Back Propagation Artificial Neural Network and Its Application in Fault Detection of Condenser Failure in Thermo Plant

Firas B. Ismail, Vinesh Thiruchelvam
School of Engineering, Asia Pacific University, Kuala Lumpur, Malaysia
E-mails: firas@apu.edu.my, Dr.vinesh@apu.edu.my

Abstract. Steam condenser is one of the most important equipment in steam power plants. If the steam condenser trips it may lead to whole unit shutdown, which is economically burdensome. Early condenser trips monitoring is crucial to maintain normal and safe operational conditions. In the present work, artificial intelligent monitoring systems specialized in condenser outages has been proposed and coded within the MATLAB environment. The training and validation of the system has been performed using real operational measurements captured from the control system of selected steam power plant. An integrated plant data preparation scheme for condenser outages with related operational variables has been proposed. Condenser outages under consideration have been detected by developed system before the plant control system

1. Introduction

Power plant trips always cause a serious problem to the industry. It will bring a huge disaster if it is left unsolved. Usually trips may cause abnormal operation to the plant, such as air in-leakage, tube corrosion, cooling malfunction, inadequate air removal, condenser failures and so on. This will reduce the efficiency of the condenser in cooling exhaust steam [1]. Therefore if these faults could not be resolved in time it may cause the system or plant to shutdown, equipment damage and loss [2] and even disasters such as core meltdown, steam explosion and so on. Due to these trips or faults, a process of fault detection and diagnosis plays an important role in a power plant. The process of fault detection and diagnosis is to monitor the condition of machines operating and to ensure all are functioning at normal and safe levels in the power plant [3]. There are a lot of parameters and complex measuring variables that need to be monitored continuously [2]. Thus, an intelligent real time system for fault diagnosis is required. There are many methods used to process fault diagnosis and play an important role both in the industry and academic sector, and the most common method is Artificial Neural Network (ANN) [2], [4]. This method is still under development phase since there are several other techniques that can be applied in this technology [5]. The BP algorithm uses the gradient of the performance function to find out the weights that maximize performance. Back-propagation is the technique used to determine the gradient of the performance, and this technique performs computations backward through the network [5]. The architecture of BP model includes multiple layers of neurons. Each layer consists of weight matrix, bias vector, input vector and output vector. A three-layer BP model may involve different numbers of neurons in different layers. The network is able to extract higher-order statistics when there are more hidden layers.

1 To whom any correspondence should be addressed.
2. Methodology
In this work, the feed-forward back-propagation network (BP) is implemented as supervised system, so the input and a target output are provided. Some training parameters need to be considered for implementing the BP network, such as epochs, learning rate (lr), show, goal and root means squared error. The training algorithm that is used in this BP training process is “traingd” The “traingd” (Batch Gradient Descent) is a batch mode and slow training process. The weights and biases are updated according to gradient descent [3-4]. The transfer function with the weights and biases input are used to generate output. The most commonly used transfer functions of the BP network are logsig (range: 0 to 1), tansig (range +1 to -1) and purelin. The transfer function that is chosen to be applied in this BP network is logsig. It is the most commonly used for the multilayer networks to generate an output in the range 0 to 1. Even though other transfer functions can be used, but since the target output is set into the range from 0 to 1, the transfer function logsig is more suitable to be used.

2.1 Selecting of learning rate
During the configuration of the ANN training parameter, the learning rate $\eta$ must be chosen carefully [4]. Figure 1 illustrates the graph of error against the weight for large and small learning rate. According to [4], the learning rate $\eta$ should be chosen within $0 < \eta < \frac{2}{\lambda_{max}}$ where $\lambda_{max}$ is the maximum eigenvalue of the autocorrelation matrix.

$$R_x = xx^T$$  \hspace{1cm} (1)

Where x is the input matrix of the ANN

$$\det(R_x - \lambda I) = 0$$  \hspace{1cm} (2)

![Figure 1. Changes of the Weight with Small and Large $\eta$ [4]](image)

2.2 Selecting the number of hidden layers
The next step is to select the number of hidden layers. The maximum number of hidden layers should not exceed three [4]. The difference between the numbers of hidden layers can be summarized in Table 1.

