OCTANE NUMBER PREDICTION FOR GASOLINE BLENDS USING CONVOLUTION NEURAL NETWORK

ZHU YUE

DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

OCTANE NUMBER PREDICTION FOR GASOLINE BLENDS USING CONVOLUTION NEURAL NETWORK

ZHU YUE

RESEARCH PROJECT SUBMITTED IN THE FACULTY OF ENGINEERING UNIVERSITY OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF INDUSTRIAL ELECTRONIC AND CONTROL ENGINEERING

DEPARTMENT OF ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Zhu Yue

Matric No: 17220978

Name of Degree: Master of Industrial Electronics and Control Engineering

Title of Project ("this Work"): OCTANE NUMBER PREDICTION FOR GASOLINE BLENDS USING CONVOLUTION NEURAL NETWORK

Field of Study: Artificial Intelligence

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

ABSTRACT

With the development of information technology, the development of neural network plays an important role in the prediction of various situations in real life. At present, there are many prediction algorithms based on machine learning. According to the "80/20 rule" for building machine learning models, 80% of the time is spent on finding, cleaning, and organizing data, while the remaining 20% for training of the machine learning model. Machine performance learning models depend to a large extent on the data quality used to train the model. Therefore, data preprocessing is widely considered to be one of the most critical stage in the whole process. In this project three commonly used algorithms are used for the prediction of octane number for gasoline blends, which describes the behavior of the fuel in the engine at lower temperatures and speeds, and is an attempt to simulate acceleration behavior. These three algorithms are back propagation (BP), radial basis function (RBF) and Extreme learning machine (ELM) algorithm. Simulation study on performance of these algorithms have been carried out with the available databases. Simulation results show that RBF algorithm gives the best performance.

Key words: Machine Learning; Prediction Method; Convolution Neural Network (CNN); Octane Number Prediction;

ABSTRAK

Dengan perkembangan teknologi maklumat, pengembangan rangkaian neural memainkan peranan penting dalam meramalkan pelbagai situasi dalam kehidupan nyata. Pada masa ini, terdapat banyak algoritma ramalan berdasarkan pembelajaran mesin. Menurut "peraturan 80/20" untuk membangun model pembelajaran mesin, 80% masa dihabiskan untuk mencari, membersihkan, dan mengatur data, sementara 20% sisanya untuk latihan model pembelajaran mesin. Prestasi model pembelajaran mesin banyak bergantung pada kualiti data yang digunakan untuk melatih model. Oleh itu, pemprosesan data secara meluas dianggap sebagai salah satu peringkat yang paling kritikal dalam keseluruhan proses. Dalam projek ini tiga algoritma yang paling biasa digunakan telah digunakan untuk meramalkan bilangan oktan untuk campuran petrol, yang menggambarkan tingkah laku bahan bakar dalam mesin pada suhu dan kelajuan yang lebih rendah, dan merupakan usaha untuk mensimulasikan perilaku pecutan. Tiga algoritma tersebut adalah penyebaran semula (BP, fungsi asas radial (RBF) dan algoritma pembelajaran mesin terlampau (ELM). Kajian simulasi mengenai prestasi algoritma ini telah dilakukan dengan menggunakan pangkalan data yang sedia ada. Hasil simulasi menunjukkan bahawa rangkaian neural RBF memberi keputusan yang paling memuaskan.

Kata-kata kunci: Pembelajaran Mesin; Kaedah Ramalan; Rangkaian Neural Konvolusi (CNN); Ramalan Nombor Oktana;

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Assoc. Prof. Ir. Dr. Chow Chee Onn for his guidance in completing this project. From time to time, he offered thoughtful advice and guidance to get the job done. Without his guidance, professional knowledge and patient attitude, this research project would not have been possible.

Next, I would like to thank my fellow course fellows for their encouragement, which proved to be the driving force for me to complete this research. I was pleased with their input and methodology, especially in using Google to look up data and build neural network models for machine learning.

I also want to thank my family for their support throughout the year. Their gentle treatment to me in life, their generous financial support and their encouragement when I meet difficulties all give me the motivation to move forward.

Last but not least, I would like to give some credit to the authors of all the journals and projects that I cannot mention here. I could not make this research result without the knowledge of the predecessors. The scientific research achievements they have made time and again have made me have a deeper understanding of many problems in the field of machine learning and their work is very important to help me better understand this project.

TABLE OF CONTENTS

ABBREVIATIONS

IBM	:	International Business Machines Corporation
IOT	:	Internet of Things
IT	:	Information Technology
AI	:	Artificial Intelligence
ML	:	Machine Learning
DL	:	Deep Learning
BP	:	back propagation
RBF	:	Radial Basis Function
ELM	:	Extreme Learning Machine

CHAPTER 1: INTRODUCTION

1.1 Background

Petroleum and its products are complex mixtures. Accurate analysis of these complex mixtures has always been a challenging task in the petroleum industry. In recent years, spectral analysis has become a new method for the identification of complex components and has attracted widespread attention.

Near infrared spectroscopy (NIR) is widely used in the component prediction of petroleum and its products. In terms of gasoline octane number prediction, some literatures have accurately predicted gasoline octane number by using near-infrared spectroscopy and multiple linear regression model based on 384 gasoline samples (Quinlan, J.R.1993). The traditional octane number measurement methods of gasoline including the research octane number (RON) and the motor octane number (MON). However, these methods may not suitable for production control and online testing due to high cost and long computational delay.

In this project, the octane number of gasoline is predicted using neural network algorithm (Bradley,P. 1997). This method is chosen because of its accurate predication and some other important features. First, the neural network is suitable to be used for characterization of complex systems. Second, neural network has strong self-adaptive ability. Third, the neural network algorithm uses offline learning, so it can be said that online prediction can be done in real-time manner with mall processing delay (Quinlan, J.R.1993). Last but not least, the selection of predictors of the neural network is flexible, and any data that is considered to be related to the prediction can be included as the input vector.

1.2 Problem Statement

Forecasting involves all aspects of people's work and life. People have noticed the problem of prediction since ancient times. For many centuries, many scholars have put forward many theories and models on prediction. A good prediction method can help people to make a more accurate understanding on the development trend of future events and provide guidance for future behaviors . With the progress of social science and technology, people are facing more and more problems related to prediction (Blum, A., L, Langley, & P. 1997). In this project we tackle one of the major prediction problem in the oil and gas industry, which is involved in the prediction of research octane number (RON) in gasoline blends.

1.3 Motivation

Forecast is people's expectation, estimation, analysis, judgment and speculation of the future development of objective things. Forecasting is based on the assumption that patterns drawn from past experience or data will repeat in the future.

As the saying goes, "to be prepared for a rainy day", a decision maker must make a major decision based on as accurate a prediction as possible of future events, otherwise it will be a random, aimless, clueless action. Forecasting is very important in many aspects, such as organization, management and decision making, because the grasp and evaluation of future events affect the decision-making process. The fields covered include politics, economy, history, ecology, culture, education, people's psychology, social morality and so on (Wang Fei, Yang Shengtian, Ding Jianli, Wei Yang, Ge Xiangyu, & Liang Jing. 2018). In this project, The reason for research Octane Number is obvious. Octane number plays a very important role in the composition of gasoline, and it can be used to evaluate the quality of gasoline. Good quality gasoline has very little damage to the internal machinery of cars, which is very helpful in our choice of gasoline.

1.4 Objectives

The main aim of this project is to study the ability of machine learning methods in predicting the octane number in gasoline blends. With this in mind, the following objectives are formulated:

1. To model a prediction method using machine learning algorithm to determine the octane number in gasoline blends.
2. To compare the performance of various machine learning models.

1.5 Organization of the Thesis

The thesis has five chapters. In the first chapter, a brief introduction to this research project is made. The second chapter reviews the predictive methods of neural networks using machine learning. Chapter three describes the methodology, emphasizing on the arithmetic used as well as the prediction methods, data sets, and performance indicators. Chapter four presents the simulation results and a comprehensive discussion is also given. Finally, conclusion is given in chapter 5.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview of Machine Learning

Machine learning is a data analysis method theory, and it has been involved in many fields, such as probability and statistics theory (Chen K. 2018), which is widely used. On the other hand, approximation theory is also a very representative theory. Machine learning is the key technology core of artificial intelligence. To be specific, machine learning is not a new theory, it has been widely used decades ago by predecessors, such as Bayes and others.

Research on machine learning has made great progress. At the beginning, machine learning studies the executive ability of the system, (Wang Fei, Yang Shengtian, Ding Jianli, Wei Yang, Ge Xiangyu, & Liang Jing. 2018). mainly by changing the execution environment of the machine and changing the corresponding performance parameters to detect the data fed back by the system, so as to study the executive ability of the machine, so that the system will choose an optimal environment to execute the task.

Later, researchers implanted the knowledge of various fields into the system, and used the knowledge of graph structure and its logical structure to describe the system, (Chen K. 2018) . so that the machine could simulate human learning. This has had some success, but the system cannot learn more in-depth knowledge (Wang Fei, Yang Shengtian, Ding Jianli, Wei Yang, Ge Xiangyu, & Liang Jing. 2018). so it has great limitations. The researchers then learned the concepts individually. Extend to multiple learning, that is, develop from single learning to integrated learning, and integrated learning with practical application. There are many representative works of famous scholars in this period, such as Mostow's teaching and learning, Langley's Bacon Plan and its improvement plan, etc (Li Helong, Yu Haibo, Wang Chunyu, Liu Jia, Wang Xingyuan, & Yuan Jinshuai et al. 2020).

2.1.1 Artificial Neural Network

Artificial Neural Network (ANN) is a new information processing system. Its mechanism can be understood as the development of human brain structure and activity mechanism. It's a pattern similar to how our neurons and nerves propagate. What we call the artificial neural network model, it is a cornerstone for most scholars to understand the human brain neural network, and then use physical and mathematical verification methods to abstract processing, establish the model in the direction of information processing. (Wang Fei, Yang Shengtian, Ding Jianli, Wei Yang, Ge Xiangyu, & Liang Jing. 2018). it can simulate the artificial neural network.Many of the characteristics of the human brain reflect some of the characteristics of the human mind. The characteristics of the organizational structure include parallel processing, distributed storage, and fault tolerance. So called artificial neural networks. There are many combinations of connections, simple or complex connections. The element features massively parallel processing, resulting in a highly developed nonlinear system that can run in parallel with its peers. The parallelism in structure makes the information storage of neural network inevitably adopt the distributed mode, that is, the information is not stored in a part of the network, but distributed in all the connection rights of the network (Wang Fei, Yang Shengtian, Ding Jianli, Wei Yang, Ge Xiangyu, & Liang Jing. 2018). The inherent parallelism and distribution of neural networks make it spatially distributed and temporally parallel in information storage and processing. These two characteristics make the neural network have good fault-tolerant ability. When some neurons in the network are damaged, the overall performance of the system will not be affected.

On the other hand, when the input of fuzzy, incomplete or deformed information, the neural network can restore the complete memory through association, so as to realize the correct identification of incomplete input information. Ability characteristics: self -

learning, self - organization and self - adaptability. We all say that a neural network is adaptive, and what we mean by adaptive is that it changes its performance in the face of an unexpected situation (He Qing, Li Ning, Luo Wenjuan, & Shi Zhongzhi. 2014), or a changed environment, and what a neural network learns by itself, is when some external environment changes. After a long time or a period of time. With training and self-regulation, he was able to readjust the structure and parameters of the network structure (Wang Fei, Yang Shengtian, Ding Jianli, Wei Yang, Ge Xiangyu, & Liang Jing. 2018). so that the desired output can be generated for the given input. Under the external stimulation, the nervous system adjusts the synaptic connections between neurons according to certain rules, and gradually constructs the neural network, which is the self-organization of the network. The basic functions of neural network Neural network has associative memory, nonlinear mapping, classification and recognition,

Optimization calculation, and the characteristics of the knowledge processing five types of intelligence, the focus is on the first two associative memory function, to neural network can store information and adaptive learning mechanism in advance training (Chen K. 2018), never complete information and noise interference to recover the original nonlinear mapping function, complete information refers to the neural network can through to the system input and output sample of automatic extraction of learning. Mapping rules to fit arbitrarily complex nonlinear functions with arbitrary accuracy. (He Qing, Li Ning, Luo Wenjuan, & Shi Zhongzhi. 2014).

