

Journal of Engineering Science and Technology Review 7 (3) (2014) 158-163

JOURNAL OF Engineering Science and Technology Review

Research Article

www.jestr.org

Estimation of Biodiesel Yield using Fast Decorrelated Neural Network Ensemble

Hongyan Shi^{1,2,*}, Bin Li¹, Xiu Wang¹, Ying Wang¹ and Dresnutsky Wong³

¹Dept. of Information Engineering, Shenyang University of Chemical Technology, Shenyang 110142, Liaoning, China ²Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, Liaoning, China ³Riga Vocational College of Science&Technology, Galuge Str.1001, Riga LV-1918, Latvia

Received 1 February 2014; Accepted 19 July 2014

Abstract

The accurate and reliable online measurement of the product yield is very essential for the control and optimization of the biodiesel process. A biodiesel yield prediction model based on the fast decorrelated neural network ensembles (FDNNE) was established to enhance the estimated performance. The random vector functional link (RVFL) networks were inserted into the fast decorrelated neural network ensemble frame as the base model since it could provide better generalized performance and faster speed. The FDNNE product yield prediction model initializes the hidden layer parameters of base models randomly, and calculates the output layer parameters using the least square method with negative correlation learning. Simulation results show that the proposed method has relatively higher accuracy and reliability compared with the single RVFL model.

Keywords: Biodiesel Yield, Negative Correlation Learning, Fast Decorrelated Neural Network Ensembles, RVFL

1. Introduction

Due to the continuous decline of limited petroleum reserves and the growing environmental concerns, the use of biodiesel in recent years as a fuel in the existing diesel engines has gained much importance [1], [2], [3]. Although there are many researches on the synthesis technology of biodiesel, on the experimental data processing of biodiesel yield and analysis of the interaction between factors is still less.

In the biodiesel reaction, the yield as main quality index is affected by different parameters [4]. However, four main factors have been considered by researchers as follows: molar ratio between alcohol and oil, reaction time, catalyst, and reaction temperature [5], [6]. Magnetic resonance detection method is often used to measure the yield of biodiesel while on-line hardware sensors are difficult to meet the requirements of on-site monitoring and control due to big investments, poor reliability, and long response time.

Recently, ensemble methods have received considerable attentions because it can effectively improve the validity and credibility of the regression model through building a set of models [7], [8], [9]. Many novel machine learning techniques are proposed to train an ensemble model individually or collectively such as bagging, boosting and random forests [10], [11], [12], [13]. Alhamdoosh and Wang [14] proposed an efficient algorithm to build ensemble models in a very short time named fast decorrelated neural network ensembles (FDNNE) which employed the random vector functional link (RVFL) networks as base components

and incorporated with the NCL strategy for building neural network ensembles. Negative correlation learning (NCL) [15] is an algorithm for training neural network ensemble with sound generalization capability through controlling the disagreement of base model.

In this paper, a fast biodiesel yield ensemble model using fast decorrelated neural network ensemble was established, which randomly initialized the hidden layer parameters of base RVFL networks, and then employed the least square method with negative correlation learning scheme to analytically calculate the output weights of these base networks. The proposed method was then verified by the experimental data of biodiesel product. The experimental results of the testing datasets demonstrated that the production yield ensemble model had the capacity of strong generalization and low computation load.

This paper is organized as follows: section 2 presents basics on RVFL neural networks, negative correlation learning and fast decorrelated neural network ensembles, section 3 focuses on the analysis of experiment results and section 4 presents the conclusions.

2. Biodiesel yield ensemble model based on FDNNE algorithm

2.1 The Description of the Biodiesel Process

The experimental device used for continuous process of biodiesel synthesis under supercritical methanol is shown in Fig.1. Soybean oil was placed in vessel and heated to $35 \sim 50^{\circ}$ C, the solid catalyst was packed into fix bed reaction tubes using pressurized gas analyzer. The oil and methanol were pumped into the pipeline by a high pressure pump,

^{*} E-mail address: shihongyan@sia.cn ISSN: 1791-2377 © 2014 Kavala Institute of Technology. All rights reserved.

under the conditions of a total volume flow rate of $1 \sim 4$ ml/min, after mixed in the pipeline into the reaction tube. The reaction temperature, pressure and the molar ratio of methanol to oil were controlled accurately by the reaction apparatus. The reactor temperature could be controlled between 260 ~ 400°C and the inter pressure was regulated between 13~24MPa by back pressure valve. The product effluent from the reaction tube was sampled after condenser pipe condensate.



