Temperature Prediction of Solar Array Vacuum Heat Test Based on Deep Belief Network

To cite this article: Hao Dong et al 2019 J. Phys.: Conf. Ser. 1168 062032

View the article online for updates and enhancements.
Temperature Prediction of Solar Array Vacuum Heat Test Based on Deep Belief Network

Hao Dong¹, Junwu Deng¹, Ziming Wang¹, Shuo Liang¹and Xinming Su¹
¹Beijing Satellite Environmental Engineering Research Institute
a Dong Hao: 920567145@qq.com

Abstract: The vacuum heat test is an indispensable test item in the development of spacecraft. In the vacuum heat test, the correct interpretation and prediction of temperature data is very difficult and very important. This paper proposes a thermal test temperature prediction method based on deep belief network. The method includes the construction of deep belief network, layer-by-layer pre-training of model parameters, and adopts the Adam algorithm to replace the traditional steepest gradient descent method to complete the supervised fine tuning process, so that it converges to the method described in this paper can effectively complete optimal solution more quickly. Finally, the case analysis is carried out with real test data. The results show that the e the thermal test temperature prediction.

1.Introduction
The importance of the solar array as part of the spacecraft power supply system is self-evident. To avoid defects in the materials and processes of the solar array, and the possibility of early failure. The solar wing of the spacecraft requires a thermal vacuum test. As one of the most important sub-systems in vacuum heat test, the measurement and control system play an important role in test preparation, test condition operation, test status monitoring and test result analysis. In the test, the front temperature point of the solar wing is predicted, which not only can refer to the actual measured value, but also can accurately apply to the external heat flow of the next cycle, thereby improve the control level of the test. However, since the thermal characteristics of the test piece and the container in the vacuum heat test are very complicated and the heating elements are various, there is no method for accurately predicting the temperature of various types of test pieces in the thermal test. Traditional neural networks have good function approximation capabilities without the need to know their transfer functions. But their networks are generally shallow-structured. When the variables of the network learning object and the observation data are large, the shallow structures often cannot meet the requirements.

In 2006, Hinton et al. introduced a high-efficiency unsupervised learning deep belief network (DBN) based on deep learning ideas. Because of its deep structure and unsupervised weight initialization strategy, it has a very fast development in nonlinear system modeling. The current DBN learning process is divided into two parts: unsupervised pre-training and supervised fine tuning. The fine tuning process is based on the error back propagation algorithm, which has the disadvantages of slow learning speed and easy to fall into local optimum, which affects learning accuracy and efficiency. This paper proposes a temperature prediction model based on deep belief network, and uses Adam(Adaptive moment estimation) optimization algorithm instead of error back propagation algorithm to improve the precision of network fine adjustment. Finally, in the actual temperature prediction, the effective type of the proposed method is proved.
2. Deep belief network

2.1 Deep belief network structure

DBN (Deep Belief Nets) deep belief network is a probability generation model. Hintong et al. proposed a generation model based on deep learning technology. Its constituent elements are the restricted boltzmann machine (RBM). RBM is a kind of neural sensor, which consists of a display layer and a hidden layer. The display layer and the hidden layer are bidirectionally connected. Its structure is as shown in the following figure 1, in which the hidden layer has n nodes, and the visible layer has m nodes. \( w_i^j \) is the connection weight.

![Figure 1. Restricted Boltzmann machine.](image1)

The DBN is composed of multiple restricted boltzmann machine. The hidden layer of the previous RBM is the display layer of the next RBM. During the training process, the RBM of the previous layer needs to be fully trained to train the RBM of the current layer until the last layer. The hidden layer unit is trained to capture the correlation of higher order data represented in the visible layer. The typical structure is shown in the figure 2.

![Figure 2. DBN structure diagram](image2)

2.2 Unsupervised pre-training process

The unsupervised training of the deep belief network is to determine the initial weight of the network by training each RBM layer by layer. In this way, better training results are usually obtained more than random initialization weights. The training method of RBM generally adopts CD (Contrastive Divergence) algorithm. In RBM, the strength of the connection intension is \( w \). Vectors \( v \) and \( h \) represent the status of the visible and hidden layers, respectively. The output of all neurons has only two status, generally 1 for on and 0 for off. Then the energy function of a set of visible and hidden layer status \((v, h)\) can be defined as Equation (1).

\[
E(v, h) = -\sum_{i=1}^{m} q_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m} \sum_{j=1}^{n} v_i w_{ij} h_j
\]  

