A Principal Component Analysis and Deep Back-Propagation Neural Network-based Approach to Gasoline Quality Prediction

Zihao Wang*
Wenzhou University, Wenzhou, Zhejiang Province, China
wangzihao411318@foxmail.com

Huawen Yang
Wenzhou University, Wenzhou, Zhejiang Province, China
632297365@qq.com

Liang Chen
Wenzhou University, Wenzhou, Zhejiang Province, China
1325467@qq.com

Wenhai Chen
Wenzhou University, Wenzhou, Zhejiang Province, China
whchen@wzu.edu.cn

ABSTRACT

This paper proposes an approach that combines principal component analysis with a Deep Back-Propagation Neural Network model to solve high-latitude prediction problems. The approach is applied to establish a product quality prediction model for gasoline refinement. The simulation results have demonstrated effectiveness of the approach.

CCS CONCEPTS

• **Theory of computation**; • **Theory and algorithms for application domains**; • **Machine learning theory**;

KEYWORDS

Principal component analysis, Deep back-propagation Neural networks, Prediction models

ACM Reference Format:
Zihao Wang*, Huawen Yang, Liang Chen, and Wenhai Chen. 2021. A Principal Component Analysis and Deep Back-Propagation Neural Network-based Approach to Gasoline Quality Prediction. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487075.3487129

1 INTRODUCTION

The idea of creating a mathematical model of neural networks or artificial neural networks was first put forward by neurophysiologist Warren Sturgis McCulloch and mathematician Walter Pitts in 1943 [1]. Since then, artificial neural networks have been applied in many fields, and some amazing achievements have been developed [2–4]. Many of the most exciting achievements [5, 6] in artificial intelligence research and applications were triggered in the late 1990s when deep neural networks (DNN) were introduced, including the AlphaGo network, which has since beaten many elite international and professional human Go players, and multiple automatic driving programs.

Principal component analysis (PCA) allows the reduction of the dimensionality of a data set while maintaining the characteristics of the large variance contributions within that data set [6, 7]. It is an extensively used data dimensionality reduction method that can be applied to machine learning; it thus has extensive applications in text processing, image recognition, natural language processing, and similar fields [8]. This is useful, as it is not uncommon for machine learning programs to otherwise be required to process thousands or even hundreds of thousands of dimensions.

This article proposes combining PCA and DNN to reduce the dimensionality of data in a specific use case so as to reduce resource consumption and optimize the ensuing machine learning. The approach developed is then used to establish a product quality prediction model for gasoline refinement.

The remainder of this paper is organized as follows: Section 2 introduces the theoretical algorithms used in the model, while in Section 3, a set of comparative experiments illustrating how the model can be applied to specific examples to verify its effectiveness is presented. In the last section, the overall performance and future development of the model are discussed.

2 METHODOLOGY

2.1 Principal Component Analysis (PCA)

Principal component analysis (PCA) is used to reduce the dimensionality of data sets. Specifically, given a set of n data samples$(x_1, x_2, \ldots, x_n)^T$, where

$x_k = (x_{k1}, x_{k2}, \ldots, x_{kp})$, for$k = 1, 2, \ldots, n$,

the dimensionality reduction process works as follows:

The sample data is first standardized by subtraction of the set mean from each dimension of the data, which normalizes each data sample, converting the data sample set into the form

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1p} \\ X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{np} \end{bmatrix} = (X_1, X_2, \cdot\cdot\cdot, X_p)$$

where

$$X_{ij} = \frac{x_{ij} - \bar{X}_j}{\delta_j},$$

$$\bar{X}_j = \frac{1}{n} \sum_{k=1}^{n} x_{kj}$$

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487129

and

$$\delta_j^2 = \sqrt{\frac{\sum_{i=1}^{n} (x_{ij} - \bar{X}_j)^2}{n-1}},$$

for

$$i = 1, 2, \ldots, n; j = 1, 2, \ldots, p.$$

The covariance matrix of X can then be calculated as

<?TeX

$$Cov = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix}$$

?>

where

$$r_{ij} = \frac{1}{n-1} \sum_{k=1}^{k=n} (X_{ki} - \overline{X_i})(X_{kj} - \overline{X_j}) = \sum_{k=1}^{k=n} X_{ki} X_{kj}$$