Fig. 1. Biodiesel Supercritical Synthesis Device

2.2 RVFL Neural Network

For *N* arbitrary distinct samples $(\mathbf{x}_i, \mathbf{y}_i), i = 1, ..., N$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \mathbb{R}^n$, $\mathbf{y}_i = [y_{i1}, y_{i2}, ..., y_{im}]^T \in \mathbb{R}^m$, RVFL neural network as single-layer feed-forward networks (SLFNs) is mathematically defined as:

$$f(x;\beta) = \sum_{k=1}^{L} \beta_k G(\omega_k^T \cdot x + b_k)$$
(1)

where $x \in \mathbb{R}^d$ is input vector, $\omega_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$ are input weights and hidden layer biases, $\beta = [\beta_1, \beta_2, \dots, \beta_L] \in \mathbb{R}^L$ is the output layer weights, d is the number of input layer, $G(\cdot)$ is the basis function, $f(\cdot)$ is the output of neural network. In RVFL network, the parameters of the hidden layer are assigned randomly and independently of the training data; while the linear parameters β_k of the output layer can be tuned using the least squares method.

Testing dataset with size N' can be denoted by $D_t = \{(x_1, y_1), (x_2, y_2)...(x_{N'}, y_{N'})\}$, the generalization error $E_f(D_t)$ of the RVFL network model is defined as the mean squares error (MSE) averaged over all possible realizations of D_t

$$E_{f}(D_{t}) = \sqrt{\left(\frac{1}{N'}\sum_{i=1}^{N'} f(x_{i}) - y_{i}\right)^{2}}$$
(2)

where $f(x_i)$ is the predictive value of sample (x_i, y_i) , y_i is the expected output of x_i .

2.3 Negative Correlation Learning

Negative correlation learning (NCL) is proposed to reduce the covariance among ensemble individuals while the variance and bias terms are not increased. Unlike traditional ensemble learning approaches, NCL is introduced to train base models simultaneously in a cooperative manner that decorrelates individual errors e_i [16]. Mathematically, the learning error of the *i* th sub-model given in Eq. (3) is modified to include a penalty term p_i as follows:

$$e_{i} = \sum_{n=1}^{N} \left[\frac{1}{2} (f_{i}(x_{n}) - y_{n})^{2} + \lambda p_{i}(x_{n}) \right]$$
(3)

Where $\lambda \in [0,1]$ is a regularizing factor. Set a training dataset D_t with size N, $D_t = \{(x_1, y_1), (x_2, y_2)...(x_N, y_N)\}$, where x_i is the input of the neural network, y_i is the expected output of x_i . The important steps of negative correlation learning list as follows:

Step1: Calculating the output of each sub-model $f_i(\cdot)$, and then compute the output of the whole ensemble $\overline{f}(x_n)$

$$\overline{f}(x_n) = \frac{1}{M} \sum_{i=1}^M f_i(x_n) \tag{4}$$

where M is the number of sub-model.

Step2: Using the output of sub-model and the output of whole model to decide the penalty term p_i

$$p_{i} = (f_{i}(x_{n}) - \overline{f}(x_{n})) \sum_{j \neq i} (f_{j}(x_{n}) - \overline{f}(x_{n}))$$
(5)

Step3: Adding the penalty term p_i into the error function, so the learning error of the *i*th sub- model could be defined as follows:

$$e_{i} = \sum_{n=1}^{N} \left[\frac{1}{2} (f_{i}(x_{n}) - y_{n})^{2} + \lambda p_{i}(x_{n}) \right]$$
(6)

Step4: Applying the new learning error to the update the weights.

2.4 Fast Decorrelated Neural Network Ensembles

Due to the uncertainties in the learning process that are caused by the random initializations of basis functions, a single RVFL network can not guarantee the accuracy of the forecasts. A new ensemble learning approach, which uses RVFL networks as ensemble component is adopted and it is fitted in negative correlation learning framework. Since RVFL networks are used to populate the ensemble, the output of the *i* th base network is stimulated with an instance X_n , which is given by

$$f_i(x_n) = \sum_{j=1}^{L} \beta_{ij} g_{ij}(x_n)$$
(7)

