Dynamic fine-tuning stacked auto-encoder neural network for weather forecast

Lin, S-Y., Chiang, C. C., Li, J. B., Hung, Z. S. & Chao, K-M.

Original citation & hyperlink:
Lin, S-Y, Chiang, CC, Li, JB, Hung, ZS & Chao, K-M 2018, 'Dynamic fine-tuning stacked auto-encoder neural network for weather forecast' Future Generation Computer Systems, vol. 89, pp. 446-454.
https://dx.doi.org/10.1016/j.future.2018.06.052

DOI 10.1016/j.future.2018.06.052
ISSN 0167-739X

Publisher: Elsevier

Dynamic Fine-Tuning Stacked Auto-encoder Neural Network for Weather Forecast

Szu-Yin Lin[1*], Chi-Chun Chiang[2], Jung-Bin Li[3], Zih-Siang Hung[1], Kuo-Ming Chao[4]

[1]Department of Information Management, Chung Yuan Christian University, Taoyuan City, Taiwan
[2]Institute of Information Management, National Chiao Tung University, Hsin-Chu, Taiwan
[3]Department of Statistics and Information Science, Fu Jen Catholic University, New Taipei City, Taiwan
[4]School of Computing, Electronics and Mathematics, Coventry University, Coventry, UK

Abstract

With the advent of the big data era, dynamic and real-time data have increased in both volume and variety. It is difficult to make accurate predictions regarding data as they undergo rapid and dynamic changes. Autonomous cloud computing aims to reduce the time required for traditional machine learning. The stacked auto-encoder is a neural network approach in machine learning for feature extraction. It attempts to model high-level abstractions and to reduce data dimensions by using multiple processing layers. However, some common issues may occur during the implementation of deep learning or neural network models, such as over-complicated dimensions of the input data and difficulty in processing dynamic data. Therefore, combining the concept of dynamic data-driven system with a stacked auto-encoder neural network will help obtain the dynamic data correlation or relationship between the prediction results and actual data in a dynamic environment. This study applies the concept of a dynamic data-driven system to obtain the correlations between the prediction goals and number of different combination results. Association analysis, sequence analysis, and stacked auto-encoder neural network are employed to design a dynamic data-driven system based on deep learning.

Keywords: stacked auto-encoder neural network, association analysis, sequence analysis, dynamic data-driven application systems.

1. Introduction

With the advent of the big data era, dynamic and real-time data have increased in both volume and variety, rendering the processing of data challenging. During the past few years, cloud computing has emerged as a way of distributing computation loading through the Internet. However, it suffers from latency time caused by the server assigned. Autonomous cloud computing reduces the time required for machine learning [1]. If a data analysis framework can reduce computation loading without losing its accuracy, it may achieve even better performance in the autonomous cloud computing architecture.

Dynamic data-driven application system (DDDAS) has often been used for model identification and data prediction, such as concept drift detection and weather forecasting [2]. After the initial training is complete, the predictive values of a dynamic data-driven system and the real-time dynamic data can be repeatedly used to identify the strength of association between them. The prediction accuracy is also enhanced via this process. Explorations of the association between historical data and the attribute to be predicted will assist in selecting a relatively suitable input data format based on the data characteristics. A suitable framework helps train the model and reduces resource consumption. In addition, determining the dynamic data association between the system prediction value and historical data during system operation facilitates self-regulation and learning of the dynamic data driven system, and benefits the prediction accuracy. Over the past few years, deep learning has become one of the major approaches for feature extraction in machine learning. It attempts to model high-level abstractions and reduce data dimensions by using multiple processing layers. It is possible to use different data to simplify large data dimensions, and identify the potential feature of data. In recent years, deep learning neural networks have exhibited outstanding performance in feature extraction, and have often been used to solve image identification [3], [4], speech recognition [5], and time series problems [6]. However, some common issues may occur during the implementation of deep learning, such as the inefficiency of analyzing data with complicated dimensions and difficulty of execution in a dynamic environment. Therefore, combining a dynamic data-driven system with deep learning methods in a single model and making it continuously self-adjust and learn by itself will enable the subsequent dynamic analysis and prediction process to become more efficient and accurate. Deep learning methods have been applied to solve weather forecast problems in recent years, e.g., deep neural networks for time series prediction with applications in ultra-short-term wind forecasting [7]. In addition, studies have applied deep learning to extract features from weather data for weather forecast, such as

temperature prediction [6], [8]. This study considers weather data prediction (e.g., rainfall) as an example.