Handling The peak period began with the first international academic conference on artificial neural network in 2000, and the research and application of artificial neural network was rapidly upsurge all over the world. Artificial neural network is divided into feed-forward network and feedback network according to network structure. Feed-forward networks are fed from input. A network structure that processes information

layer by layer from each hidden layer to the output layer(He Qing, Li Ning, Luo Wenjuan, & Shi Zhongzhi. 2014). Feedback network computing unit can receive input from the outside world, and it can output to the outside world.

The determination of neural network framework is generally divided into this steps:

1) Design of neural network structure

2) Determination of the number of neurons

3) Determination of neural network levels, including the determination of the number of nodes in the input layer, hidden layer and output layer

4) Connection of network units

5) Determination of energy function and closed value of neural network.

The explosive increase of data in modern society is a feature of the era of big data. It has various types of data, which will be constantly analyzed and analyzed through machine learning neural network, (He Qing, Li Ning, Luo Wenjuan, & Shi Zhongzhi. 2014), such as image semantic network data, etc. which is in the environment we live in now. With the advent of the era of big data, machine learning efficiency is very high and has become the main driving force.Its practical applications are mainly divided into the following:

(1) Machine learning based on learning strategies

There are mainly two kinds of machine learning:Deductive learning, analogy learning, etc (Chen K. 2018). There is also traditional machine learning based on numerical analysis to choose the appropriate mathematical model and parameters. Use

the appropriate learning algorithm to train the input sample data, and then analyze and predict the trained model.

(2) Machine learning based on learning styles

Machine learning based on learning mode includes supervised learning, unsupervised learning, reinforcement learning and so on. Supervised learning refers to input. There are supervisory signals in the data, and the function or neural network is used as the basis function model for iterative calculation, (Quinlan, J.R.1993) and finally a function junction is obtained.Unsupervised learning means that there is no supervised signal in the input data. It is a classification learning method, and the results obtained are categories.Reinforcement learning is a learning method which is carried out by means of statistics or dynamic programming with reward and punishment signals as input.

(3) Data based machine learning

There are two kinds of machine learning based on data form: structured learning and unstructured learning. Structured learning is a process in which learning models are obtained through numerical calculation or symbolic deduction based on structured data.Such as neural network learning, statistical learning, decision tree learning, rule learning and so on.Unstructured learning refers to taking unstructured data as input (Chen K. 2018) such as explanatory learning and text mining. Image mining, Web mining and so on.

In addition, the application scenarios of machine learning are very wide, and each science and technology field can join machine learning methods to implement. which are common in the following fields of science and technology:

(1) Data analysis and mining

Data analysis and data mining refer to the data processing technology that collects data through various devices or sensors and makes analysis and judgment on the collected data. Machine learning itself needs a large amount of data to do the support, through the study of data samples to get a model (Blum, A., L. , Langley, & P. 1997), so that the new data can be quickly processed. Data analysis and mining integrate machine learning algorithms and data access, use machine learning to conduct rapid statistics, analysis and discovery of massive data, and at the same time use data access technology to realize efficient reading and writing of data. Machine learning has endowed computers with powerful capabilities in big data processing, making them irreplaceable in the field of data analysis and mining.

(2) Pattern recognition

Model is the core knowledge system extracted from production and living experience. Pattern recognition is the representation of things or phenomena. It is very useful for what we know as pattern recognition, which is usually used for image speech. The analysis processing and speech classification, but also the communication computer to assist in the diagnosis of data mining. Machine learning is a process in which a machine looks for and extracts some rules from known empirical data in a certain way (Blum, A., L. , Langley, & P. 1997), and then the rules extracted can be used to judge some unknown things. Pattern recognition is to input a variety of feature descriptions to the machine, so that the machine can judge unknown things.

(3) Data security

Data security is an increasingly serious problem. Taking malware as an example, Israeli firm Deep Learning Technology points out that the code of each new malware is

usually 90% to 92% similar to the old version, so as long as the learning model can tell. The remaining 2 to 10 percent of code changes predict which files are malware. Machine learning can also learn patterns in which data is accessed in the cloud to report anomalies. Machine learning has also proved valuable in travel security, effectively reducing false positives and finding things that human security agents at airports, stadiums, concerts, and so on might miss. It can greatly speed up the security process, and at the same time improve people's safety at important events.

(4) Health care

Machine learning is able to process more information and discover more patterns than humans. Computer-assisted diagnostic technology (CAD) can be used to study early mammograms of women with breast cancer. In 52 percent of breast cancer cases, the computer can identify the disease a full year before it is officially diagnosed. In addition, machine learning can be used to understand risk factors for large groups of diseases. An algorithm developed by MED can predict avoidable hospitalization in patients with diabetes by identifying 8 variables. Machine learning can also be used for early drug discovery. Drug discovery and development is a very expensive and time-consuming work.Machine learning can also be used to maintain electronic health records. The main effect is to simplify the process, save time, effort and Money.

(5) Smart city

Machine learning is also playing a big role in the construction of smart cities. In the case of traffic, machine learning can help to estimate areas that may be congested based on daily experience. When a user bookings an online car-hailing service, the application will estimate the price of the trip, and in the sharing service, uses machine learning to minimize detours. On the city video surveillance system, the current video surveillance

system is driven by artificial intelligence, which can detect crime before it happens.learn, A good model would have detected unusual behaviour and alerted police officers, potentially averting an accident,In addition, when these activities are reported and counted as real incidents, the data will help improve the monitoring service system, which is supported by machine learning on the back end.

2.2 Classical Machine Learning Algorithms

Classical machine learning algorithms mainly include:

(1) Decision tree algorithm

Decision tree algorithm is a kind of supervised learning algorithm, and supervised learning is that each input sample has an attribute or category as a label, through learning to get a classifier. Decision tree is a graphical method for intuitively running probability analysis to get mapping results. It has fast operation speed and high accuracy, does not need parameter assumption and is suitable for high-dimensional data, but it is prone to over-fitting and ignores the correlation between attributes.

(2) Naive Bayes algorithm

The Naive Bayes algorithm is a classification algorithm in which the classifier assumes that each feature generates probability independently and there is no correlation between features. Native Bayes algorithms require relatively little training, but features must be found before making predictions. Individual probability distribution parameters, but this can usually be done quickly and with high accuracy. So Naive Bayes classifiers are very useful in high dimensional data.

(3) Support vector machine algorithm

The support vector machine algorithm first needs to transform the space into higher dimensions and then take the optimal linear classification surface in the new complex space. The problem is transformed into a linear separable problem by the holding vector machine, which greatly simplifies the common classification and regression problems. However, the algorithm consumes too much computing resources when facing large

training samples, so it is difficult to carry out large-scale training and is not suitable for multi-partitioning Kind of a problem.

(4) Artificial neural network algorithm

Artificial neural network algorithm is composed of individual units connected to each other, each unit is similar to a linear combination of functions. The Algorithms need to have a learning rule to learn. When the network makes mistakes in judgment, it can reduce the possibility of making mistakes through learning. Artificial neural network algorithms have a good application in the field of higher performance requirements, but it is very difficult if the neural network has problems. Finding the problem is difficult and time-consuming, and the algorithm requires millions of labeled samples.

(5) Deep learning

Deep learning is to learn the inherent laws of sample data, which are used to model and divide text, image, sound and other data It is of great help in speech and image recognition, and its actual results are far better than those of other previous technologies. A computer's neural network requires huge amounts of data to train a basic skill, while the human mind is highly abstract. So it takes a computer to recognize a cat by looking at a picture of thousands of cats, and even a child has the same ability to look at a cat two or three times. The final point is to explain the "depth" of deep learning, which is the number of layers you go from the "input layer" to the "output layer". the number of layers of "hidden layers". The more layers you have, the deeper the bench.Therefore, the more complex the selection problem is, the more layers of depth are needed.

2.3 BP Artificial Neural Network

2.3.1 Principle and Algorithm of BP Artificial Neural Network

There are many models and algorithms of artificial neural network, among which the neural network of error back transmission training algorithm is one of the more widely used at present. The algorithm is proposed by the group founded by Xian etc. which belongs to the "teacher" type neural network structure, and carries on a detailed analysis of the error back propagation algorithm of the multi-layer perceptron with nonlinear continuous function, and realizes the idea of multi-layer network, namely network. Multiplayer feed-forward neural network refers to neural network, that's short for it, and that's by far. Very wide range of applications.Neural networks. The model is generally divided into three parts, namely input implied output. The hidden layer may include particularly many layers.

BP neural network, for example, consists of two processes. Involving the forward transmission of nerve signals. There's error return.In the process of further propagation, samples will be processed every time they pass through the hidden layer. Once it's done, it goes from the input to the output layer. Therefore, if the result of the output layer is different from the expected result, then the error will be back-propagated to the input part. This input is the way it propagates up layer by layer.

To each neuron to pick up the error signal and correct the error. Through the continuous iteration of forward propagation and back propagation, the weight is adjusted constantly, and finally the signal error can reach an acceptable degree or reach the preset learning times. That is, "positive direction of learning, error, reverse direction, alternate training, error convergence "process, this learning rule is called" generalized rule ".

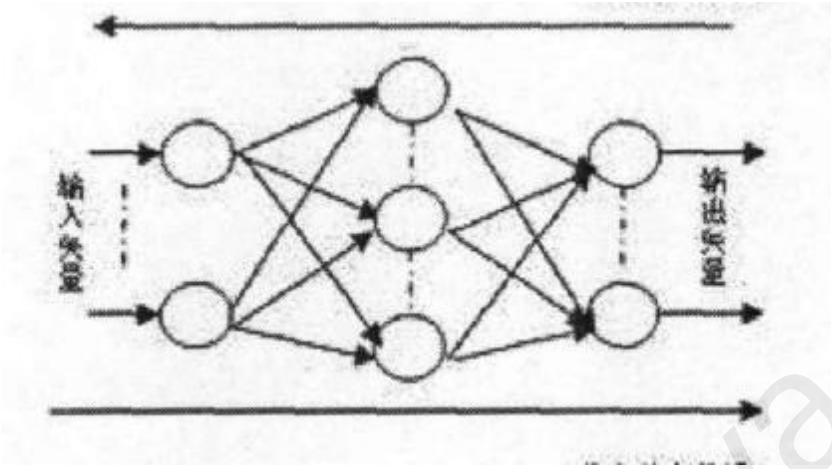

Figure 2-1 BP neural network model

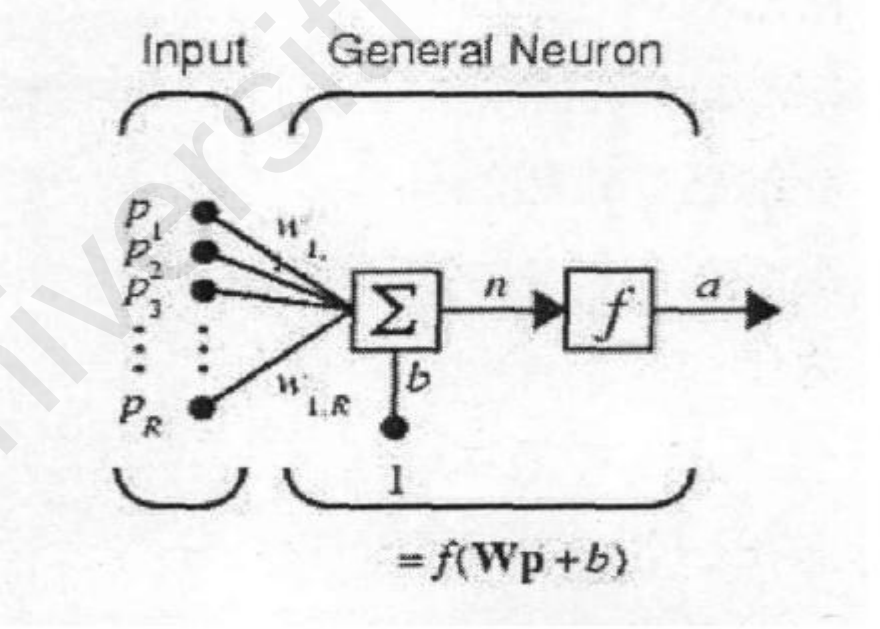

Figure 2-2 BP neuron function expression

A BP model with R inputs is shown in the figure. The product of the independent variable and the corresponding weight vector of each group of inputs plus the illustration value is put into its transfer function, that is, the transfer function f(x) is usually a differentiable monotone increasing function. If the output layer of BP neural network adopts Sigmoid type function, then the output of network is limited to [-1, +1], and the network is used. When the linear function purelin is the transfer function of the output layer, the output can take any value.Therefore, Sigmoid type function is often used in the hidden layer to transfer the intermediate results, while Purelin type transfer function is used in the final output layer to extend the range of the output.