This allows the eigenvalues and corresponding eigenvectors of the matrix Cov to be calculated, assuming eigenvalues: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$, and the corresponding eigenvectors

$$a_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{p1} \end{bmatrix}, a_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{p2} \end{bmatrix}, \cdot s, a_p = \begin{bmatrix} a_{1p} \\ a_{2p} \\ \vdots \\ a_{pp} \end{bmatrix}$$

The set $V_1, V_2, \cdot s V_t$ is selected to represent the principal eigenvectors where t is the minimal number such that $\frac{\sum_{i=1}^{t} \lambda_i}{\sum_{i=1}^{p} \lambda_i}$ exceeds the selected value, which is often in the region of 98%. This set of principle eigenvectors constitutes the orthogonal basis of the ensuing dimensionally reduced space.

2.2 Back-Propagation Neural Networks (BP Neural Networks)

A BP neural network is a multi-layer feedforward neural network that utilizes a back-propagation algorithm. The main idea underlying BP networks is to divide the training process into two stages. The first stage is a forward propagation process during which input information is processed and calculated layer by layer for all hidden layers, with the actual output values being given in the output layer. The second stage is the back-propagation process, during which the difference between the actual output and the expected output is calculated recursively for each layer, and the weights of the connections are adjusted accordingly.

A structural diagram of a BP neural network is shown in Figure 1 The first layer is the input layer node, while the middle or hidden layer is formed of one or more layers of nodes, and the last layer consists of a given number of output layer nodes. When propagating in the forward direction, information is processed from the input layer through the hidden layer, and finally relayed to the output layer. The state of each layer of neurons thus only affects the state of the next layer of neurons. However, if the information does not generate the desired output in the output layer, it then propagates back, returning an error signal along the original path. By means of such iteration, the weight value of each neuron connection can be gradually modified until the signal error reaches a preset expected value.

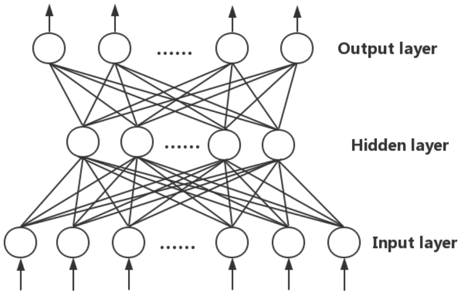


Figure 1: BP Neural Network.

3 EXPERIMENT

The octane number (expressed in RON) is the most important indicator of combustion performance in gasoline samples. It is also used as the commercial brand name of a given gasoline (for example, 89#, 92#, and 95#). In recent years, the desulfurization of FCC gasoline and the application of olefin reduction processes have produced a general reduction in the octane numbers of gasolines.

Sinopec Gaoqiao Petrochemical's catalytic cracking gasoline refining and desulfurization unit has now been in operation for four years, and it has generated a large database of information on gasoline properties. Data from this set for the period April 2017 to May 2020 was used in this experiment.

The experiment was based on original data from 325 samples, each with seven raw material property variables, two spent adsorbent property variables, two regenerated adsorbent property variables, and 354 operating values, as collected from the refining unit. This data was first preprocessed, then projected into a lower-dimensional space as generated via a PCA algorithm. Finally, a BP neural network was applied to establish product sulfur contents based on a RON loss prediction model.

3.1 Data Pre-processing

In the original data, the variables showed normal distributions; however, the data for each set of devices demonstrated problems at certain points. For some variables, data was available for only part of the period of interest, while other variables were entirely empty of data or displayed other omissions. The raw data thus required pre-processing, which was done by removing all outliers based on applying a3σ criterion.