where *L* is the number of hidden neurons in the *i* th individual RVFL network, β_{ij} is the output weight connecting the *j* th hidden neuron with the output neuron in the *i* th base model, $g_{ij}(x_n) = (w_j, b_j, x_n)$ is the output of the *j* th hidden neuron in the *i* th base model, and *G* can be any squashing basis function. The parameters (w_j, b_j) of the basis functions g_{ij} are randomly set while the only parameters to be tuned are the output weights β_{ij} . In order to get the best performance of model, $\nabla e_i = 0$, for i = 1, ..., M should be satisfied, which leads to

$$\frac{\partial e_i}{\partial \beta_{ij}} = \sum_{n=1}^{N} \frac{\partial e_i(x_n)}{\partial \beta_{ij}} = 0 \quad \text{for } j = 1, \dots L$$
(8)

All base networks are assumed to have similar architecture and the same dataset is used to train all of them. It can get that

$$\sum_{k=1}^{L} C_{1}\varphi(i,j,i,k)\beta_{ik} + \sum_{\substack{l=1\\l\neq i}}^{M} \sum_{k=1}^{L} C_{2}\varphi(i,j,l,k)\beta_{lk} = \varphi(i,j)$$
(9)

where

$$C_1 = 1 - 2\lambda \frac{(M-1)^2}{M^2}$$
(10)

$$C_2 = 2\lambda \frac{M-1}{M^2} \tag{11}$$

$$\varphi(i, j, i, k) = \sum_{n=1}^{N} g_{ij}(x_n) g_{kl}(x_n)$$
(12)

$$\varphi(i, j) = \sum_{n=1}^{N} g_{ij}(x_n) y_n$$
(13)
 $i = 1, ..., M, j = 1, ...L$

where *M* is the number of sub-model, *L* is the number of hidden neuron, λ is a regularizing factor, and C_1, C_2 are two constants. $\varphi(i, j, l, k)$ represents the correlation between the *j* th hidden neuron of the *i* th individual RVFL network

and the *l* th hidden neuron of the *k* th individual RVFL network, and $\varphi(i, j)$ represents the correlation between the

j th hidden neuron of the i th base network and the target value. To facilitate this computational task, this linear system in a matrix form can be designed by equation (14).

$$H_{corr}B_{ens} = T_h \tag{14}$$

where H_{corr} is called the hidden correlation matrix, B_{ens} is the global output weights matrix and T_h is the hidden-target matrix. H_{corr} is defined by equation (15) or (16).

$$H_{corr}(p,q) = \begin{cases} C_1 \varphi(m,n,k,l) & if \quad m=k\\ C_2 \varphi(m,n,k,l) & otherwise \end{cases}$$
(16)

where

$$m = \left[\frac{p}{L}\right], \quad k = \left[\frac{q}{L}\right] \tag{17}$$

$$n = ((p-1) \mod L) + 1$$
 (18)

$$l = ((q-1) \mod L) + 1 \tag{19}$$
$$p, q = 1, \dots, M \times L$$

And mod is the modulo operation. B_{ens} and T_h are defined as follows:

$$B_{ens} = \left[\beta_{11} \cdots \beta_{1L} \ \beta_{21} \cdots \beta_{2L} \cdots \right]_{ML}$$

$$\beta_{M1} \cdots \beta_{ML} \int_{ML}^{T} \beta_{MLM}$$
(20)

$$T_{h} = [\varphi(1,1) \cdots \varphi(1,L) \ \varphi(2,1) \cdots \varphi(2,L) \cdots \varphi(M,1) \cdots \varphi(M,L)]^{T}_{ML \times 1}$$

$$(21)$$

The output weights β_{ij} can be got by equation (22)

$$B_{ens} = H_{corr}^{-1} T_h \tag{22}$$

Then the output of i th base network, simulated with an sample x_n can be got by equation (7). Uniform averaging weights are used to combine ensemble base RVFL network, and the output of the ensemble can be obtained as follows:

$$\overline{f}(x_n) = \frac{1}{M} \sum_{i=1}^M f_i(x_n)$$
⁽²³⁾

2.5 FDNNE Algorithm

Require: Training dataset D_t , a basis function $G: R \rightarrow R$; default is Sigmoid, scaling coefficient $\lambda \in [1.0]$, the size of base model L, and the number of base models M. **Ensure**: Trained DNNE model.

- 1) Initialize the architecture of M SLFNs to be the ensemble base models.
- 2) Initialize the basis functions of base models randomly i.e. w_i, b_i are randomly set.
- 3) Calculate the outputs of the hidden layers of base models for all examples in D_t i.e., calculate $g_{ii}(x_n)$
- 4) Calculate the constants C_1 and C_2 .