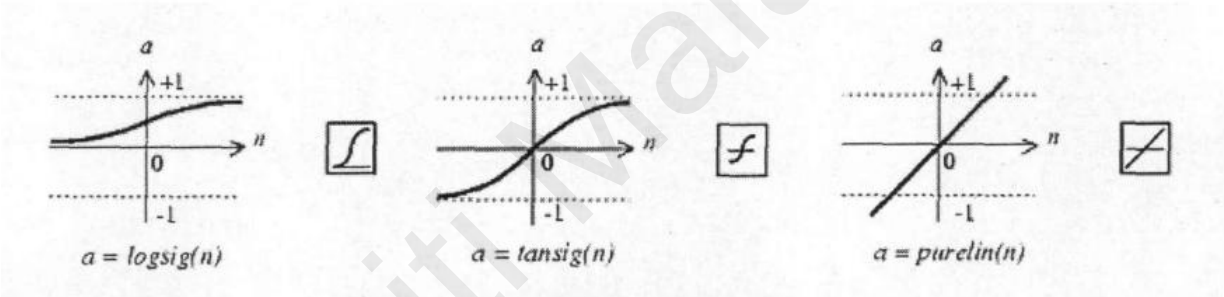

Figure 2-3:Several transfer functions commonly used in BP neurons

The BP neural network is currently widely used, mainly because of the multi-layer sensing device. Has the following functions:

(c) Non-linear mapping capability

A large number of pattern mappings can be stored and learned, the pattern should include input and output, and does not need to understand his mathematical expressions. Only need to provide a large number of sample models to come in, learning and training can complete the nonlinear mapping from the input space to the output space.

(b) Generalization ability

The so-called generalization ability means that the neural network can correctly map from the input place to the input to the output place. This ability is called multi-layer perception. A kind of generalization ability After we train the samples extracted, the relationship of the non-linear mapping is stored in the evidence of the masses.

(c) Fault tolerance.

The biggest advantage is when the input causes a large number of errors, a large number of errors or a single error due to various factors. Through the adjustment of the warrant matrix, statistical features can also be extracted from a large number of samples. So this reflects that the correct law comes from the entire sample, and the error of a single sample will not affect the adjustment of the weight matrix.

2.3.2 Mathematical model of BP neural network

A typical neural network consists of three perceptron layers, including input layer, hidden layer and output layer. Let the input vector be:

$$X = (x_1 \quad x_2 \quad \cdots \quad x_i \quad \cdots \quad x_n)^T$$

The output vector of the output layer is:

$$Y = (y_1 \quad y_2 \quad \cdots \quad y_j \quad \cdots \quad y_m)^T$$

The output vector of hidden layer is:

$$O = (o_1 \quad o_2 \quad \cdots \quad o_k \quad \cdots \quad o_l)^T$$

The expected output vector is :

$$\mathrm{d} = (d_1 \quad d_2 \quad \cdots \quad d_k \quad \cdots \quad d_l)^T$$

The weight matrix between the input layer and the hidden layer is expressed as:

$$\mathrm{V} = \begin{pmatrix} v_1 & v_2 & \cdots & v_j & \cdots & v_m \end{pmatrix}$$

The column vector V is the weight vector corresponding to the JTH neuron in the hidden layer. The weight matrix between hidden layer and output layer Denoted by W:

$$\mathrm{W} = \begin{pmatrix} w_1 & w_2 & \cdots & w_k & \cdots & w_l \end{pmatrix},$$

For the output layer there is:

$$O_k = f(net_k), \quad net_k = \sum_{j=0}^{m} w_{jk} y_i \circ \quad k = 1 \quad 2 \quad \cdots \quad l$$

For the hidden layer there is:

$$y_i = f(net_j), \quad net_j = \sum_{i=0}^{n} v_{ij} x_i \circ \quad j = 1 \quad 2 \quad \cdots \quad m$$

In the above two formulas, the transfer function f(x) is uni-polar Sigmoid function:

$$f(x) = \frac{1}{1+e^{-x}}$$

When the network output is different from the expected output, there is an output error, which is defined as follows:

$$E = \frac{1}{2}(d-O)^2 = \frac{1}{2}\sum_{k=1}^{l}(d_k - O_k)^2$$

The above error definition is extended to the hidden layer as follows:

$$E = \frac{1}{2}\sum_{k=1}^{l}[d_k - f(net_k)]^2 = \frac{1}{2}\sum_{k=1}^{l}[d_k - f(\sum_{j=0}^{m} w_{jk} y_j)]^2$$

Further expanded to the input layer are:

$$E = \frac{1}{2}\sum_{k=1}^{l}[d_k - f(net_k)]^2 = \frac{1}{2}\sum_{k=1}^{l}\{d_k - f[\sum_{j=0}^{m} w_{jk} f(net_j)]\}^2 = \frac{1}{2}\sum_{k=1}^{l}\{d_k - f[\sum_{j=0}^{m} w_{jk} f(\sum_{i=0}^{n} v_{ij} x_i)]\}^2$$

The error signal can be obtained by derivation:

$$\delta_k^o = (d_k - o_k) o_k (1 - o_k)$$

$$\delta_j^y = (\sum_{k=1}^{l} \delta_k^o w_{jk}) y_j (1 - y_j)$$

The weight adjustment formula of neural network learning algorithm is:

$$\Delta w_{jk} = \eta \delta_k^o y_j = \eta (d_k - o_k) o_k (1 - o_k) y_j$$

$$\Delta v_{ij} = \eta \delta_j^y x_i = \eta (\sum_{k=1}^{l} \delta_k^o w_{jk}) y_j (1 - y_j) x_i$$

2.4 RBF Artificial Neural Network

2.4.1 Principle and Algorithm of RBF Artificial Neural Network

The full name of RBF network is radial basis function, which is a neural network algorithm. It is constructed based on the approximation theory. This kind of network learning is equivalent to finding a best fitting value in a multi-dimensional space. And in this whole operation process, the error will be propagated back to the input part and a neural network function of each node. In the radial basis function, generally only a few neurons are used to determine the output. RBF neural network, mainly used for. Price prediction function approximation and identification problems, so it is suitable for studying a prediction model of this article. The core idea of the radial basis function neural network is that the most basic is to use RBF as a hidden element. In this way, a hidden layer space can be formed, and then the input vector can be transformed in the hidden layer space. Convert it to a high-dimensional space, which is confirmed in this way. The central point of BF is that I can still determine the mapping relationship.

(1) In the neural network structure of RBF. The analysis results include the hidden layer of the input layer and the output layer. Then from input to hiding, the conversion is a non-linear conversion. The conversion from hidden to output space is a linear conversion. So RBF is a feed-forward neural network with a single hidden layer. The RBF neural network is shown in the following figure:

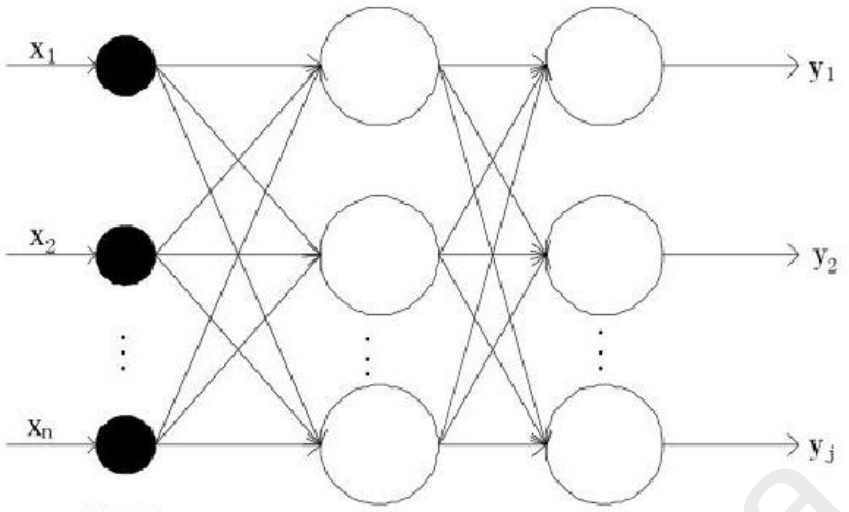

Figure 2-4:Topological structure diagram of RBF neural network

The first input layer: composed of signal source nodes, it only plays the role of data information transmission, and does nothing for input information transform. The second hidden layer: as many nodes as needed.The hidden layer neuron kernel function (action function) is a Gaussian function, which is a logarithmic input Enter the information to carry on the transformation of space mapping. The third output layer responds to the input pattern. The function of the output layer neuron is a linear function of the hidden layer. The output information of the neuron is linearly weighted and output as the output result of the whole neural network.

2.4.2 Mathematical Model of RBF Neural Network

It is the core idea of RBF neural network to take radial basis function as the mapping function of hidden layer neurons. (Blum, A., L., Langley, & P. 1997). The specific structure of the neuron is shown in the figure. It can be found from the figure that the RBF activation function takes the Euclidean distance between input vector and weight as the function input:

$$y = f(\|W - X\| \cdot b) = radbad(\|W - X\| \cdot b)$$

the weight:

$$W = \begin{pmatrix} w_{i1} & w_{i2} & \cdots & w_{in} \end{pmatrix}$$

X is the input vector of neural network:

$$X = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}$$

The following formula is Euclidean distance:

$$\|W - X\|$$

The general form of Gaussian function is:

$$a(x) = e^{-x^2}$$

The Euclidean distance between the input vector X and the weight vector W is solved as follows:

$$\|W - X\| = \sqrt{\sum_{i=1}^{R} (w_{1,i} - x_i)^2} = \sqrt{\left[(W - X) \cdot (W - X)^T \right]}$$

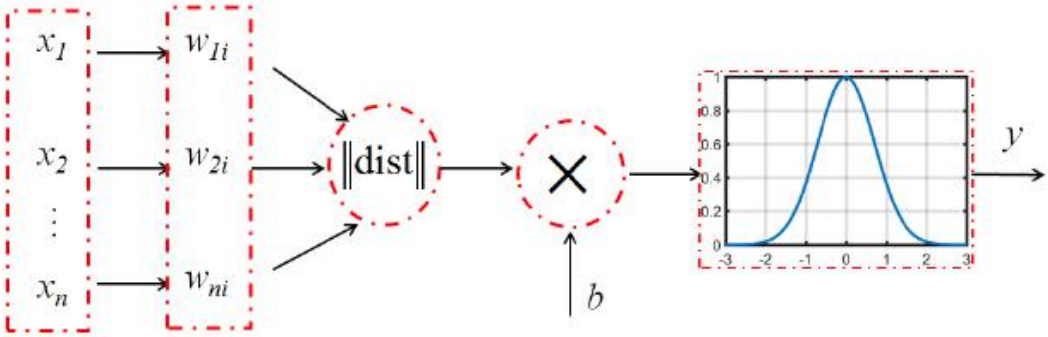

Figure 2-5:Radial basis neuron model

2.5 ELM Artificial Neural Network

2.5.1 Principle and Algorithm of ELM Artificial Neural Network

Neural network has been an important tool for pattern recognition and system identification since its appearance. With the deepening of theoretical research, more kinds of neural networks and corresponding improvement methods have been put forward continuously. These neural networks are suitable for solving different problems according to their own characteristics. The Extreme learning machine belongs to a kind of single-hidden Layer Feed-forward Neural Network (SN traditional Feedforward Neural Network, including BP, RB, etc) which is generally based on gradient descent method for training. The training speed of this training method has always been criticized. It is difficult to select the learning step size, easy to fall into the local minimum value, multiple iterations waste time and so on. Although many scholars have proposed a variety of quick methods, the cumbersome improvement loses the original simplicity of the algorithm and increases the complexity of the algorithm

Huang Gua ngbin pointed out in his research on SLFN that, for a limited data training set, an SLFN neural network with N nodes and a wireless differentiable excitation function in any interval can approach arbitrary linear and nonlinear functions with zero error. In fact, this theory shows that for SLFN, the learning ability is no longer affected by the input weights of the network and the threshold of hidden layer nodes, and there is no need to train the input weights through cumbersome training algorithms. In this case, SLFN is simplified into a linear system, namely ELM. The input weights of ELM and node parameters of the hidden layer are randomly selected, and the establishment and training of the neural network is completed only by solving the pseudo inverse of the hidden layer output weights to obtain the output weight matrix.Experiments show that The adaptive ability and generalization ability of ELM

are very good. Compared with BP neural network and RBF neural network, it has the ability of faster learning, training and approximation.