This study applies the concept of a dynamic data-driven system to obtain the correlation between the prediction targets and number of different combination results. Association analysis, sequence analysis, and stacked auto-encoder neural network are also applied to design a dynamic data-driven fine-tuning approach. Our improved prediction model is accurate and demonstrates good efficiency in a dynamic environment.

2. Related Works

2.1 Association analysis

Association rule [9] analysis is an important method of data mining, and it is used to determine valuable data association relationships from a large volume of data. Association rules are commonly applied in solving problems such as shopping basket analysis: if a customer buys product A, what is his/her probability of buying product B? Similar to many other data mining methods, the association rule analysis extracts information that is not pre-defined, and makes such information sufficiently concise to be understood. In this method, the order of items in an item set is not important, but the combination of items is. In the case of sequence analysis, the order of items and the time interval are considered. In sequence analysis, two thresholds are often used: minimum support and minimum confidence. The minimum support defines the minimum number of data that a rule must cover, and the minimum confidence defines the prediction strength of the association rules. These two thresholds control the availability and certainty of the association rules, respectively. With these two thresholds, a more sophisticated data format for different machine learning methods can be compiled as input. As the association rule method evolved [10], Apriori algorithm was proposed in 1994 by Agrawal and Srikant of IBM Almaden Research Center, showing that all frequent item sets in the database satisfy minimum support. An item group containing k items is called k-item set, and the algorithm repeatedly searches from 1 to k until all eligible items are found. These frequent sets are called candidate sets. Subsequently, these candidate sets are screened to verify if they satisfy the threshold to form an association rule. For example, XY is a 2-item set, whereas X and Y are 1-item sets. If X -> Y satisfies the minimum confidence, an association rule is established. To increase the association rules, the minimum support value can be reduced; however, the resulting rules may lose their significance. Considering the relevance, it is necessary to form association rules with discretion. Birn et al.

proposed the lift correlation coefficient in 1997 to evaluate the effectiveness of the rules.

2.2 Deep learning

Deep learning is an emerging subfield of machine learning. In machine learning, deep learning is a multi-hierarchical approach to process data in data retrieval, image, and speech recognition domains. It extracts meaningful image features in different layers, and modifies the image content into abstract semantic concepts [11]. Its aim is to abstract the data through multiple non-linear transformations. Deep learning is a type of artificial neural network learning method that simplifies data dimensions and discovers potential features. However, it is challenging to train a neural network with deep architecture effectively while preserving the relevance between the parameters across the layers. In previous studies, it was indicated that setting more layers to a shallow neural network does not improve the performance of model training. In 2006, Hinton's revolutionary work spearheaded the advance of deep belief networks. Moreover, Hinton [12], Bengio [13], and Ranzato [14] have published papers to popularize deep learning by solving the problem of training deep neural networks. The training process is generally divided into two stages: (1) unsupervised learning of representations is used to train each layer, and the representation learned at each level is the input of the next layer; (2) using supervised training to fine-tune all the layers to minimize prediction error. This greedy layer-wise unsupervised pre-training method has been applied in many fields. It is commonly used for unsupervised stacking models such as restricted Boltzmann machines (RBM), convolutional neural network (CNN), and auto-encoder. The greedy unsupervised layer-wise training strategy helps to optimize deep networks and obtain better generalization for networks, because this strategy initializes upper layers with better representations of relevant high-level abstractions. In this study, the greedy layer-wise strategy is applied with a stacked auto-encoder. Moreover, in the field of deep learning, although CNN has successfully been applied to the analysis of visual imagery, its task is to find a set of locally connected neurons with convolutional kernels.