5) for $p \leftarrow 1$ to $M \times L$ do 6) $m \leftarrow \left[\frac{p}{L}\right]$ 7) $n \leftarrow ((p-1) \mod L) + 1$ for $q \leftarrow 1$ to $M \times L$ do 8) $k \leftarrow \left[\frac{q}{L}\right]$ 9) $l \leftarrow ((q-1) \mod L) + 1$ 10)11) if m = k then $H_{corr}[p,q] \leftarrow C_1 \varphi(m,n,k,l)$ 12) 13) $H_{corr}[p,q] \leftarrow C_2 \varphi(m,n,k,l)$ 14) 15) end if 16) end for 17) end for 18) $k \leftarrow 1$ 19) for $i \leftarrow 1$ to M do 20) for $j \leftarrow 1$ to L do $T_{k}[k] \leftarrow \varphi(i, j)$ 21) 22) $k \leftarrow k + 1$ 23) end for 24) end for 25) Calculate H^{-1}_{corr} , the pseudo-inverse inverse of H_{corr} . 26) Calculate the estimated global output weights matrix B_{ens} from Eq. (22).

27) for $i \leftarrow 1$ to M do

28) Output weights of base model: $i \leftarrow B_{ens}[(i-1)L+1:iL]$

29) end for

30) return Ensemble model (DNNE).

3. Results and discussion

In this section, the performance of FDNNE with single RVFL network was compared by the analysis of biodiesel dataset, the yield index was predicted by analyzing temperature, pressure and volume ratio. All the experiments were carried out in a MATLAB 7.5.0 environment running with CPU 2.55GHz and 2GB RAM.

For RVFL networks, there is only one parameter with L hidden nodes needs to be determined. For the FDNNE

network, three parameters with L hidden nodes, M submodels and the regularizing factor λ are chosen. Taking the impact that the number of hidden nodes L to the performance of the ensemble into account, a number of experiments can be conducted by changing the number of L from 20 to 110 with the interval of 10 and keeping the other parameters same. Here M is 8 and λ is 0.6, 50 times are finished for every L.

In this paper, the root mean squares error (RMSE) and correlation coefficient (R^2) are calculated to reflect the performance of the models. RMSE metric is defined by Equation (24), which represents the smaller the RMSE, the better performance of the network.

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(\frac{1}{M} \sum_{i=1}^{M} f_i(x_n) - y_n\right)^2}$$
(24)

 R^2 is calculated by Equation (25), and $R^2 \in [0,1]$. If it is closer to one, the model will have better performance.

$$R^{2} = \frac{N\sum_{i=1}^{N} f_{i}(x_{n})y_{n} - \sum_{i=1}^{N} f_{i}(x_{n})\sum_{i=1}^{N} y_{n}}{(N\sum_{i=1}^{N} f_{i}(x_{n})^{2} - (\sum_{i=1}^{N} f_{i}(x_{n})))(N\sum_{i=1}^{N} y_{n}^{2} - (\sum_{i=1}^{N} y_{n})^{2})}$$
(25)

Performance comparison of different number of hidden nodes is shown in Fig.2. Fig.3 illustrates how the number of the sub-model affects the accuracy of the model. Let M change in (2, 20) with the interval of 2 and keeping L and λ same. Not only does the number of hidden nodes impact the performance of the ensemble but also the size of the ensemble. Set L=100, λ =0.6, for every situation, it has finished 50 times, and the average RMSE and R^2 are obtained. From Fig.3, it can be concluded that the best size of our ensemble is six.



Fig.2. Performance Comparison of Different Number of Hidden Nodes



Fig.3. Performance Comparison of Different Number of Sub-Models

As shown in Figure 4, it is the comparison of the expected value and the predicted value of FDNNE. Figure 5 shows the comparison of RVFL network. In this experiments, L=100, M=6, and λ =0.6. After several experiments, it can be found that FDNNE perform better than RVFL network when the same numbers of hidden nodes are used.



Fig.4. Comparison of the Expected Value and Predicted Value of FDNNE network



Fig.5. Comparison of the Expected Value and Predicted Value of RVFL network

The average value of RMSE and R^2 of the training data and testing data after running 100 times are presented in Table 1, in which FDNNE training means that the input is the training dataset using FDNNE model, and FDNNE testing means the input is the testing dataset different from training data. It is clear that the RMSE of FDNNE model is smaller than RVFL networks model, and R^2 of FDNNE model is closer to one. The performance of FDNNE model is better than RVFL networks.