Reality only has the ability to reduce error. ELM is artificial intelligence.Field machine learning in an artificial neural network model, is a solution, is a single hidden layer of a feed-forward neural network algorithm. For a traditional neural network you have to set a lot of parameters. Using a large number of parameters for training, it can be set up into a network structure, which is also very complex. ELM neural network, there is no need to set other parameters. And it's very easy to use. The weights are randomly generated and do not need to be adjusted, nor do they need to be adjusted throughout the execution, which radically speeds up the computation

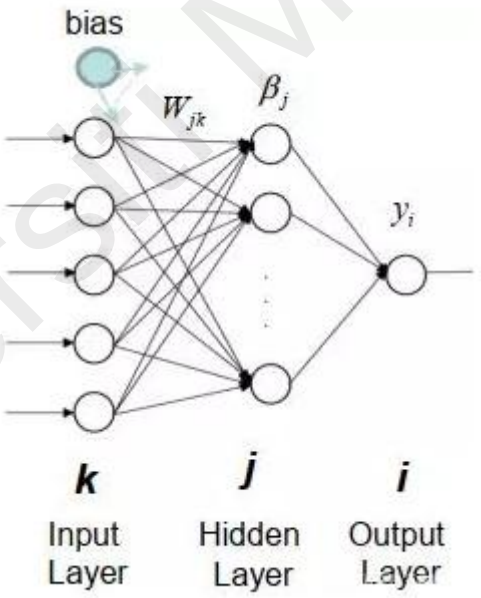

Figure 2-6：ELM artificial neural network model

2.5.2 Mathematical Model of ELM Neural Network

For N different samples (x, t), where：

$$\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$$
$$\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in \mathrm{R}^n, \boldsymbol{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in \mathrm{R}^m$$

The mathematical model of SLFN in which the number of nodes in the hidden layer is M and the excitation function is g(x) is:

$$\sum_{i=1}^{M} \beta_i g_i(\boldsymbol{x}_j) = \sum_{i=1}^{M} \beta_i g(\boldsymbol{a}_i \cdot \boldsymbol{x}_j + b_i) = \boldsymbol{t}_j, \; j = 1, .., N$$

Input weight of connecting the ith hidden layer node:

$$\boldsymbol{a}_i = [a_{i1}, a_{i2}, \ldots, a_{in}]^T$$

Output weights of connecting I hidden layer nodes:

$$\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$$

The excitation function G (x) can be "Sigmoid", "Sine", etc.

The above equation in matrix form can be written as:

$$H\beta = \boldsymbol{T}$$

$$H(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M, b_1, \ldots, b_M, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \begin{bmatrix} g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_1 + b_1) & \ldots & g(\boldsymbol{a}_N \cdot \boldsymbol{x}_1 + b_M) \\ \ldots & \ldots & \ldots \\ g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_N + b_1) & \ldots & g(\boldsymbol{a}_N \cdot \boldsymbol{x}_N + b_M) \end{bmatrix}_{N \times M}, \quad \beta = \begin{bmatrix} \beta_1^T \\ \ldots \\ \beta_M^T \end{bmatrix}_{M \times m}, \quad \boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \ldots \\ \boldsymbol{t}_N^T \end{bmatrix}_{N \times m}$$

E(W) represents the sum of the squares of errors between the expected value and the actual value, and the above problem can be converted into the weight of solving the optimal solution (to minimize E(W) :

$$\mathbf{W} = (\boldsymbol{a}, \boldsymbol{b}, \beta)$$

The mathematical expression can be written as:

$$\underset{\mathbf{W}=(\boldsymbol{a},b,\beta)}{argmin} E(\mathbf{W}) = \underset{\mathbf{W}=(\boldsymbol{a},b,\beta)}{argmin} \| \varepsilon \|^2$$

$$s.t. \sum_{i=1}^{N} \beta_i g(\boldsymbol{a}_i \cdot \boldsymbol{x}_j + b_i) - \boldsymbol{t}_j = \varepsilon_j, \quad j = 1, \ldots, N$$

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{+}\boldsymbol{T} = \left(\boldsymbol{H}^{T}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{T}\boldsymbol{T}$$

Where H+ is called the Moore-Penrose generalized inverse of the hidden layer output matrix H. Therefore, the algorithm of extreme learning machine is given a training sample:

$$\mathrm{N} = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) \mid \boldsymbol{x}_i \in \mathrm{R}^n, \boldsymbol{t}_i \in \mathrm{R}^m, i = 1, \ldots, N\} .$$

And the excitation function G (x) and the number of hidden layer nodes is M.

(1) Randomly select weight I and bias:

$$b_i (i = 1, \ldots, M) :$$

(2) Calculate the hidden layer output matrix

$$\boldsymbol{H} = \begin{bmatrix} g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_1 + b_1) & \cdots & g(\boldsymbol{a}_N \cdot \boldsymbol{x}_1 + b_M) \\ \cdots & \cdots & \cdots \\ g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_N + b_1) & \cdots & g(\boldsymbol{a}_N \cdot \boldsymbol{x}_N + b_M) \end{bmatrix}_{N \times M}$$

(2) Calculate the output weight：

$$\beta : \beta = \left(\boldsymbol{H}^T \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{T}$$

The classification by ELM can achieve good real-time and accuracy.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 MATLAB implementation of BP Neural Network and RBF Neural Network

3.1.1 Mathematical Model of Neural Network

(1) NEWFF: BP neural network parameter setting function

Function: Build a BP neural network

Net = newFF (P, T, S, TF, BTF, BLF, PF, IPF, OPF, DDF)

P: Input data matrix (the input vector of the training set is a matrix composed of columns)

T: Output data matrix (matrix formed by the expected vector of the training set as a column)

S: Number of hidden laminated nodes

Tf: Node transfer function, including hard limiting transfer function hardlim, symmetric hard limiting transfer function handles, linear

BLF: Network learning function, including BP learning rule learning and BP learning rule learned that drives quantity items.

PF: Performance analysis function, including mean absolute error performance analysis function MAE and mean square error performance analysis function MSE.

IPF: The input handler function.

OPF: The output handler function.

DDF: Verify the data partition function.

In general, the first 6 parameters are set in the process of use, and the last 4 parameters are the default parameters of the system.

(2) Train: BP neural network training function

Function: BP neural network is trained with training data.

[net, tr]=train(net, X, T, Pi, Ai)

NET: Network to be trained

X: Input data matrix

T: Output data matrix

PI: Initializes input layer conditions

AI: Initializes output layer conditions

NET: Trained Network

TR: Record the training process

Generally, the first three parameters are set in the process of use, and the last two parameters are the default parameters of the system.

(3) Sim: BP neural network prediction function

Function function: the trained BP neural network is used to predict the function output.

Y = Sim (Net, X)

NET: Trained Network.

X: Input data.

Y: Network forecast data.

3.1.2 Correlation Function of RBF Neural Network

(1) newrb

This function can be used to design an approximate radial basis network. Its invocation format is:

[net,tr]=newrb(X, T, GOAL, SPREAD, MN, DF)

Wherein, X is an R*Q dimensional matrix composed of Q group of input vectors. T is the S*Q dimension composed of target classification vectors of group Q Matrix; Goal is Mean Squared Error Goal, which defaults to 0.0; Spread is the expansion speed of

radial basis function, which defaults to 1. Mn is the maximum number of neurons, which defaults to Q. DF is the number of neurons added between two displays, which defaults to 25; Net is the return value, an RBF network is the return value, tr is the return value, the training record uses newrb() to create RBF network is a process of continuous trial, in the process of creation, need to increase the number of neurons in the middle layer continuously, until the output error of the network meets the preset value.

(2) newrbe()

This function is used to design an exact radial basis network, whose call format is net= newrbe (X, T, SOREAD). Where X is an R*Q dimensional matrix composed of Q group of input vectors; T is the S*Q dimension composed of target classification vectors of group Q Matrix; Spread is the expansion speed of the radial basis function, which defaults to 1. Unlike newrb(), newrbe() can quickly and error-free design a radial basis network based on design vectors. Suitable for the case of a small number of input vectors.

(3) radbas()

This function is the radial basis transfer function. Its invocation format is

A=radbas(N)

Info=radbas(code)

Where, N is the S*Q dimension matrix of the input (column) vector; A is the return matrix of the function, which is one-to-one corresponding to N. Each element in N is obtained by A radial basis function. Info =radbas(code) means return according to the code value. Different information about functions. Include:

Derive ----- Return the name of the derivate function.

Output ---- returns the input range.

Active ----- returns a range of available inputs

3.2 Neural Network Parallel Operation

The MATLAB parallel computing toolbox can parallelize the neural network simulation calculation, thus reducing the computing time. The implementation of rowization is to split the whole training data set into different workers of the (multi-core) computer, and the operation of the whole data set. (Blum, A., L., Langley, & P. 1997). The calculated result is the calculated value of all the merged workers. For parallel operation of neural network, it is necessary to open MatlabPool first, and set UseParallel parameter in neural network train and SIM function:

```
%% OPEN MATLAB workers
matlabpool open

% check worker number
poolsize=matlabpool('size')

%% set train and sim Parameters in the function “Useparallel” is “yes”。
net2=train(net1,x,t,'Useparallel','yes')
y=sim(net2,x,'Useparallel','yes')]
```

3.2.1 BP Neural Network for Nonlinear Function Fitting

$$y = x_1^2 + x_2^2$$

The first step is data selection and normalization. 2000 sets of data of the function are randomly obtained according to the nonlinear function equation, and the data are stored in the data.mat file (after downloading and copying to the current directory of MATLAB), where input is the function input data and output is the function output data. 1900 data were randomly selected from the input and output data as the network training data and 100 groups as the network test data, and the data were normalized.

```
%% Clear environment variables
clc
clear

%% Training data prediction data extraction and normalization
% Load input and output dataload data input output
% Randomly sort from 1 to 2000
k=rand(1,2000);
[m,n]=sort(k);

% Find out the training data and prediction data
input_train=input(n(1:1900),:)';
output_train=output(n(1:1900));
input_test=input(n(1901:2000),:)';
output_test=output(n(1901:2000));

% Normalization of selected sample input and output data
[inputn,inputps]=mapminmax(input_train);
[outputn,outputps]=mapminmax(output_train);
```

The second step is to establish and train BP neural network

BP neural network is constructed and trained with training data, so that the network can predict the output of nonlinear function.

```
%%Build and train BP neural network
% BP neural network construction
net=newff(inputn,outputn,5);

%Network parameter configuration (number of iterations, learning rate, target)
net.trainParam.epochs=100;
net.trainParam.lr=0.1;
net.trainParam.goal=0.00004;

%Network training, and record the training time
tic
net=train(net,inputn,outputn);
t1=toc
disp(['The training time of the neural network is',num2str(t1),'seconds']);
```

The third step is BP neural network prediction. The trained BP neural network is used to predict the output of nonlinear function.

```
%% BP network prediction
% Forecast data normalization
inputn_test=mapminmax('apply',input_test,inputps);

% Network prediction output
an=sim(net,inputn_test);

% Network output denormalization
BPoutput=mapminmax('reverse',an,outputps);
```

The fourth step is the result analysis. The fitting ability of BP neural network is analyzed by the predicted output and expected output of BP neural network.

```
%% Result analysis
figure(1)
plot(BPoutput,':og')
hold on
plot(output_test,'-*');
legend('Predicted output','Expected output')
title('BP network prediction output','fontsize',12)
ylabel('Function output','fontsize',12)
xlabel('sample','fontsize',12)
% Prediction error
error=BPoutput-output_test;
figure(2)

plot(error,'-*')
title('BP network prediction error','fontsize',12)
ylabel('error','fontsize',12)
xlabel('sample','fontsize',12)
figure(3)
plot((output_test-BPoutput)./BPoutput,'-*');
title('Percentage of neural network prediction error')
errorsum=sum(abs(error));
```