Instead of using a CNN, auto-encoder, a type of artificial neural network, has been used for learning efficient encodings and minimizing error or reconstruction. It is suitable for real numbers to determine the most efficient compact representation in linear problems. Its typical application is to reduce data dimensions [15]. It extracts data features via an encoding process, and restores data by decoding these features. The setting of the hidden layer is crucial to the learning performance of the auto-encoder [16]. Auto-encoder exhibits good performances in solving both numerical and categorical problems, and it can also be combined with other neural

network theories [17]. There are other extensions of auto-encoders, such as denoise auto-encoder [4] [18] and sparse auto-encoder [19] [20] [21]. As the auto-encoder outperforms RBM-based deep neural networks in literature [4] [17], it has been adopted as the base network structure of our previous study in [22].

2.3 Fine-tuning the neural network

The objective of fine-tuning is to adjust the weights of the trained model from the final phase to improve the prediction outcome. This procedure, based on the concept of transfer learning [23] [24], includes the process of pre-training neural networks with a generative objective followed by additional training procedures with a discriminative objective on the same dataset [25] [26], but some other studies follow the process of re-using weight values from large datasets as initialization in applications with limited access to labeled data [27] [28]. Many fine-tuning works have been conducted to improve the performance of CNN [29] [30] and auto-encoders [4] [31].

3. Dynamic Fine-Tuning Stacked Auto-encoder Neural Network

In this section, a dynamic fine-tuning stacked auto-encoder neural network is demonstrated. First, the problems related to dynamic data environment are addressed. Second, the model of feature extraction and stacked auto-encoder neural-network in DDDAS is introduced. Third, the proposed system is introduced in detail. Finally, the investigation and evaluation of the proposed system are presented.

3.1 System background

Fast-growing and rapidly changing data are difficult to analyze using traditional prediction models, especially in dynamic, real-time, or multi-dimensional time series conditions. This study proposes a dynamic fine-tuning stacked auto-encoder neural network framework. The fine-tuning deep learning architecture used in the system is based on the past studies of auto-encoder [32] and stacked auto-encoder using a greedy layer-wise pre-training method [13]. This system explores the correlation between historical data in advance. In other words, it extracts the most significant influential factors from historical data over a period, and uses these factors to represent the features of input data in order to help the training and dynamic fine-tuning process of the prediction model. The dynamic fine-tuning process follows the most common method used in the neural network field i.e., backpropagation method. Error gradients can be propagated from the output layer to input layer sequentially. Using the above methods, the correlation between data is enhanced, and

the prediction accuracy of the deep learning model is improved. In the experiment, various possible combinations of data dimensions and cases will be employed in order to determine the most suitable deep learning method in a dynamic data environment.

3.2 Proposed architecture and system

As shown in Fig. 1, the proposed operation of DDDAS can be divided into two phases. The first phase is the training of the deep learning model. The model training process comprises two steps: feature extraction and deep learning based on neural network. The former extracts feature via association analysis, and the latter trains the deep learning prediction model using the extracted features. The second phase is dynamic fine-tuning and data prediction. The prediction model continues to fine-tune itself dynamically with the newest data, and data prediction is also included in this phase.