This 3σ criterion assumed that n data points were obtained in the form $x_1, x_2, \cdot \cdot \cdot, x_n$,all with the same accuracy, with an arithmetic mean x, residual errors $v_i = x_i - x (i = 1, 2, \cdot \cdot \cdot, n)$, and a standard deviationσ. Where the residual error $v_b$of a measured value$x_b (1 \leq b \leq n)$ meets the requirement$|v_b| = |x_b - x| < 3\sigma, x_b$ must contain a bulky error value, which indicates that it is contaminated data that therefore should be removed. The standard deviation,σ, was

Table 1: Principal Component Information

Principal Component	Eigenvalues	Contribution Rate	Cumulative Contribution Rate
F_1	74.2117	0.3041	0.3041
F_2	25.0809	0.1028	0.4069
F_3	17.7093	0.0726	0.4795
F_4	13.9378	0.0571	0.5366
F_5	10.3694	0.0425	0.5791
F_6	8.323	0.0341	0.6132
F_7	7.0832	0.029	0.6423
F_8	6.2133	0.0255	0.6677
F_9	4.8397	0.0198	0.6876
F_{10}	4.5751	0.0188	0.7063
F_{11}	3.755	0.0154	0.7217
F_{12}	3.5205	0.0144	0.7361
F_{13}	3.423	0.014	0.7502
F_{14}	3.0328	0.0124	0.7626
F_{15}	2.9376	0.012	0.7746
F_{16}	2.6499	0.0109	0.7855
F_{17}	2.4383	0.01	0.7955
F_{18}	2.3455	0.0096	0.8051
F_{19}	2.2647	0.0093	0.8144
F_{20}	1.9593	0.008	0.8224
F_{21}	1.8541	0.0076	0.83
F_{22}	1.6863	0.0069	0.8369
F_{23}	1.6712	0.0068	0.8438
F_{24}	1.5458	0.0063	0.8501
......	......	......	......

calculated using the Bessel formula:

$$\sigma = \left[\frac{1}{n-1} \sum_{i=1}^{n} v_i^2 \right]^{1/2}$$

3.2 PCA Model

A PCA model was applied to the data obtained after pre-processing, which featured over 354 operation variables.

Using the PCA algorithm in MATLAB, the eigenvalues, contribution rates, and cumulative contribution rates of the corresponding principal components were then obtained, as shown in Table 1

The first$t$principal eigenvectors where the cumulated contribution exceeded a certain value $R\%$ were selected. This involved identifying$t$, the minimal number meeting the requirement$\frac{\sum_{i=1}^{t} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \geq R\%$.

For this experiment, the value was set as$R$=85%. Accordingly, the number of principal components t=24, as denoted by $F_1, F_2, \cdots, F_{24}$, with these principal components containing more than 85% of the information within the original data. Each of the original 325 sample data points contained seven original attribute variables, two spent adsorbent property variables, two regenerated adsorbent property variables, and 354 operating variables. With the 24 eigenvectors acting as an orthogonal basis for the required dimensionally reduced space, the 354 operating variables sample can thus be represented a vector of 24 dimensions.

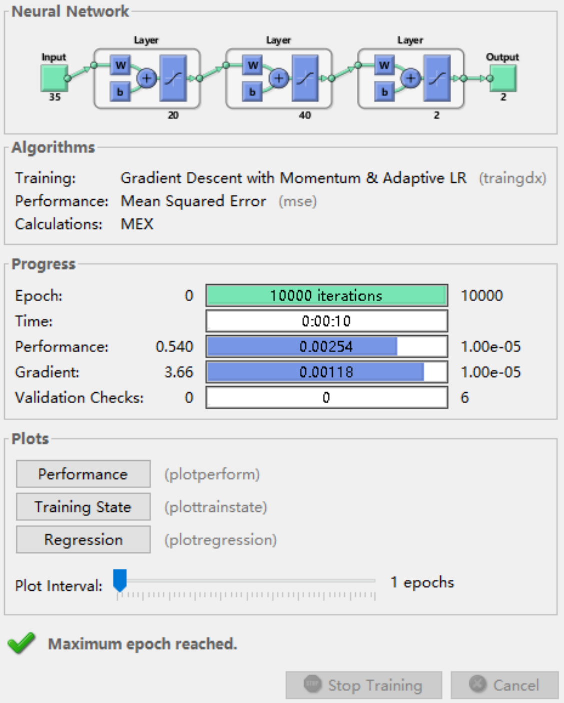
3.3 BP Neural Network Model

The BP neural network model, available in the MATLAB neural network toolbox, was then utilized for training and testing.