3.2.2 RBF Neural Network for Nonlinear Function Fitting

The first step is to establish the exact RBF neural network fitting and observe the fitting effect

```
% Regression of RBF Network--Realization of Nonlinear Function Regression
%% Clear environment variables
clc
clear
%% Generate input and output data
% Set step size
interval=0.01;
% Produces x1 x2
x1=-1.5:interval:1.5;
x2=-1.5:interval:1.5;
% According to the function, first obtain the corresponding function value as the output of the network.
F =20+x1.^2-10*cos(2*pi*x1)+x2.^2-10*cos(2*pi*x2);
%% Network establishment and training
% Network establishment The input is [x1;x2], and the output is F. Spread uses the default.
net=newrbe([x1;x2],F)
%% network performance verification
% Bring the original data back to test the network effect:
```

```
ty=sim(net,[x1;x2]);
% Use images to see how the network fits the nonlinear function
figure
plot3(x1,x2,F,'rd');
hold on;
plot3(x1,x2,ty,'b-.');
view(113,36)
title('Visualization method to observe the fitting effect of accurate RBF
neural network')
xlabel('x1')
ylabel('x2')
zlabel('F')
grid on
```

The second step establishes the Approximate RBF neural network fitting

```
x=rand(2,ld);
% Convert x to between [-1.5 1.5]
x=(x-0.5)*1.5*2;
The first line of% x is x1, and the second line is x2.
x1=x(1,:);
x2=x(2,:);
% Calculate the network output F value
F=20+x1.^2-10*cos(2*pi*x1)+x2.^2-10*cos(2*pi*x2);
%% Build RBF neural network
% Adopt approximate RBF neural network. spread is the default
net=newrb(x,F);
```

```
% generate the testing data
interval=0.1;
[i, j]=meshgrid(-1.5:interval:1.5);
row=size(i);
tx1=i(:);
tx1=tx1';
tx2=j(:);
tx2=tx2';
tx=[tx1;tx2];
%% Use the established RBF network to simulate and get the network

output
ty=sim(net,tx);
%% Use images to draw 3D graphs
% Real function image
interval=0.1;
[x1, x2]=meshgrid(-1.5:interval:1.5);
F = 20+x1.^2-10*cos(2*pi*x1)+x2.^2-10*cos(2*pi*x2);
subplot(1,3,1)
mesh(x1,x2,F);
zlim([0,60])
title('Real function image')
```

```
% Function image derived from the network
v=reshape(ty,row);
subplot(1,3,2)
mesh(i,j,v);
zlim([0,60])
title('RBF neural network result')
% Error image
subplot(1,3,3)
mesh(x1,x2,F-v);
zlim([0,60])
title('error image')

set(gcf,'position',[300 ,250,900,400])
```

3.3 Octane Number Prediction for Gasoline Blends

3.3.1 Problem Formulation

Octane number is the most important quality index of gasoline. Traditional laboratory testing methods have problems such as large sample consumption, long test cycle and high cost, and are not suitable for production control. Especially for online testing, the near infrared spectroscopy (NIR) developed this year, as a fast analysis method, is not suitable for production control. It has been widely used in agriculture, pharmacy, biochemistry, petroleum products and other fields. Its advantages are nondestructive testing, low cost, no pollution, online analysis, more suitable for production and control needs.

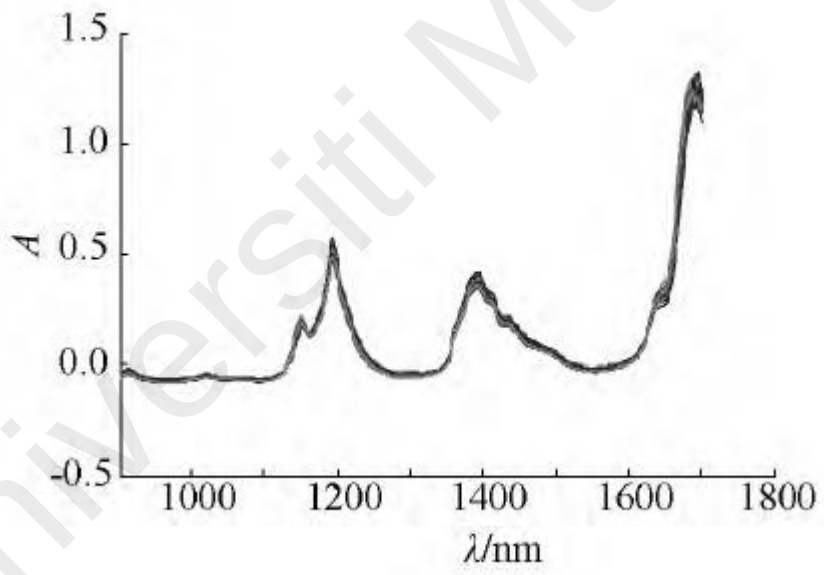

Figure 3-1:Near Infra-red Spectrum of 60 Samples

3.3.2 Design

According to the requirements in the problem description, the model establishment and performance evaluation of BP neural network and RBF neural network are realized, which can be divided into the following steps in general:

1) Generate a training set/test set

In order to ensure that the established model has good generalization ability, the training set samples are required to be large enough and have good representativeness.It is generally believed that it is appropriate for the number of samples in the training set to account for 2/3~3/4 of the total number of samples, and the remaining 1/4~1/3 of the samples in the test set. And try to make the distribution law of samples of training set and test set approximately the same.

2) Create/train BP neural network

The establishment of BP neural network needs to determine the structure of the network, that is, the following parameters need to be determined: the number of input variables, the number of hidden function layers, the number of neurons at each layer, and the number of output variables. As can be seen from the problem description, the number of input variables is 401 and the number of output variables is 1. There is one hidden layer (which can approximate any nonlinear function). The number of hidden layer neurons has a great influence on the performance of BP neural network. If the number of hidden layer neurons is small, the network cannot adequately describe the relationship between output and input variables.On the contrary, if there are more neurons in the hidden layer, the learning time of the network will be longer, and even the problem of over-fitting will appear. Generally, the method to determine the number of neurons in the hidden layer is based on the empirical formula, and the influence of different number of neurons in the hidden layer on the model performance is compared, so as to make the selection.After the network structure is determined, the network can be trained by setting relevant training parameters (such as training times, learning rate, etc.).

3) Create/train RBF neural network

When creating RBF neural network, the influence of spread value on network performance should be considered.In general, the larger the value of spread, the smoother the fitting of the function will be. However, an excessively large SPREAD value would require a very large number of neurons to accommodate rapid changes in the function; On the contrary, if the spread value is too small, it means that many neurons are needed to adapt to the slow change of the function, leading to poor network performance.

4) Simulation test

After the model is established, the input variables of the test set are sent into the model, and the output of the model is the corresponding prediction result.

5) Performance evaluation

The generalization ability of the model can be evaluated by calculating the error between the predicted value and the real value of the test set. On this basis, further research and improvement can be carried out.

3.3.3 Realization

The first step is to generate the training set/test set. Spectral and octane data of the 60 samples are stored in the spectra_data.Mat file, which contains two variable matrices: spectral data of the sample with 60 rows and 401 columns for NIR and octane data with 60 rows and 1 column for Octane. Here, the training set and test set are generated by random methods. 50 samples are randomly generated as the training set and the remaining 10 samples as the test set. The procedure is as follows:

```
%%
% 4.Simulation test
t_sim = sim(net,p_test);

%%
% 5. Data denormalization
```

```
%% II.Training set/test set generation
%%
% 1. Import Data
load spectra_data.mat

%%
% 2. Randomly generate training set and test set
temp = randperm(size(NIR,1));
% Training set-50 samples
P_train = NIR(temp(1:50),:)';
T_train = octane(temp(1:50),:)';
% Test set-10 samples
P_test = NIR(temp(51:end),:)';
T_test = octane(temp(51:end),:)';
N = size(P_test,2);

%% III. Data normalization
[p_train, ps_input] = mapminmax(P_train,0,1);
p_test = mapminmax('apply',P_test,ps_input);

[t_train, ps_output] = mapminmax(T_train,0,1);
```

(1) Due to the randomness generated by the training set/test set, the results of each run may be different.

(3) The function random(n) is used to generate a random sequence of positive integers of length n. The second step is to create/train BP neural network and simulation test. The use of MATLAB neural network toolbox functions can be convenient for BP neural network creation, training and simulation test. follows: Create RBF neural network and simulation test. The use of MATLAB neural network toolbox functions, can be convenient for RBF neural network creation, training and simulation test. The procedure is as follows:

```
%% III. RBFneural network creation, training and simulation test
%%
% 1. Create a network
net = newrbe(P_train, T_train, 30);

%%
% 2. Simulation test
T_sim = sim(net, P_test);

%% IV. Performance evaluation
%%
% 1. Relative error
error = abs(T_sim - T_test)./T_test;

%%
% 2. Coefficient of determination R^2
R2 = (N * sum(T_sim .* T_test) - sum(T_sim) * sum(T_test))^2 / ((N * sum((T_sim).

%%
% 3.  Comparative Results
result = [T_test' T_sim' error']
```

The fourth step is performance evaluation

After the simulation test of BP neural network and RBF neural network, the generalization ability of the network can be evaluated by calculating the deviation between the predicted value and the real value. Two evaluation indexes are adopted, which are relative error E and determination coefficient 2R respectively, and their calculation formulas are respectively

$$E_i = \frac{|\hat{y}_i - y_i|}{y_i} \quad , \quad i = 1, 2, ..., n$$

$$R^2 = \frac{\left(n\sum_{i=1}^{n} \hat{y}_i y_i - \sum_{i=1}^{n} y_i \right)^2}{\left(n\sum_{i=1}^{n} \hat{y}_i^2 - \left(\sum_{i=1}^{n} \hat{y}_i \right)^2 \right)\left(n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i \right)^2 \right)}$$

The smaller the relative error is, the better the model performance is. Within the range of [0,1], the closer the coefficient is to 1, the better the fitting performance of the model; otherwise, the closer it is to 0, the worse the fitting performance of the model. The procedure is as follows:

BP

```
%% V. Performance evaluation
%%
% 1. Relative error
error = abs(T_sim - T_test)./T_test;

%%
% 2. Coefficient of determination R^2
R2 = (N * sum(T_sim .* T_test) - sum(T_sim) * sum(T_test))^2 / ((N * sum((T_sim).

%%
% 3. Comparative Results
result = [T_test' T_sim' error']
```

RBF

```
%% IV. Performance evaluation
%%
% 1. Relative error
error = abs(T_sim - T_test)./T_test;

%%
% 2. Coefficient of determination R^2
R2 = (N * sum(T_sim .* T_test) - sum(T_sim) * sum(T_test))^2 / ((N * sum((T_sim).

%%
% 3.  Comparative Results
result = [T_test' T_sim' error']
```