(1)
Among them, \( a \) is the offset of the visible layer node, \( b \) is the offset of the hidden layer node, and \( w \) is the connection weight matrix between the visible layer and the hidden layer. In a RBM, the probability that a hidden layer neural unit \( h \) being activated can be defined as Equation (2):

\[
P(h_j|v) = \sigma(b_j + \sum_{i=1}^{m} W_{ij}v_i)
\]

Since it is bidirectionally connected, the probability that the visible layer neurons being activated can be defined as Equation (3):

\[
P(v_i|h) = \sigma(a_i + \sum_{j=1}^{m} W_{ij}h_j)
\]

\( \sigma \) is the activation function, the criteria for determining the activation function and starting are usually achieved by setting a threshold as Equation (4):

\[
h_j = \begin{cases} 1, & p(h_j = 1|v) > \xi \\ 0, & p(h_j = 1|v) < \xi \\ \end{cases}
\]

\( \xi \) is a constant between 0.5 and 1. Since the display layer and the hidden layer are binary status, the criteria for their values can be achieved by setting a threshold. For each training sample, "data" and "model" can be used to represent the two probability distributions of \( P(h|v) \) and \( P(v|h) \) respectively. By calculating the gradient of the maximum likelihood function \( \log P(v; \theta) \), the RBM weight updating formula can be obtained as Equation (5):

\[
\Delta w_{ij}^r = E_{\text{data}}(v_i h_j) - E_{\text{model}}(v_i h_j)
\]

Among them, \( \eta \) is the learning rate, \( E_{\text{data}}(v_i h_j) \) is the expectation of the observation of the data in the training, \( E_{\text{model}}(v_i h_j) \) is the expectation of the distribution determined by the model.

2.3 Supervised fine tuning based on Adam algorithm

After the pre-training is completed, the overall parameters need to be supervised and fine-tuned to optimize the training model. The traditional method is to use the steepest gradient descent method. However, this method often has the disadvantages of slow learning speed and easy to fall into local optimum. The gradient descent method updates the model parameters by finding the minimum value and controlling the variance. Take the output layer and the last hidden layer as examples, \( y_i \) and \( y_i' \) are the desired output and the actual output of the DBN, respectively. Then define the loss function as the Equation (6). The formula for updating the parameters of the network is Equation (7).

\[
F(\tau) = \frac{1}{2} \sum_{k=1}^{K} (y_i - y_i')^2
\]

In the formula, \( \tau \) and \( K \) are the number of iterations and the number of observed samples, respectively, and Weight update can be expressed as formula (7).

\[
W_{\text{out}}(\tau + 1) = W_{\text{out}}(\tau) - \eta \frac{\partial F(\tau)}{\partial W_{\text{out}}(\tau)}
\]

\( \eta \) is the learning rate. Unlike the traditional gradient descent method, the Adam algorithm adds an adaptive time estimation variable which learning rate is adaptively changed by calculating the first order moment estimate and the second order moment estimate of the gradient, as shown in Equation (8). In practical applications, compared with other optimization algorithms, the convergence speed is faster, the learning effect is more effective, and the problems in other optimization techniques can be corrected, such as the learning rate disappears, the convergence is too slow, or the high variance.
The final formula for parameter updates is shown as Equation (9).

\[(W_{t+1})_i = (W_i) - \eta \sqrt{1 - (\beta_1)^2} \frac{(m_i)}{1 - (\beta_2)^2} \sqrt{(v_i)} + \epsilon \quad (9)\]

Among them, \(\beta_1 = 0.9\), \(\beta_2 = 0.999\), \(\epsilon = 10^{-3}\) is the default value.

3. Vacuum heat test data analysis and pretreatment

A large amount of data information is included in the entire vacuum heat test process, including vacuum degree, temperature of the measuring point, temperature of the controlled object, and container information. There are many factors affecting the controlled object during the test, so it is difficult to establish a heat transfer mechanism model. This paper takes the solar array without internal heat source as the pre-target. The variable factor which has a great influence on the controlled object is taken as the input variable, for example, the current front measuring point temperature, the back measuring point temperature, the current heat sink temperature, the current heating element input current. The output variable is the next time temperature of the measuring point.

The measurement period for the solar array during the test was 6S, and the control period of the control program was also 6S. When it is necessary to manually intervene in the control program, it is necessary to stop the running of the program first, and then restart after the modification. At this time, the time recorded by the control program is the time to restart the program. This will cause the historical data of the current to be inconsistent due to the different current change intervals during the test. Before using the data, the method of square integration of the current data is used to integrate the current data in the adjacent temperature sampling interval by time, and the integration domain is one measurement data sampling period. The equivalent current applied by the interval can be obtained by integrating the current that has changed during the time interval. After this process, the current data is matched with the temperature data.