The input layer of the BP neural network model in this experiment contained 24 main operating variables that representing the 354 operating variables, seven raw material property variables, two spent adsorbent property variables, and two regenerated adsorbent property variables.

Figure 2 shows that the number of nodes in the input layer was thus 35. The sulfur content and octane number (RON) loss-related data was set as the objectives, to represent product performance, and the number of nodes in the output layer was set to two. The model training time only took 10 seconds.

As shown in Figures 2-4, this experiment shows that a BP neural network has the advantage of a strong nonlinear mapping ability that can approximate any nonlinear continuous function with arbitrary precision. In addition, it has a high self-learning ability and high adaptive ability, as well as several other advantages. However, the BP neural network algorithm also has some shortcomings, such as slow convergence. As the BP neural network algorithm is fundamentally a gradient descent, when the objective function is too complicated, particularly when there are too many variables in the objective function, a "sawtooth phenomenon" appears which reduces the efficiency of the BP algorithm. This experiment thus used the 24 main variables derived from the PCA model, rather than the original 354 operating variables, as input layer nodes for



Algorithms

Training: Gradient Descent with Momentum & Adaptive LR (traingdx)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	10000 iterations	10000
Time:		0:00:10	
Performance:	0.540	0.00254	1.00e-05
Gradient:	3.66	0.00118	1.00e-05
Validation Checks:	0	0	6

Plot Interval: 1 epochs

Maximum epoch reached.

Figure 2: BP Neural Network Training Diagram.

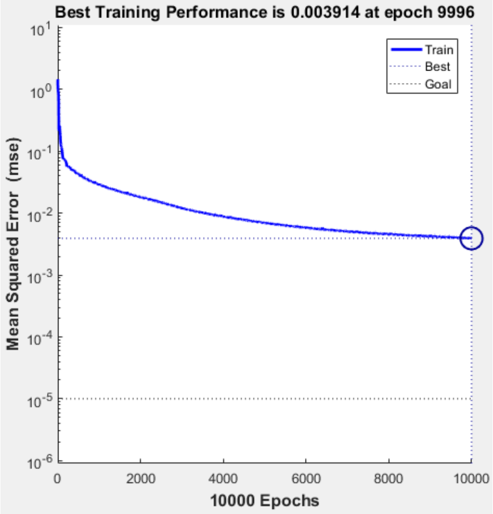


Figure 3: BP Neural Network Training Performance Graph.

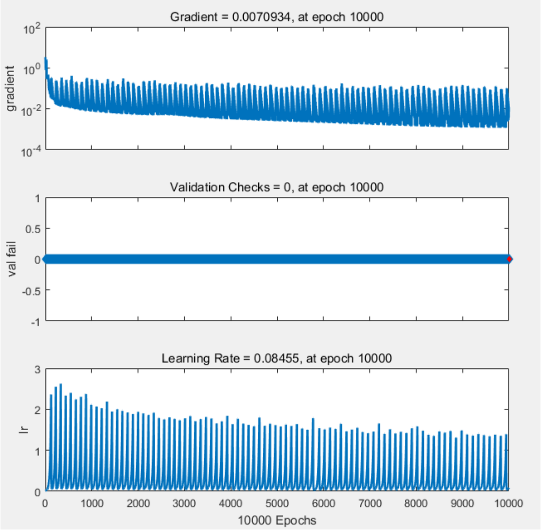


Figure 4: BP Neural Network Training State Diagram.