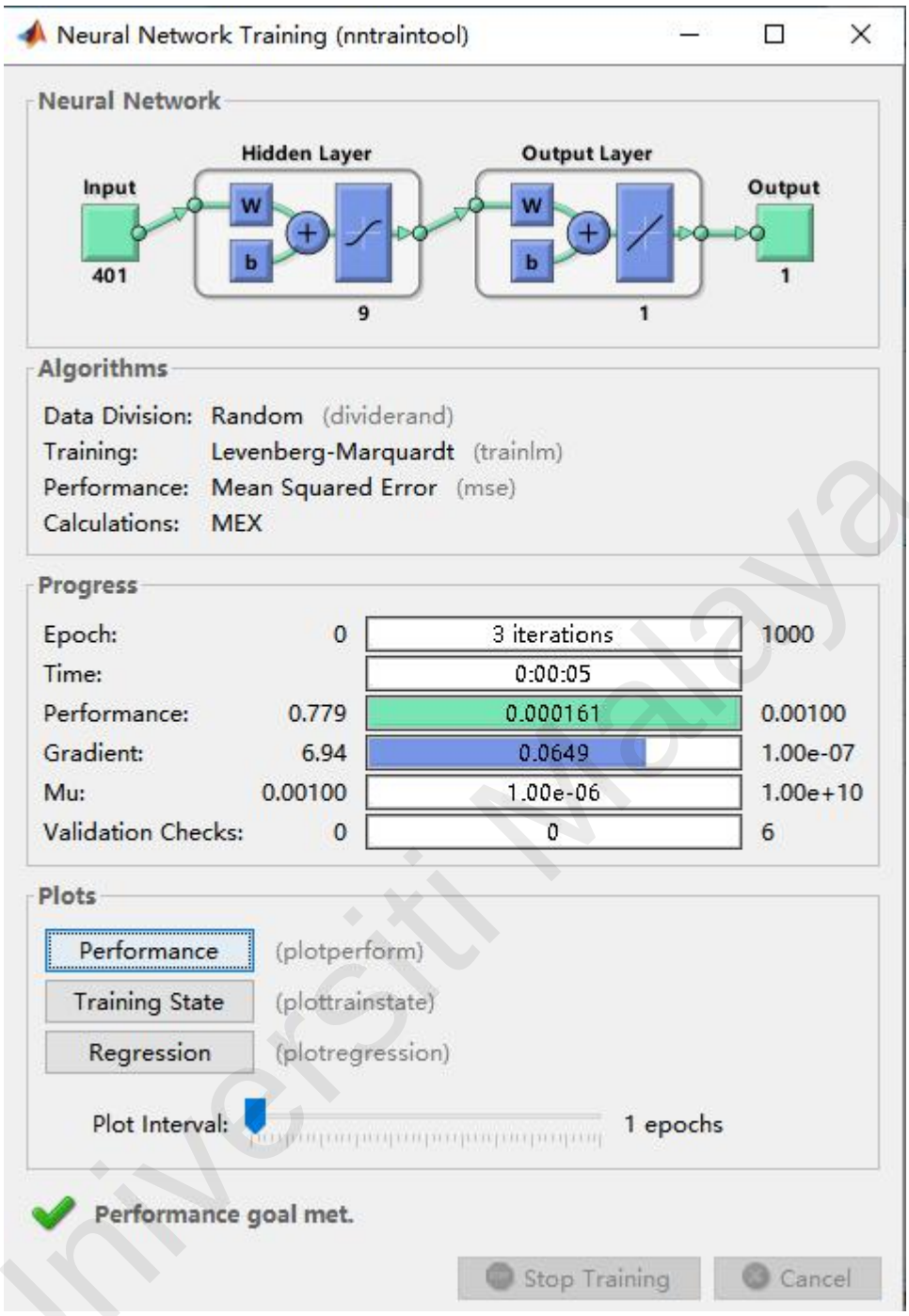

Figure 3-2:Neural Network Training

Step 5 Drawing

In order to observe and analyze the results more intuitively, the results are presented in the form of pictures. The procedure is as follows:

BP:

```
%% VI. Drawing
figure
plot(1:N,T_test,'b:*',1:N,T_sim,'r-o')
legend('actual value','Predictive value')
xlabel('Prediction sample')
ylabel('Octane number')
string = {'Comparison of prediction results of octane content in test sets';['R^2=' num2str(R2)]};
title(string)
```

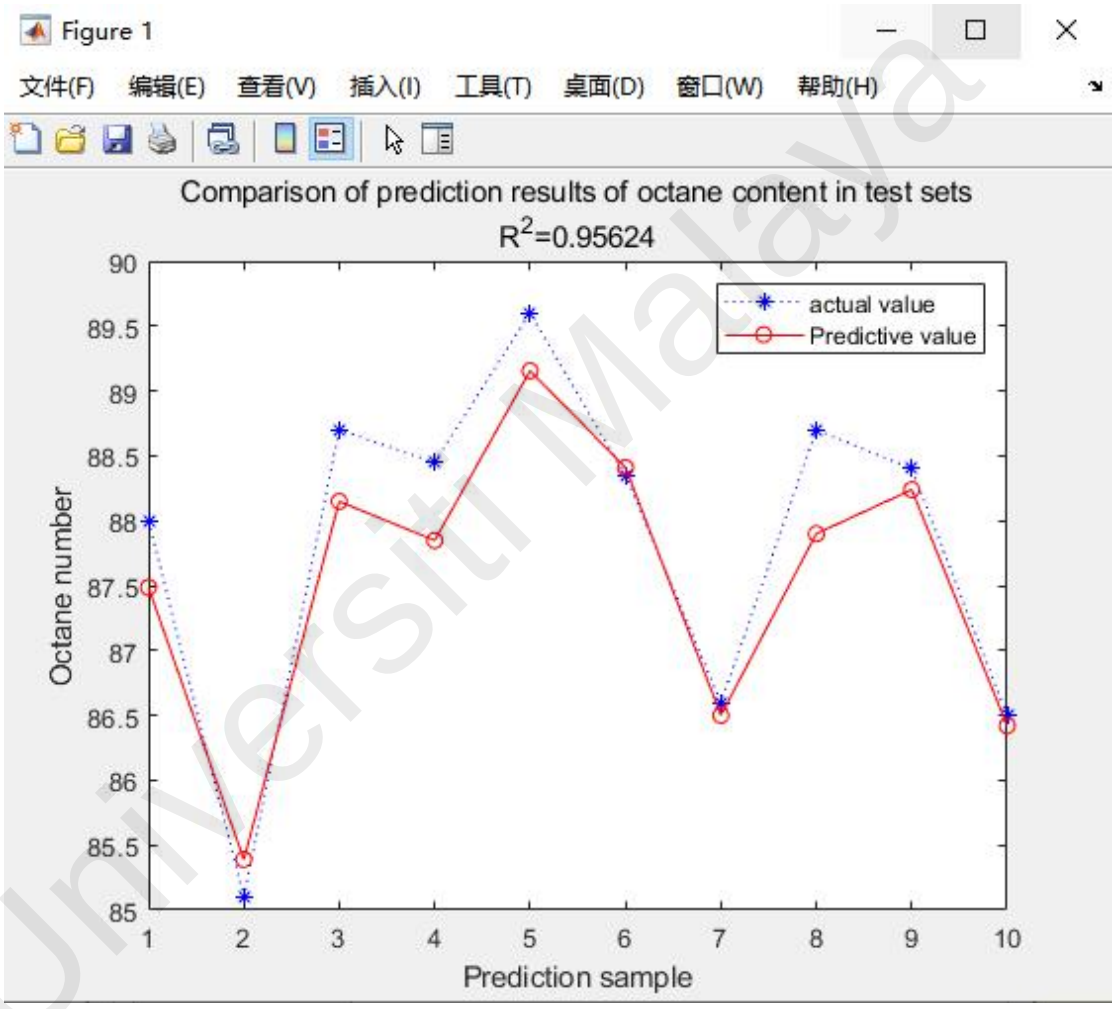

Figure 3-3 :COMPARISON OF PREDICTION RESULTS(BP)

RBF:

```
%% V. Drawing
figure
plot(1:N,T_test,'b:*',1:N,T_sim,'r-o')
legend('actual value','Predictive value')
xlabel('Prediction sample')
ylabel('Octane number')
string = {'Comparison of prediction results of octane content in test sets';['R^2=' num2str(R2)]};
title(string)
```

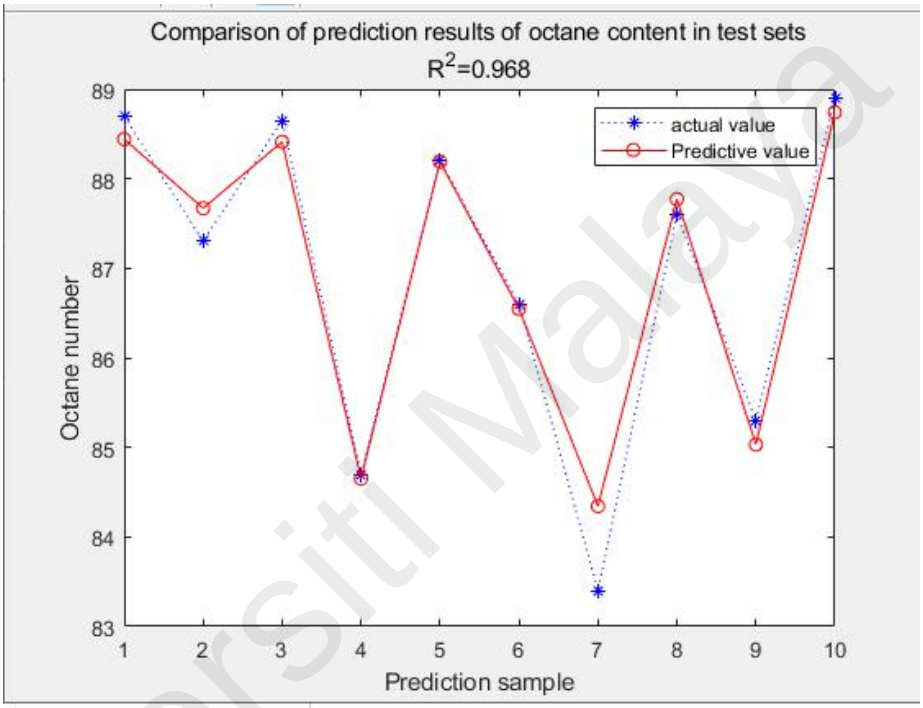

Figure 3-4:COMPARISON OF PREDICTION RESULTS(RBF)

3.4 Network Parameters Selection

The number of hidden layer neurons has a great influence on the performance of BP neural network. If the number of hidden layer neurons is small, the network cannot adequately describe the relationship between output and input variables. On the contrary, if there are more neurons in the hidden layer, the learning time of the network will be longer, and even the problem of over-fitting will appear. Generally, the method to determine the number of neurons in the hidden layer is based on the empirical formula, and the influence of different number of neurons in the hidden layer on the model performance is compared, so as to make the selection. Specific method: When the number of neurons in the hidden layer was 3~10, the program ran 10 times and the average value of the corresponding determination coefficient 2R was obtained. The number of hidden layer neurons corresponding to the maximum mean value is the best choice. For RBF neural network, generally speaking, the greater the value of spread, the smoother the fitting of the function will be. However, an excessively large SPREAD value would require a very large number of neurons to accommodate rapid changes in the function; On the contrary, if the spread value is too small, it means that many neurons need to adapt to the slow change of the function, leading to poor network performance. Specific methods: When spread values are 0.1,0.2,...When 0.9 and 1, the program runs 10 times, and the average value of the corresponding determination coefficient 2R is obtained. The spread value corresponding to the maximum average value is the best choice (programmed myself to find the best value, and drew the spread value -- the graph of determining coefficient)

3.5 ELM

3.5.1 Interpretation Function

NARGIN: NARGIN is an abbreviation of "number of input arguments". When you define a function in MATLAB, within the body of the function, the margin is the function used to determine the number of input variables. To get help information for this function, type Help Nargin or Doc Nargin in the MATLAB command window.

PINV: Find the pseudo inverse matrix

Sin/hardlim: The function hardlim(N), given the input vector matrix N of the network, returns the output vector matrix A for that layer. If the element in N is greater than or equal to zero, the value returned is L; Otherwise it's 0. That is to say, if the network input reaches the threshold, then the output of the hard-limiting transmission function is 1; Otherwise, it is 0.

Custom function description:

Elmtrain:

```
function [IW,B,LW,TF,TYPE] = elmtrain(P,T,N,TF,TYPE)
% ELMTRAIN Create and Train a Extreme Learning Machine
% Syntax
% [IW,B,LW,TF,TYPE] = elmtrain(P,T,N,TF,TYPE)
% Description
% Input
% P   - Input Matrix of Training Set  (R*Q)
% T   - Output Matrix of Training Set (S*Q)
% N   - Number of Hidden Neurons (default = Q)
% TF  - Transfer Function:
%       'sig' for Sigmoidal function (default)
%       'sin' for Sine function
%       'hardlim' for Hardlim function
% TYPE - Regression (0,default) or Classification (1)
% Output
% IW  - Input Weight Matrix (N*R)
% B   - Bias Matrix  (N*1)
% LW  - Layer Weight Matrix (N*S)
% Example
% Regression:
% [IW,B,LW,TF,TYPE] = elmtrain(P,T,20,'sig',0)
% Y = elmtrain(P,IW,B,LW,TF,TYPE)
% Classification
```

```
if nargin < 2
    error('ELM:Arguments','Not enough input arguments.');
end
if nargin < 3
    N = size(P,2);
end
if nargin < 4
    TF = 'sig';
end
if nargin < 5
    TYPE = 0;
end
if size(P,2) ~= size(T,2)
    error('ELM:Arguments','The columns of P and T must be same.');
end
[R,Q] = size(P);
if TYPE  == 1
    T  = ind2vec(T);
end
[S,Q] = size(T);
% Randomly Generate the Input Weight Matrix
IW = rand(N,R) * 2 - 1;
```

```
% Randomly Generate the Bias Matrix
B = rand(N,1);
BiasMatrix = repmat(B,1,Q);
% Calculate the Layer Output Matrix H
tempH = IW * P + BiasMatrix;
switch TF
    case 'sig'
        H = 1 ./ (1 + exp(-tempH));
    case 'sin'
        H = sin(tempH);
    case 'hardlim'
        H = hardlim(tempH);
end
% Calculate the Output Weight Matrix
LW = pinv(H') * T';
```

