation Algorithm for the Broadcast Time in k -Path Graph[J]. Journal of Interconnection Networks, 2019, 19(4):1950006.

[5] V Patel, G Regts. Deterministic Polynomial-Time Approximation Algorithms for Partition Functions and Graph Polynomials[J]. SIAM Journal on Computing, 2017, 46(6):1893-1919.

[6] Rishikesh K , Pramod R . Phase unwrapping algorithm using polynomial phase approximation and linear Kalman filter[J]. Appl Opt, 2018, 57(4):702-708.

[7] Ottaviano, Stefania, De, et al. Optimal curing policy for epidemic spreading over a community network with heterogeneous population[J]. Journal of Complex Networks, 2018, 6(5):800–829.

[8] Wu Y , Zhao X . Adaptive Backstepping Control Based on Functional Link Radial Basis Function Neural Network for PMLSM[J]. Diangong Jishu Xuebao/Transactions of China Electrotechnical Society, 2018, 33(17):4044-4051.

[9] R Bar-Yehuda, Beder M , D Rawitz. A Constant Factor Approximation Algorithm for the Storage Allocation Problem[J]. Algorithmica, 2017, 77(4):1105-1127.

[10] Runovskii, K V. Trigonometric polynomial approximation, K-functionals and generalized moduli of smoothness[J]. Sbornik Mathematics, 2017, 208(2):70–87.