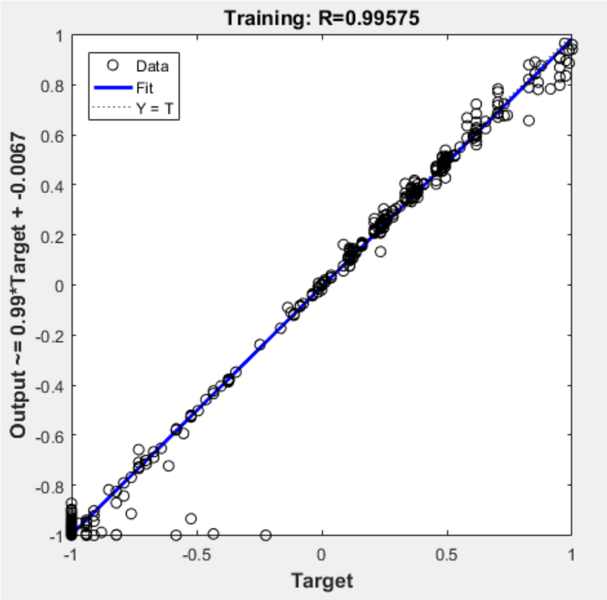


Figure 5: BP Neural Network Training Regression.

the BP neural network prediction model, which greatly reduced the complexity of the BP algorithm objective function, allowing the BP algorithm to be more efficient.

As shown in Figures 5 , the BP neural network model error is consistent with established standards, based on the results of this experiment. Figure 6 shows that the main operating variables derived from the PCA algorithm model can be sorted into new sample data, and that the results obtained by using the BP neural network prediction model on new sample data are satisfactory.

3.4 BP Neural Network Comparison Experiment

The input layer of the BP neural network model in the comparison experiment contained the 354 original manipulated variables, seven raw material property variables, two spent adsorbent property variables, and two regenerated adsorbent property variables.

Figure 7 shows that the number of input layer nodes in the BP neural network comparison experiment was 365, and the number of nodes

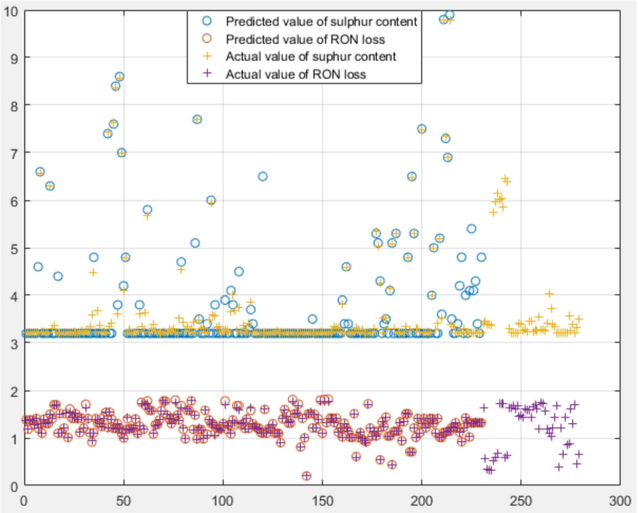
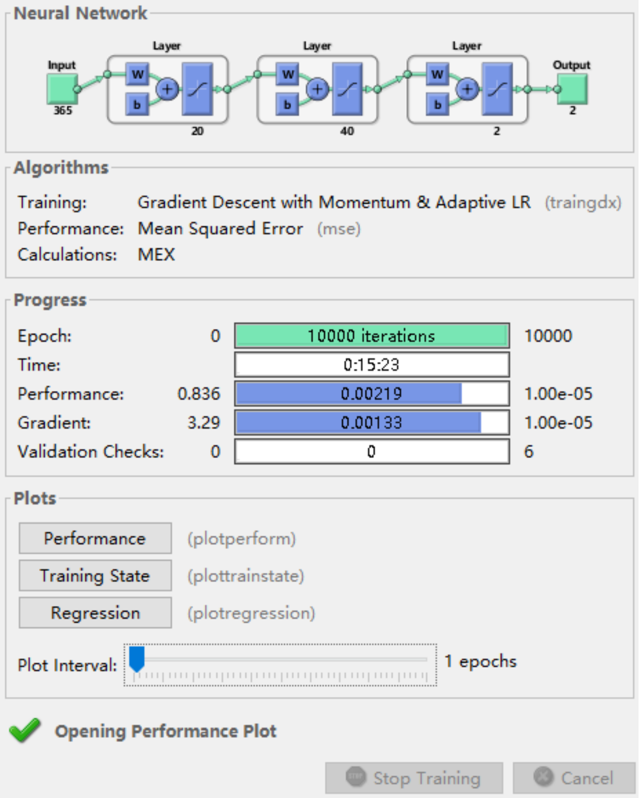