4. Experimental verification analysis

4.1 DBN network parameter design

4.1.1 Input layer and output layer

By analyzing the factors that have the greatest influence on the temperature of the test piece in the test, the front point temperature, back surface temperature, heating element current, vacuum degree and heat sink temperature of the measuring point are finally determined as input layer. The positive measurement point temperature is the output variable at the following moment.

4.1.2 Activation function

The main features of the activation function in the neural network is to provide the nonlinear modeling ability of the network. After adding the transfer function, the deep neural network has the layered nonlinear ability. The activation function usually has properties such as non-linearity, differentiability, monotonicity, and range of output values. Common activation functions include Sigmod, tanh, Relu and so on. The comparison of the three activation functions is shown in the figure 3;
Figure 3. Activation function comparison

The Sigmod function can map a real number to the interval \((0,1)\), which is better when the feature difference is more complicated or the difference is not particularly large. Tanh function will continue to expand the feature effect during the cycle. Compared with sigmod, tanh is 0 mean, which is better in practical applications. The main advantage of Relu is to increase the convergence speed. A large number of experiments have proved that the use of binary variables in the training process is relatively simple and easy to sample, which can reduce the reconstruction error, the sigmod function can effectively convert continuous real numbers into binary variables. So the transfer function of this article is chosen as sigmod.

4.1.3 Network structure selection

The number of layers of the hidden layer and the number of each hidden layer unit in the prediction model have a great influence on the prediction accuracy and the operation time, and basically determines the overall framework of a deep learning model. But in the current research process, there is no scientific and reasonable way to know how to set the parameters. Therefore, this paper tests different parameters in a certain interval. The final model structure is finally determined. The prediction model uses a 4-layer structure, and the hidden layers are set to 20 and 15 neural units, respectively.

4.1.4 Experimental results and analysis

In order to fully verify the validity and scientificity of the method described in this paper, the feasibility of the established prediction model was tested. In this paper, the frontal temperature of the solar array thermal test data of a satellite model is selected as the training sample, and the temperature of the solar array on the other side of the satellite is used for online prediction. The training samples are shown in the figure 4: In order to highlight the advantages of the method described in this paper, BP neural network method is selected for comparison. The comparison is shown in Figure 4;
The above proved that the DBN algorithm can predict the temperature of the solar array well, and the prediction error is basically not exceeding ±1°C. At the same time, there is a relatively large error in the sudden heating or cooling phase. The reason is that the change of the operating conditions of the heat source during the test usually does not occur when the measurement and control cycle alternates, but occurs in the data measurement interval. Take the temperature rise as an example, generally speaking, the measurement and control period for the solar array is 6S. When the program records the time \( k \) data, the controller turns on the heating element before the next moment, obviously the actual temperature at the moment \( k+1 \) will be much higher than the predicted temperature, resulting in a large error after several cycles. The cooling situation is the same.

5. Summary

DBN is a hierarchical feature representation model based on the establishment of deep learning, which has strong ability to analyze, learn and predict. In this paper, we use the solar array vacuum heat test data to propose an online prediction model based on DBN, and use the Adam algorithm instead of the traditional steepest gradient descent method to improve the training accuracy and speed. And select the real test data to apply to the real-time prediction of solar panel vacuum heat test. The experimental results show that the algorithm can better predict the temperature data and has higher prediction accuracy.
accuracy. It has important practical significance for improving the control method.

References
[1] Yu Tao, Cui Junfeng. Spacecraft Environment Engineering, 22(4), 202-203 (2005)
[2] Weng Jian-hua, Pan Zeng-fu, Min Gui-rong. Chinese Space Science and Technology, 12(6), 1-6 (1997)
[3] HINTON G E, OSINDERO S, TEH Y W. Neural Computation, 18(7), 1527-1554 (2006)
[4] LIU W, WANG Z, LIU X, et al. Neurocomputing, 234, 11-26 (2016)
[5] SHEN F R, CHAO J, ZHAO J X, Neurocomputing, 167, 243-253 (2015)
[6] KUREMOTO T, KIMURA S, KOBAYASHI K, et al. Neurocomputing, 137(15), 47-56 (2014)
[7] HINTON G E, SALAKHUTDINOV R R. Science, 313(5786), 504-507 (2006)
[8] MOHAMED A, DAHL G E, HINTON G E. Speech, and Language Processing, 20(1), 14-22 (2012)
[9] LEROUX N, BENGIO Y. Neural Computation, 20(6), 1631-1649 (2008)
[10] BENGIO Y. Foundations and Trends in Machine Learning, 2(1), 1-127 (2009)