ElmPredict:

```
function Y = elmpredict(P,IW,B,LW,TF,TYPE)
% ELMPREDICT Simulate a Extreme Learning Machine
% Syntax
% Y = elmtrain(P,IW,B,LW,TF,TYPE)
% Description
% Input
% P   - Input Matrix of Training Set  (R*Q)
% IW  - Input Weight Matrix (N*R)
% B   - Bias Matrix  (N*1)
% LW  - Layer Weight Matrix (N*S)
% TF  - Transfer Function:
%       'sig' for Sigmoidal function (default)
%       'sin' for Sine function
%       'hardlim' for Hardlim function
% TYPE - Regression (0,default) or Classification (1)
% Output
% Y   - Simulate Output Matrix (S*Q)
% Example
% Regression:
% [IW,B,LW,TF,TYPE] = elmtrain(P,T,20,'sig',0)
% Y = elmtrain(P,IW,B,LW,TF,TYPE)
% Classification
% [IW,B,LW,TF,TYPE] = elmtrain(P,T,20,'sig',1)
```

```
% Y = elmtrain(P,IW,B,LW,TF,TYPE)
if nargin < 6
    error('ELM:Arguments','Not enough input arguments.');
end
% Calculate the Layer Output Matrix H
Q = size(P,2);
BiasMatrix = repmat(B,1,Q);
tempH = IW * P + BiasMatrix;
switch TF
    case 'sig'
        H = 1 ./ (1 + exp(-tempH));
    case 'sin'
        H = sin(tempH);
    case 'hardlim'
        H = hardlim(tempH);
end
% Calculate the Simulate Output
Y = (H' * LW)';
if TYPE == 1
    temp_Y = zeros(size(Y));
    for i = 1:size(Y,2)
        [~,index] = max(Y(:,i));
        temp_Y(index,i) = 1;
    end
    Y = vec2ind(temp_Y);
end
```

3.5.2 Simulation

1.Random assignment of node parameters: At the beginning of the calculation, node parameters are independent of input data and their parameters are randomly generated.

2. Calculate the output matrix of hidden layer: input the training data as the number of rows first, and then set the number of hidden nodes as the number of columns

3.Solve output weights: For ELM algorithm, the most important and key significance is to solve the output weights, because solving the output weights can minimize the error function.

4.In the evaluation part, the error value can be found, and the generalization ability of the model can be evaluated by measuring the error value, so as to conduct further in-depth study and improve the study

3.5.3 Realization

The first step is to generate the training set/test set:

Spectral and octane number data of 60 samples are stored in the spectra_data.mat file, which contains two variable momentsArray: NIR is the sample spectral data of 60 rows and 401 columns, Octane is the octane number data of 60 rows and 1 column. Here, the training set and test set are generated by random method. 50 samples are randomly generated as the training set and the remaining 10 samples as the test set. The procedure is as follows:

```
%% II. Training set/test set generation
%%
% 1. Import Data
load spectra_data.mat

%%
% 2. Randomly generate training set and test set
temp = randperm(size(NIR,1));

% Training set-50 samples
P_train = NIR(temp(1:50),:)';
T_train = octane(temp(1:50),:)';

%Test set-10 samples
P_test = NIR(temp(51:end),:)';
T_test = octane(temp(51:end),:)';
N = size(P_test,2);
```

The second step is data normalization:

```
%% III. Data normalization
%%
% 1. Training set
[Pn_train,inputps] = mapminmax(P_train);
Pn_test = mapminmax('apply',P_test,inputps);
%%
% 2. Test set
[Tn_train,outputps] = mapminmax(T_train);
Tn_test = mapminmax('apply',T_test,outputps);
```

The third step

Create/train neural networks and simulation tests. The use of MATLAB neural network toolbox functions, you can facilitate the creation and training of ELM neural network, And simulation test. The relevant training parameters can be set before training, or they can be set by default. The procedure is as follows:

```
%% IV. ELM Create/train
[IW,B,LW,TF,TYPE] = elmtrain(Pn_train,Tn_train,30,'sig',0);

%% V. ELM Simulation test
tn_sim = elmpredict(Pn_test,IW,B,LW,TF,TYPE);
%%
% 1. Denormalization
T_sim = mapminmax('reverse',tn_sim,outputps);
```

The fourth step

Performance evaluation

```
%% VI. Comparative Results
result = [T_test' T_sim'];
%%
% 1. Mean square error
E = mse(T_sim - T_test);

%%
% 2. decisive factor
N = length(T_test);
R2=(N*sum(T_sim.*T_test)-sum(T_sim)*sum(T_test))^2/((N*sum((T_sim).
```

Drawing: In order to observe and analyze the results more intuitively, the results are presented in the form of pictures. Procedure is as follows

```
%% VII. Drawing
figure(1)
plot(1:N, T_test, 'r-*', 1:N, T_sim, 'b:o')
grid on
legend('actual value', 'Predictive value')
xlabel('Prediction sample')
ylabel('Octane number')
string = {'Comparison of octane content prediction results in test sets (ELM)';
title(string)
```

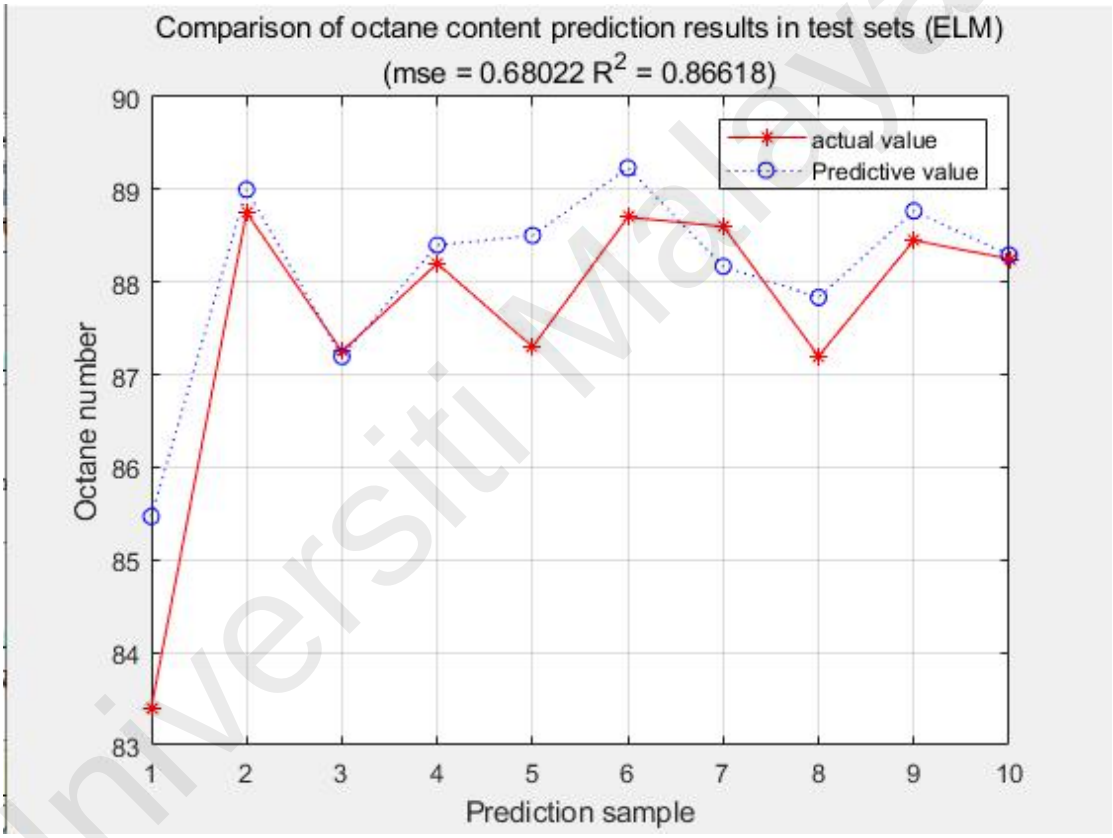

Figure 3-5:COMPARISON OF PREDICTION RESULTS(ELM)

CHAPTER 4: RESULTS AND DISCUSSION

Single hidden Layer Feed-forward Neural Network, SLFN has been widely used in many fields because of its good learning ability. However, some inherent shortcomings of traditional learning algorithms (such as BP algorithm, etc.) have become the main bottleneck restricting its development. Most feed-forward neural networks adopt gradient descent method, which mainly has the following shortcomings:

A. Slow training.

B. It is easy to fall into the local minimum point and cannot reach the global minimum.

C. Sensitive selection of learning rate.

The Extreme Learning machine is a new algorithm for SLFN. The algorithm randomly generates the connection value between input layer and hidden layer and the threshold value of hidden layer neurons, and the unique global optimal solution can be obtained by setting the number of hidden layer neurons without adjustment in the training process. Compared with traditional training methods, this method has the advantages of fast learning speed and good generalization performance. The ELM network training process is very simple. But studies have shown that: In order to compare and analyze the prediction accuracy of step by step linear regression and neural network algorithm, we selected a case from the data level analysis and EVIEWS application server edited by Professor Yi Danhui. This case shows the level of vehicle gasoline consumption (in thousands of gallons) and vehicle ownership X1 (in units) in the United States from 1950 to 1987. motor vehicle gasoline retail price X2 (unit: US $/ gallon), population number X3 (unit: thousand people) and GNP

GNPX4 (unit: US $1 billion). Multiple linear regression methods are required to regression fit the relationship between QMG and MOB, PMG, POP and GNP. This is a typical case of violating the condition of multiple linear regression, with multicollinearity, heteroscedasticity and autocorrelation. Stepwise regression was adopted to eliminate its influence as far as possible. MATLAB numerical calculation software was used to obtain the stepwise regression equation as follows: The variable of population number is deleted from the stepwise regression equation, indicating that this variable is redundant. The prediction is made by using the regression equation. Using the MATLAB neural network toolbox, the above three kinds of neural network intelligent algorithm can be very convenient to achieve, write the corresponding calculation program, the original data normalization, with Y as the output variable, X1, X2, X3, X4 as the input variable:

$$Y = 17025279.4234 + 1.36407929968 * X1 - 27235486.5738 * X2 - 30106.6573861 * X4$$

BP neural network (hidden layer 6), RBF radial basis neural network (expansion coefficient 0 smooth 3) and extreme learning machine network (hidden layer 6) were used to fit these data respectively. During network training, the first 33 data were used for training, and the last 5 data were used for testing. Comparison of the four predicted values is shown in Figure 4.1, and comparison of the relative errors of the four prediction methods is shown in Figure 4.2.

Time	The real value	BP prediction relative error	RBF forecast relative error	ELM forecast relative error
1950	40617285	004148	0	0 . 096
1951	43896887	0. 01643	1. 018e – 15	0. 0587
1952	46428148	0. 00668	3. 21e – 16	0. 01632
1953	49374047	0. 00217	9. 054e – 16	0. 0045
1954	51107135	0. 0146	2. 915e – 16	0. 0173
1955	54333255	0. 01282	1. 37e – 16	0. 02515
1956	56022406	0. 0131	1. 3299e – 16	0. 03345
1957	57415622	0. 0073	2. 59e – 16	0. 0304
1958	59154330	0. 0228	0	0. 0526
1959	61596548	0. 03467	2. 419e – 16	0. 0539
1960	62811854	0. 03572	2. 37e – 16	0. 0440
1961	63978489	0. 03367	1. 16e – 16	0. 03944
1962	62531373	0. 01647	2. 38e – 16	0. 02768
1963	64779104	0. 00770	1. 150e – 16	0. 03318
1964	67663848	0. 00059	4. 48e – 16	0. 0318
1965	70337126	0. 000234	2. 1e – 16	0. 0438
1966	73638812	0. 00235	0	0. 0477
1967	76139326	0. 0093	0	0. 0447
1968	80772657	0. 00708	0	0. 0323
1969	85416084	0. 017233	0	0. 00300
1970	88684050	0. 017143	3. 3e – 16	0. 01756
1971	92194620	0. 00297	1. 6e – 16	0. 00604
1972	95348904	0. 0095	0	0. 01728
1973	99804600	0. 00283	0	0. 0152
1974	100212210	0. 00694	0	0. 02181
1975	102327750	0. 00654	2. 9e – 16	0. 0417
1976	106972740	0. 01002	0	0. 0375
1977	110023410	0. 01607	1. 3e – 16	0. 01969
1978	113625960	0. 03086	1. 3e – 16	0. 01498
1979	107831220	0. 00106	1. 38e – 16	0. 0253
1980	100856070	0. 00615	0	0. 00611
1981	100994040	0. 03648	0	0. 02201
1982	100242870	0. 00389	1. 486e – 16	0. 00037
1983	101515260	0. 00674	0	0. 0502
1984	102603690	0. 05709	0. 0100	0. 0427
1985	104719230	0. 0504	0. 0103	0. 1176
1986	107831220	0. 0344	0. 0388	0. 2079
1987	110467980	0. 0815	0. 06179	0. 203