| Number of Hidden Layer | Result |
|------------------------|--------|
| None                   | Only capable of representing linear separable functions. |
| 1                      | Is able to approximate arbitrarily with any systems that contains continuous mapping from one finite space to another. |
| 2                      | Is able to represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping. |
2.3 Selecting the number of hidden neurons

Once the number of hidden layer is selected, the next step is to select the number of neurons in the hidden layer. The hidden neurons in the hidden layer have large influence on the final output. If used too less, it will result in underfitting. Here, the neurons are too few to able to adequately detect the signals in a complicated system. If uses too many, the result would be overfitting. Here, the neural network has too much information processing capacity that the limited amount of information contained in the training set is not enough to train all the neurons [5]. There is no proper method of selecting the correct number of neurons in hidden layer, but with the forward selection method, we are able to obtain suitable hidden neuron number systematically. The forward selection method is summarized in figure 2.

![Figure 2. Selecting the Number of Hidden Neurons with Forward Selection Method [5]](image)

2.4 Performance Function: Root Mean Square Error

The Root Mean Square Error (RMS) is the measurement difference between the predicted values and the actual values [5].

\[
\text{RMS Errors} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}
\]  

(3)

Where the \(\hat{y}_i\) is the predicted value and \(y_i\) is the observed value.

3. Pre Process

Before all the reading can be feed as training data into the ANN system as training data and verifications, they are being pre-processed. The missing values are being treated using interpolation. The trend of parameters with missing values can be represented with a linear straight line. This will show the behaviour of each parameter from normal operation to failure.

After all missing values are being treated; they are being normalized between the ranges of 0 to 1 which is to meet the requirement length size for the ANN inputs. This is done with max-min normalization, where the value is mapped to a value between 0 and 1, where 0 represents the minimum value and 1 represents the maximum value in a range. Max-min normalization is an accurate representation of a value for use in the ANN and is thus used.

\[
\text{value}_\text{normalised} = \frac{\text{value} - \text{min}}{\text{max} - \text{min}}
\]  

(4)

4. Implementation

There are 1817 input vector with 241 conditions. The ratio for data segmentation process is 0.7(70%):0.3(30%). That’s means the training input data is divided into 70% for the training and 30% for validation testing based on 241 conditions. According to Shahin [6], the more data input was used, the smaller the root mean square error. Besides that, it is also used to ensure that the ANN can
be applicable to actual data other than the training set. At this step, we have to make sure that the verification data have a 100% success rate to tell that the system is able to handle with practical situation. Therefore, all the training input data are arranged into the dimensions of 169×1817. While for the validation testing data is 72×1817 dimensions.

5. Results and Observation
The result of using the Batch Gradient Descent is shown in figure 3. The network is able to train all 169 data successfully and the errors are reduced based on the figure shown. The errors performance decreased smoothly until it reaches the performance goal. In the beginning, the weights are in random condition and the network doesn’t “know” which direction to search. Thus, the algorithm is trying to adjust the weight in order to minimize the error. Once the algorithm had found the suitable weights, the errors performance will decrease linearly as shown in figure 3.

![Figure 3](image_url)

**Figure 3.** The plot performance of the training

6. Conclusion
BP neural network was introduced into the fault diagnosis of the condenser for coal power plant and a program was constructed. The preparation of the data was explained. “traind” training algorithm was used in the program along with the data segmentation of 0.7(70%):0.3(30%) and total of 241 conditions. 169 from the total condition for training was successful as the RMS error was lower than 0.05. The result of the training performance had explained the smooth decrease of errors and reaching to the performance goal. After 2152 epochs, the program had reached the performance goal of 0.0025. This proposed fault diagnosis method can be used as a practical application in the power plant system.

References

[1] I. Homepage, 2011. [Online]. Available: http://www.intekflow.com/technical-services/power-plant-condenser-experts/default.html.

[2] Eslamloueyan R 2011 *Applied Soft Computing* **11(1)** 1407-1415

[3] Tan S C, Lim C P, Rao M V C 2007 *Engineering Applications of Artificial Intelligence* **20(2)** 203-213

[4] Nimmo I 1995 *Chemical Engineering Progress* **91(9)**

[5] Zhong C, Ding Y, Fu J 2010 *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on* IEEE 2 414-417

[6] Shahin M A, Maier H R, Jaks M B 2000 *Evolutionary data division methods for developing artificial neural network models in geotechnical engineering* University of Adelaide, Department of Civil and Environmental Engineering