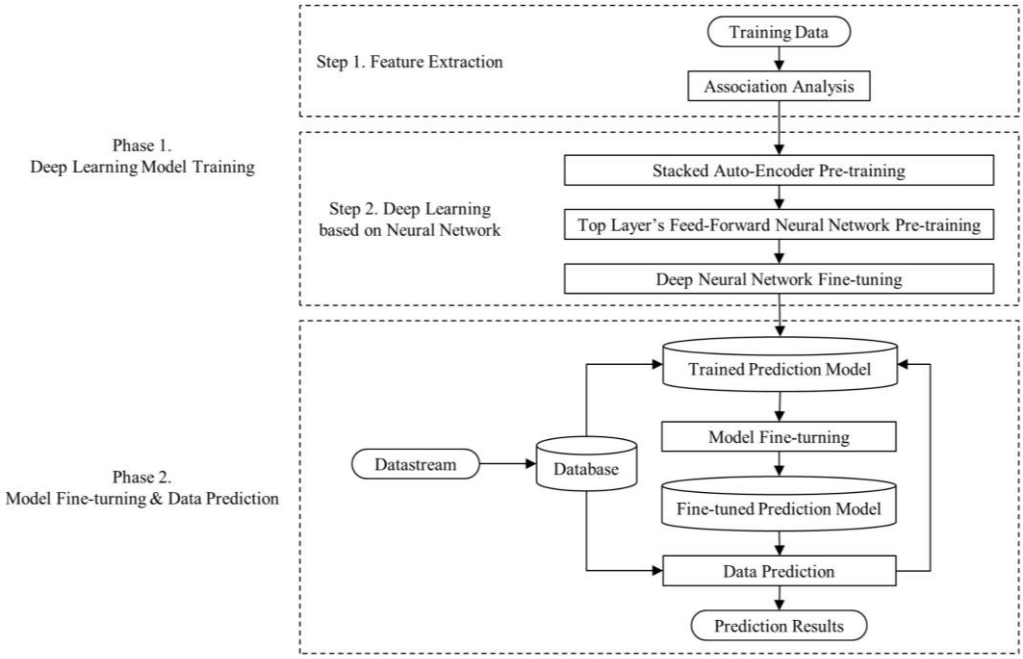

Fig. 1 Mechanism of the proposed model

3.3 System details

As mentioned above, there are two phases in the proposed system, and all the notations are listed in Table 1.

Table 1. Symbol definitions

Symbol	Definition
$b_{i,t}$	The value of the i_{th} attribute at time t
v_i	The change amount per unit time of the i_{th} attribute
$\overline{v_i}$	The average change amount per unit time of the i_{th} attribute
n	The total number of data in the training dataset
t	Time
c_i	The binary up-down value of the i_{th} attribute after associativity transform
h_i	The number of nodes in the i_{th} layer of the deep learning network, where i = 1 denotes the bottom layer.

3.3.1 Feature extraction using association analysis

Input data can be transformed into a structure similar to transaction records after they undergo the following data association transformation, and thereafter, this format can be used for all historical data. After the minimum support and minimum confidence thresholds are defined, many association rules can be determined using Apriori algorithm, and these rules are considered the most influential factor for prediction. In the process, data correlation of historical data will be identified via association analysis and sequence analysis. Therefore, it is necessary to transform the data into an executable format before calculating the association rules. In this case, numeric data will be transformed into a format of increasing or decreasing status via the data changing situation of different attributes between every two consecutive time points. Using the association rules and sequence analysis, the rising or falling correlation of different attributes for different time intervals can be determined. The process of data transformation is as follows. First, the value of $\overline{v_i}$, which represents the average change amount per unit time of each attribute i, can be calculated using Equation (1). The value of $\overline{v_i}$ is thereafter compared with the value of $\overline{v_i}$ in Equation (2), in order to obtain four associativity conditions—substantially increasing, slightly increasing, substantially decreasing, and slightly decreasing.

$$\bar{v}_i = \frac{1}{n} \sum_{t=2}^{n} |b_{i,t} - b_{i,t-1}| \quad (1)$$

$if\ v_i > 0\ and\ |v_i| \geq \bar{v}_i\ then\ c_i\ is\ substantially\ increasing.$

$if\ v_i > 0\ and\ |v_i| < \bar{v}_i\ then\ c_i\ is\ slighty\ increasing.$

$if\ v_i < 0\ and\ |v_i| \geq \bar{v}_i\ then\ c_i\ is\ substantially\ decreasing.$

$if\ v_i < 0\ and\ |v_i| < \bar{v}_i\ then\ c_i\ is\ slighty\ decreasing.$

where $v_i = b_{i,t} - b_{i,t-1}$ (2)

t	$b_{1,t}$
1	24.5
2	23.6

$\bar{v}_i = 1.2$

t	v_i
1 to 2	-0.9

t	c_i Substantially Increasing	c_i Slightly Increasing	c_i Substantially Decreasing	c_i Slightly Decreasing
1 to 2	0	0	0	1