Figure 4-1: BP , RBF, ELM prediction error comparison

According to the table, we can get the line chart as shown in the figure below

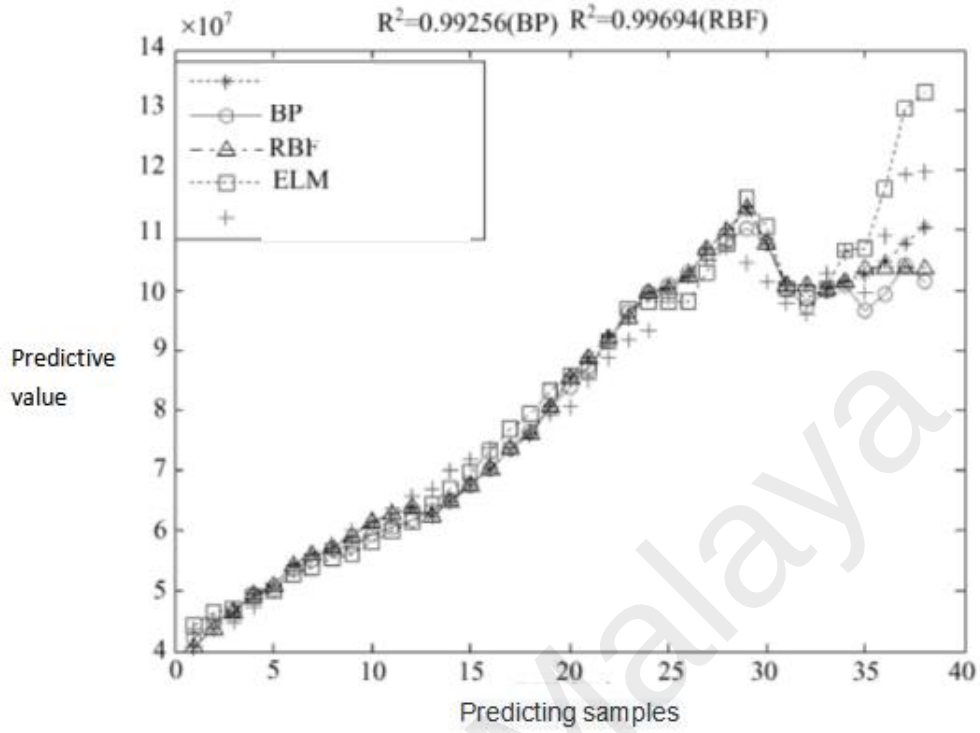

Figure 4-2:Comparison of the four predicted values

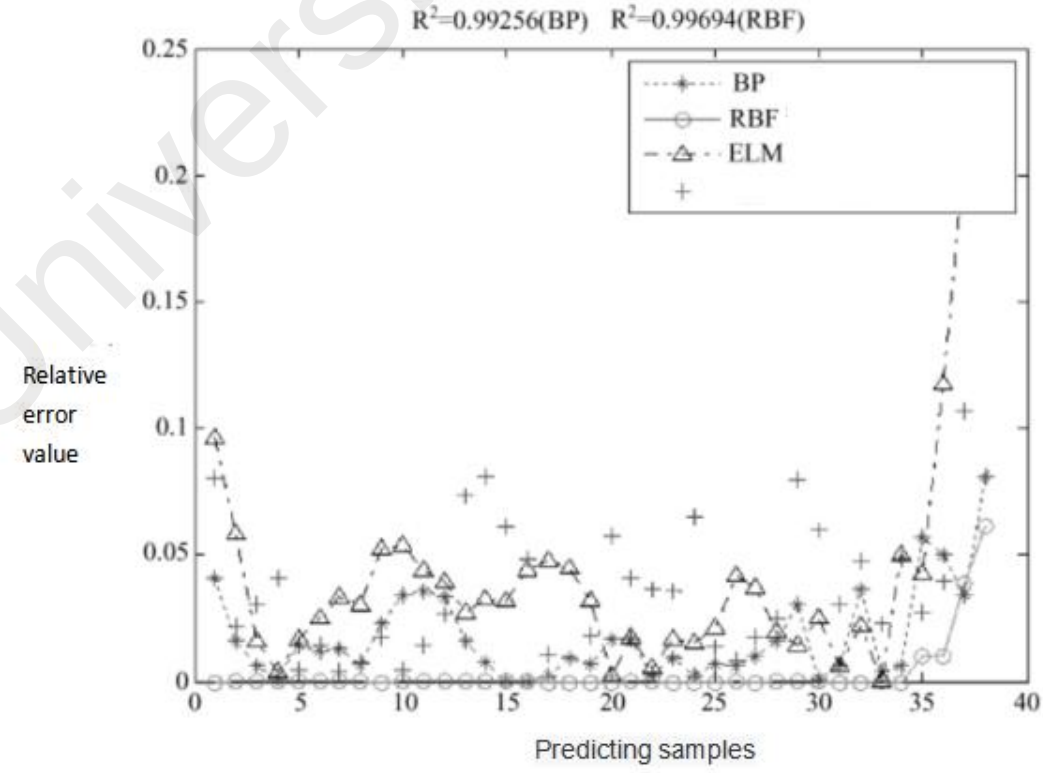

Figure 4-3 Comparison of the relative errors of the four prediction methods

Theoretically, it has been proved that RBF network can approximate any nonlinear function with any accuracy under the condition of enough hidden layer nodes and has the most general function approximation ability. In addition, it has fast convergence speed, strong anti-noise and repair capabilities. if the RBF network has enough hidden nodes and sufficient learning conditions, it can infinitely approximate any nonlinear function with arbitrary precision, and it has the ability to approximate functions. In addition, the convergence speed of RBF determines the strength of anti-noise and repair capabilities. The faster the convergence, the stronger the anti-noise and repair capabilities. In terms of reality and theory. Both the RBF function and the BP neural network can use any precision to approach any non-linear function, but the only difference is the excitation function used, so the performance of the approximation is also different. Some scholars have proved that RBF is the approximate best in continuous function, but BP network is not. Another point worth noting is that the training time required by the BP network is very long. So it is easy to fall into a local optimum. Moreover, a node in the hidden layer of the BP neural network is dependent on experiments, empirical experiments and experimental calculations, so it is difficult to obtain the optimum. However, the RBF network overcomes the above shortcomings. The learning and training speed of the local activation function is thousands of times higher than that of the BP neural network. And the number of nodes in the hidden layer has been determined during training, so its convergence is easier to guarantee than BP, so the global optimal solution can be obtained.

CHAPTER 5: CONCLUSIONS

In this project, three deep learning methods, namely BP neural network, RBF radial network and ELM extreme learning machine, are used to predict the octane number in gasoline blends. The findings of this research project are:

1) After data fitting and testing, RBF neural network has higher prediction accuracy than other algorithms, RBF neural network has the best prediction effect in terms of prediction accuracy, followed by BP neural network and stepwise linear regression. ELM extreme learning machine has the worst prediction effect..

2) Theoretically, ELM extreme learning machine network should have better prediction effect on BP or RBF, but as mentioned above, ELM network is strongly dependent on the number of hidden layer nodes, and the closer it is to the number of training samples, the better. However, only 6 hidden layer nodes are selected here, so the prediction effect is not as good as RBF network with the same number of hidden nodes.

3) On the other hand, it can be found from the prediction graph of the simulation just above that the prediction result produced by using RBF neural network is obviously better than that of the other two kinds of fitting.

This study is only based on a typical case of violation of multiple linear regression. Although the neural network method has achieved a satisfactory application effect in this case, the universality of this research result remains to be further discussed, it is precisely because of the shortcomings of the traditional methods to produce a variety of intelligent bionic algorithms, and its application is far better than the traditional method. I believe that with the deepening of research and application, intelligent algorithm will get a greater and wider space for development and application.

After that, will continue to carry out for more research in neural networks for predicting, my current work is far to meet the needs in the development of neural network now, in the future I will continue to study more different neural network in the prediction of the performance, then acquire a more powerful and more convincing results.

REFERENCES

Bradley, P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7), 1145-1159.

He Qing, Li Ning, Luo Wenjuan, & Shi Zhongzhi. (2014). Overview of machine learning algorithms under big data.Pattern Recognition and Artificial Intelligence, 27(004), 327-336.

Mitchell, T. M. . (2003). Machine Learning. McGraw-Hill.

Quinlan, J. R. . (1993). Program for machine learning.

Blum, A., L. , Langley, & P. (1997). Selection of relevant features and examples in machine learning. ARTIFICIAL INTELLIGENCE -AMSTERDAM-ELSEVIER-.

Bi Hua, Liang Hongli, & Wang Jue. (2009). Resampling Method and Machine Learning.Chinese Journal of Computer Science (05), 18-33.

Zhu J, & Hu WB. (2015). A review of the frontiers of Bayesian machine learning.Computer Research and Development, 52(1), 16-26.

Wang Fei, Yang Shengtian, Ding Jianli, Wei Yang, Ge Xiangyu, & Liang Jing. (2018).Environmental Sensitive Variable Optimization and Machine Learning Algorithm for Prediction of Soil Salinity in Oasis.Transactions of the Chinese Society of Agricultural Engineering, 34(22), 102-110.

Cai WH. (2018). Prediction of ozone concentration in air using machine learning method.China Environmental Management, 10(2), 78-84.

Li Helong, Yu Haibo, Wang Chunyu, Liu Jia, Wang Xingyuan, & Yuan Jinshuai et al. (2020). Fault Prediction Method and System of Intelligent Electricity Meter Based on Machine Learning. CN110794360A.

Zhou S Y, Luo W, Huang B. (2020). Discussion on the prediction method of coal and gas outburst strength by machine learning.Shandong Coal Science and Technology, 240(08), 177-180+184.

Qi Chong, Chen Qiusong, Zhang Qinli, Tang Xiaolin, Wang Hao, & Feng Yan, et al. (2019). A Method for Predicting Strength Parameters of Filling Materials Using Machine Learning. CN109523069A.

Chen K. (2018). A method for predicting bacterial outer membrane proteins based on machine learning. CN108009405A.

Buchwalow, I. B., and Böcker, W. (2010). *Immunohistochemistry: basics and methods.* Berlin: Springer Verlag.

Jordan, M. I. , & Mitchell, T. M. (2015). Machine learning: trends, perspectives, and prospects. Science, 349(6245), 255-260.

Kellermann, K. I. , EB Fomalont, Mainieri, V. , Padovani, P. , Rosati, P. , & Shaver, P. , et al. (2008). Temperature Compensation Modeling of Infrared Methane Detector Based on BP and RBF Neural Network. Proceedings of the Second International Symposium on Test Automation & Instrumentation(Vol.4).

Kellermann, K. I. , EB Fomalont, Mainieri, V. , Padovani, P. , Rosati, P. & Shaver, P. , et al. (2008). Temperature Compensation Modeling of Infrared Methane Detector Based on BP and RBF Neural Network. Proceedings of the Second International Symposium on Test Automation & Instrumentation(Vol.4).
Kellermann, K. I. , EB Fomalont, Mainieri, V. , Padovani, P. , Rosati, P. , & Shaver, P. , et al. (2008). Temperature Compensation Modeling of Infrared Methane Detector Based on BP and RBF Neural Network. Proceedings of the Second International Symposium on Test Automation & Instrumentation(Vol.4).

Zhang, X. R. , Fang, C. L. , Wang, Z. B. , & Hai-Tao, M. A. . (2013). Prediction of urban built-up area based on rbf neural network——comparative analysis with bp neural network and linear regression. Resources and Environment in the Yangtze Basin.

Zhu, W.2014). Constrained Extreme Learning Machine: A novel highly discriminative random feedforward neural network. International Joint Conference on Neural Networks. IEEE.

ZCK Siew. (2006). Extreme learning machine: theory and applications. Neurocomputing.

Huang, G. B. , & Siew, C. K. . (2012). Extreme learning machine: RBF network case. Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th. IEEE.