Tab.1. Comparison of testing FDNNE and single RVFL

	FDNNE training	RVFL training	FDNNE testing	RVFL testing
RMSE	0.9985	1.5101	2.07846	2.2228
\mathbb{R}^2	0.9866	0.9681	0.94916	0.94142

4. Conclusions

Although the RVFL modeling method has improved the generalized performance, it was difficult to achieve better stability. In this work, a biodiesel yield model was built using fast decorrelated neural network ensemble by building a set of models, rather than a single model.

The ensemble model employed random vector functional link (RVFL) networks as base model to initialize the hidden layer parameters of base models randomly and calculated the output analytical solution using the least square method with negative correlation learning. The experiment results show that the proposed model has a higher relatively accuracy and faster learning speed than the single RVFL model. In this study, the results are limited to the operating conditions in the laboratory and more industrial experiments should be done.

Acknowledgments

The authors would like to acknowledge the financial support provided by the National Natural Science Foundation of China (No.61203102) and Science Research Foundation of Liaoning Provincial Department of Education (No.L201214 5) & (No.L2013158).

References

- Moradi, G.R., Dehghani, S., Khosravian, F., and Arjmandzadeh, A., "The optimized operational conditions for biodiesel production from soybean oil and application of artificial neural networks for estimation of the biodiesel yield", *Renewable Energy* 50(1), 2013, pp. 915-920.
- Jeong, G.T., Yang, H.S. et al., "Optimization of transesterification of animal fat ester using response surface methodology", *Bioresourse Technology* 100(1), 2009, pp. 25-30.
- 3. Yusuf, N., Kamarudin, S., "Overview on the current trends in biodiesel production", *Energy Conversion and Management* 52(7), 2011, pp. 2741–2751.
- Sharma, Y.C., Singh B., and Upadhyay S.N., "Advancements in development and characterization of biodiesel: a review", *Fuel* 87(12), 2008, pp. 2355-2373.
- Demirbas A. "Biodiesel fuels from vegetable oils via catalytic and non-catalytic supercritical alcohol transesterifications and other methods: a survey", *Energy Conversion and Management* 44(13), 2003, pp. 2093-2109.
- 6. Banerjeea, A., Chakraborty, R., "Parametric sensitivity in transesterification of waste cooking oil for biodiesel production a review", *Resource Conservation and Recycling* 53(9), 2009, pp. 490–497.
- Kourentzes, N., Barrow, D. K., and Crone, S. F., "Neural network ensemble operators for time series forecasting", *Expert Systems* with Applications 41 (9), 2014, pp. 4235–4244.
- Niu, D. P, Wang, F. L., Zhang, L. L., He, D. K., and Jia, M. X., "Neural network ensemble modeling for nosiheptide fermentation process based on partial least squares regression", *Chemometrics* and Intelligent Laboratory Systems 105(1), 2011, pp. 125-130.

- 9. Breuer, L., Huisman, J. A., and Willems, P., "Assessing the impact of land use change on hydrology by ensemble modeling (LUCHEM). I: model intercomparison with current land use", Advances in Water Resources 32 (2), 2009, pp. 129-146.
 Breiman, L., "Bagging predictors", Machine Learning 24(2), 1996,
- pp. 123-140.
- 11. Schapire, R. E., "The strength of weak learn ability", Machine Learning 5(2), 1990, pp. 197-227.
- 12. Assaad, M., Bone, R., and Cardot, H., "A new boosting algorithm for improved time-series forecasting with recurrent neural networks", Information Fusion 9(1), 2008, pp. 41-55.
- 13. Dietterich, T. G., "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization", *Machine Learning* 40(2), 2000, pp. 139–157. 14. Alhamdoosh, M., Wang, D.H., "Fast Decorrelated Neural Network
- Ensembles with Random Weights", Information Sciences 264(4), 2014, pp. 104-117.
- 15. Liu, Y., Yao, X., "Ensemble learning via negative correlation", *Neural Networks* 12(10), 1999, pp. 1399–1404.
 Liu, Y., Yao, X., "Negatively correlated neural networks can
- produce best ensembles", Australian Journal of Intelligent Information Processing Systems 4 (3/4), 1997, pp. 176–185.