Fig. 2. Process of associativity transform

Fig. 2 shows the process of the associativity transform. The values of input data change on a temporal basis, and such a relation can be transformed to a structure similar to a transaction record in shopping basket analysis. For example, if "the tendency of b_i at $t = 1$ to 2 " in Fig. 2 is regarded as a record, the item 'c_i substantially increasing with the value 0 can be treated as the purchase of no commodities in the transaction. If the value of slightly decreasing item 'c_i is 1, it denotes a purchase in this transaction. By applying the Apriori algorithm of association rules, we explore the relations among the tendencies of data attributes. Sequence analysis is applied when the factor of time is considered to determine the relations among the tendencies of data attributes at different times. For example, if event C usually occurs after two appearances of event A, this pattern appears two times in a certain period, marked with yellow and green highlights, as illustrated in Fig. 3.

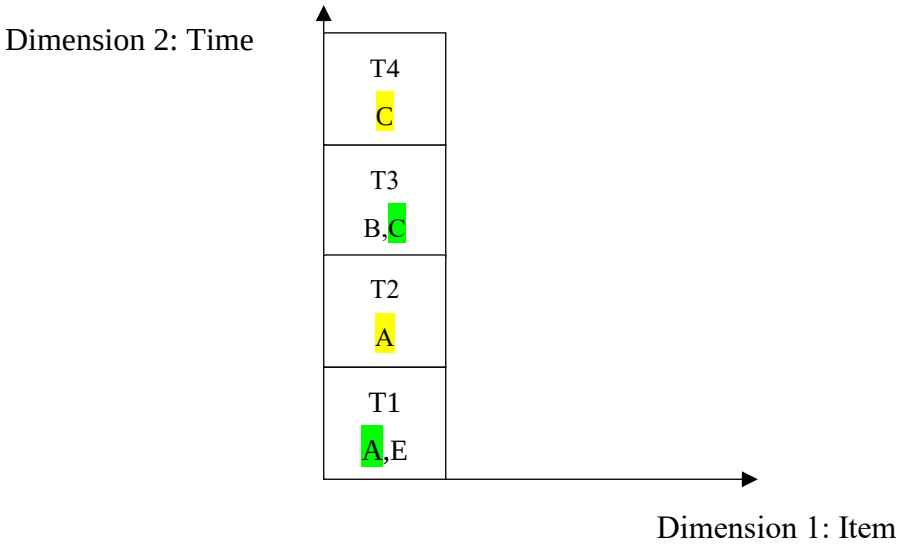

Fig. 3. Multi-dimensional data association rules and sequence analysis

The Apriori algorithm has two stages. In the first stage, it discovers all item sets that exceed the minimum support value; the second stage summarizes the association rules from large items sets. As the first part consumes the majority of computation loading, we focus on the exploration of large item sets with efficiency. Previous improvements include hashing, sampling, database table partitioning, frequent pattern (FP)-growth, etc. FP-growth is the most efficient algorithm to compress all complicated items into an FP-tree while preserving relational information among items [33]. Another advantage of this algorithm is that it scans a database only twice at most without generating numerous candidate item sets.

In the case of weather prediction, all historical data are transformed and analyzed using FP-growth. Based on the data of previous years, the prediction model explores the relations among the tendencies of data attributes. The threshold values of minimum support and minimum confidence are chosen based on the attribute to be predicted, and thereafter, different combinations of associated item sets are obtained. The input data influencing the observed question can be determined via the following experiments.

3.3.2 Deep learning based on neural network

(1) Stacked auto-encoder pre-training

The association rules obtained in the previous phase and the original observation data are used as the input for deep neural network pre-training. A stacked auto-encoder neural network is constructed by training the auto-encoders using a supervised training technique and stacking them based on the concept of sparse auto-encoder.