Figure 6: BP Neural Network Prediction Model Output Graph.



Algorithms

Training: Gradient Descent with Momentum & Adaptive LR (traingdx)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

```
Epoch:                0  10000 iterations  10000
Time:                        0:15:23
Performance:      0.836      0.00219       1.00e-05
Gradient:          3.29      0.00133       1.00e-05
Validation Checks:    0          0         6
```

Plots

Plot Interval: 1 epochs

Opening Performance Plot

Figure 7: BP Neural Network Comparison Experiment Training Diagram.

in the output layer was again set to two. Under the condition that other model parameters remained unchanged, the model training time was 15 minutes and 23 seconds. Replacing the original sample data with dimensionality-reduced sample data as the input of the BP neural network model can reduce the training time from 15 minutes to 10 seconds.

Through comparative experiments, it can be concluded that the dimensionality of the data directly affected the complexity of the resulting machine learning algorithm, and that the complexity of the ensuing machine learning was exponentially related to the dimensionality of the data.

In the example used in this experiment, the original sample data only had 365 dimensions, which is relatively uncomplicated. Machine learning often encounters situations requiring the processing of tens of thousands or even hundreds of thousands of high-dimensional points of data. In such cases, resource consumption in machine learning can be huge, and dimensionality reduction of the data is thus necessary to reduce resource consumption while retaining the bulk of the information.

4 CONCLUSION AND FUTURE WORK

The experiment showed that an approach that combines a PCA algorithm and a BP neural network model can be highly effective in solving high-latitude prediction problems. The PCA algorithm can be used to effectively reduce the dimensionality of a data set while maintaining the characteristics of large variance contribution in that data set. In this experiment, as shown in Table 1, the PCA algorithm model was successfully used to extract data from the 24 main variables that retained more than 85% of the information in the original data from 354 operating variables, effectively reducing the dimensionality of the sample data. The combination of the PCA algorithm and the BP neural network thus greatly improved the efficiency of the latter, reducing resource consumption during machine learning.

In the future, this predictive model could be applied in various fields to solve problems. In addition, it may be worth attempting to establish an optimization model based on the prediction model. Using the experiment discussed in this article as an example, this could facilitate the establishment of an optimization model based on prediction results that could identify the optimal operating variables to allow a product to meet or exceed the standards required in the process of gasoline refining at a lower cost.

REFERENCES

[1] Warren S. McCulloch and Walter Pitts. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), pp. 115-133.

[2] Rosenblatt F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), pp. 386-408.

[3] Minsky M, Papert S. Perceptions. Oxford: MIT Press, 1969:57-89.

[4] Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. Nature 323, 533–536 (1986).

[5] Geoffrey Hinton *et al.* (2006). Unsupervised Discovery of Nonlinear Structure Using Contrastive Backpropagation. Cognitive Science, 30(4), pp. 725-731.

[6] Haijian Wu. (2003). The basic idea and application examples of principal component analysis. Situation and statistics of Henan province, pp. 30-31.

[7] Ruiyou Li *et al.* (2020). BP neural network and improved differential evolution for transient electromagnetic inversion. Computers and Geosciences, 137.

[8] Lee Loong Chuen and Jemain Abdul Aziz. (2021). On overview of PCA application strategy in processing high dimensionality forensic data. Microchemical Journal, 169.