The number of nodes in each hidden layer decreases with the increase in the number of layers. For example, if the structure consists of three hidden layers, the number of nodes would conform to the order $h_3 > h_2 > h_1$. The structure ensures the dimension reduction of the original data by the abstracted layers. In this study, there are two time slots—one is set for every 6 h and another is set for every 12 h—where the hourly-observed data used for input are considered for the period of the last three and a half years. It would be ideal if the proposed layer-wise training can reduce the data dimensions by more than half that of the original before being inputted into the supervised regression model.

(2) Top layer's feed-forward neural network pre-training

The stacked model constructed in the previous phase considers the original data as input and generates new data with comparatively lower dimensions. The new data will thereafter be used as input for the top layer's feed-forward neural network pre-training. The data of the next time period will be set as the target value, which would be the actual data value for the designated time period during the prediction. The process will generate a trained regression data model, which will thereafter be stacked with the previous stacked neural network model to form the deep learning neural network structure. This stage provides efficient initialization for the weights in the deep learning network, which can be remodeled for data prediction after performing the fine tuning technique, which will be introduced in the next phase.

(3) Deep neural network fine-tuning

The objective of fine-tuning is to adjust the weights of the trained model from the previous phase to improve the prediction outcome by using all the training dataset. This procedure utilizes a loss function and gradient descent algorithm from the stacked sparse auto-encoder.

3.3.3 Dynamic fine-tuning & data prediction

This phase ensures that the data are collected and stored dynamically during each time slot. Fine-tuning and prediction are executed during every time slot. The model first performs fine-tuning based on all the input data, to generate a trained model that is dynamically adjusted. When the newest data are collected, the model produces the prediction results in the next time slot.

3.4 Summary

In comparison with the traditional deep learning prediction methods, the use of association rules assures the trends of data prediction. However, the number of feature

selection creates drawbacks in the process of sacrificing some insignificant feature attributes. The use of the top layer's supervised training provides the deep neural network a consistent basis for adjustment, which also benefits the later dynamic fine-tuning process by utilizing the completely deep network, unlike the traditional classifiers such as support vector machine, which can only be adjusted based on the result of the stacked auto-encoder. Thus, the method can reduce the training time and achieve precise prediction.

4. Simulation and Results Analyses

In this section, we present the simulation scenarios in order to verify the proposed system. The dataset, simulation cases, and evaluation metrics are also illustrated.

4.1 Data sets and data preprocessing

The CWB (Central Weather Bureau) in Taiwan has provided a massive volume of observation data on CWB Observation Data Inquiry System. Based on this system, real weather data could be applied in this experiment. The time range of the data set is covering the period from June 1, 2012 to January 31, 2016. The number of records is approximately 32,000. Fifteen attributes are included in the historical weather data set, e.g., pressure, temperature, precipitation, humidity, etc. Temperature is the prediction target in the experiment. Some attributes such as sunshine usually have missing values. Some attributes such as wind direction do not have numerical significance. Therefore, according to the requirement of our algorithm, we performed pre-processing to filter these attributes out from the data set. After the pre-processing, only eleven attributes remained in the experiment data set.

4.2 Simulation design

In this study, in order to simulate and evaluate the data prediction performance in the dynamic data environment, the simulation experiments were designed and developed in MATLAB and its toolbox. We developed the algorithm of association rules in the earlier stage of dynamic fine-tuning process, and thereafter designed and modified the neural network algorithm of stacked auto-encoders. In the experiments, we attempted a few combinations of hyper-parameters, such as layer depth, number of neurons, and learning rates. After comparing the performances and execution times of different combinations, we chose the most stable and efficient one as the deep neural network structure. We used a four-layer deep neural network model, consisting of an input layer, auto-encoder layer, and feed-forward neural network top layer, to predict the temperature at the next time point. Fig. 4 shows the structure of the deep neural

network. In the model training process, we used the last 12 hourly weather data tuples to forecast the temperature in the next hour. Each tuple in the raw data set has 11 attributes every hour, and hence, the total number of attributes is 132. After the association analysis, we used the extracted features to make predictions.

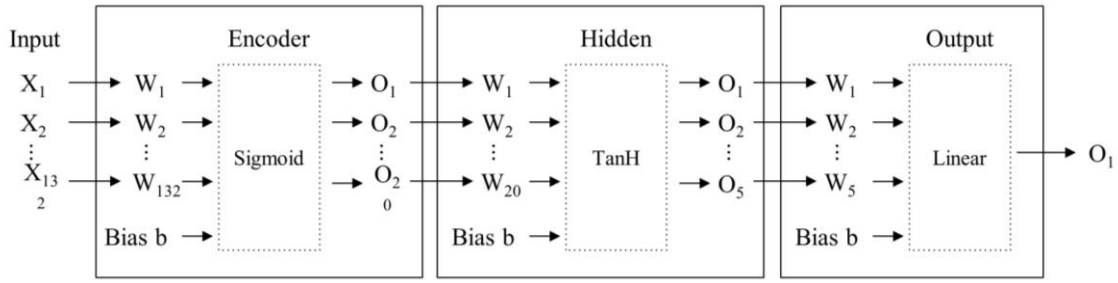

Fig. 4. Structure of the deep neural network

4.3 Evaluation metrics

In order to evaluate whether the proposed system predicts the temperature accurately and efficiently, we define three metrics: (1) initial training time & average dynamic fine-tuning time, given by the amount of time taken when using different number of attributes; (2) normalized mean squared error (NMSE), which measures the deviation between the actual values and predicted values, as shown in Equation (3); (3) directional symmetry (DS), which is the correctness percentage of the predicted direction, as shown in Equation (4).

$$\text{NMSE} = \frac{1}{n} \times \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\bar{x}\,\bar{\hat{x}}} \tag{3}$$

$$DS = \frac{1}{n} \times \sum_{i=1}^{n} d_i \tag{4}$$

$$\text{Where} \quad d_i = \begin{cases} 1, (x_i - x_{i-1})(\hat{x}_i - \hat{x}_{i-1}) \geq 0 \\ 0, otherwise \end{cases}$$

4.4 Case definition

The experiment consists of 12 different test cases with the combination of three control variables: the number of association rules, size of the testing dataset, and the usage (versus non-usage) of the dynamic fine-tuning process. The result will be used to verify the proposed method. Table 2 presents the details for each test case.

4.4.1 Number of association rules

The association rules are applied to determine the relation between the weather conditions for the past 12 days and the prediction result. The thresholds are set at lower values in order to obtain more comprehensive association rules (Thresholds: minimum support = 0.05, minimum confidence = 0.2). The objective here is to first rank the association rule by minimum support, and thereafter compare the prediction result by the different percentages of the ranked association rules used, such as the top 25%, top 50%, and 100%. The analysis provides information on the influence of the volume of association rules on the prediction result.

4.4.2 Size of testing datasets

The test cases also differ by the size of datasets—some used the datasets from one week, whereas others used the dataset from one month. The purpose of employing this variation is to observe the difference between the results of short-term and long-term predictions.

4.4.3 Usage of dynamic fine-tuning

Dynamic fine-tuning is used as a control variable to understand whether it can improve the efficiency of the prediction model. In consideration of computational complexity of our model, it varies from the framework settings and parameters of neural networks which cause differences and difficulties in performance evaluation. Thus we analyze model performances based on the same model setting and framework to observe the performance difference between whether dynamic fine-tuning process is adopted or not, namely the computational complexity of model is fixed in the following comparison. The result is analyzed in terms of accuracy and computing time.

Table 2. Case definitions

No.	Number of Association Rules (%)	Size of Testing datasets	Usage of Dynamic Fine-Tuning
1	25%	1 week	No
2	50%	1 week	No
3	100%	1 week	No
4	25%	1 month	No
5	50%	1 month	No
6	100%	1 month	No
7	25%	1 week	Yes
8	50%	1 week	Yes

9	100%	1 week	Yes
10	25%	1 month	Yes
11	50%	1 month	Yes
12	100%	1 month	Yes

4.5 Simulation results

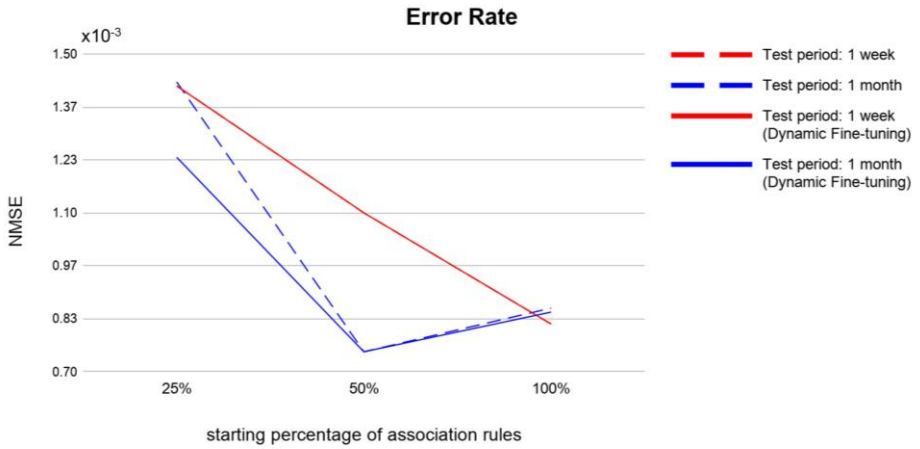

Fig. 5. NMSE comparison chart

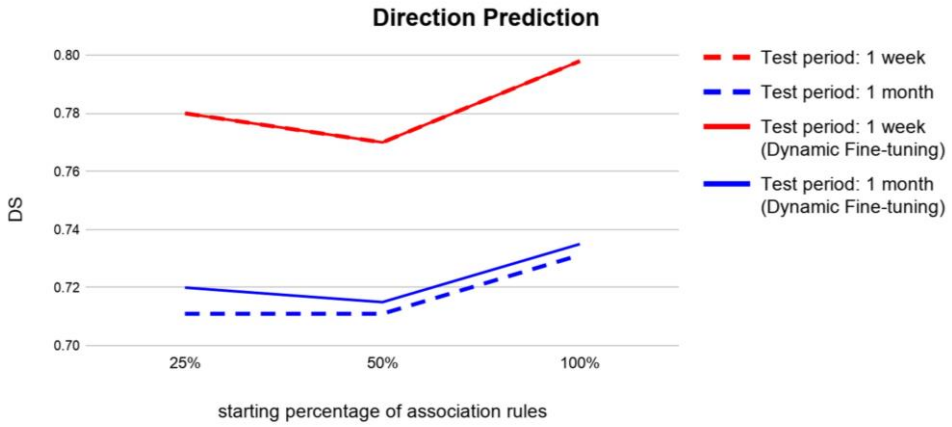

Fig. 6. DS comparison chart

Fig. 5 presents the NMSE comparison chart in this experiment. In the case of one-week prediction, the error rate decreases when the feature number increases. Two red lines overlapped with each other, indicating that the adoption of dynamic fine-tuning did not affect their NMSE performances. However, in the case of one-month prediction, the error rate increases when the full feature set is used, and the

best performance is achieved when the top 50% associated features are used. This is because when considering excessive number of features in the training process, minor noises are included, thus causing overfitting. This phenomenon does not appear in the one-week predictions as 168 testing tuples are not sufficient for it to occur. Dynamic fine-tuning seems inessential in the case of one-week prediction as the model does not have adequate time for the learning behavior to take effect and would only result in a waste of computing resources. In the one-month prediction, the dynamic fine-tuning effectively reduces the error rate and improves the overall accuracy. The best NMSE for all the cases reaches the rate 7.5e-4, and the dynamic fine-tuning reduces the error rate by 3%.

Fig. 6 presents the DS comparison chart for all the cases. It appears that the one-week prediction outperforms the one-month prediction. This is because continuous training renders the model more applicable to different kinds of data; therefore, the overall performance for the one-month prediction is more stable. In the one-week scenario, two lines are overlapped, and the learning direction is better and achieves the best accuracy rate of 79.8%, owing to its inexperience of too much data. Moreover, the use of the top 50% associated features does not result in a better performance, because the transformation is subjected to the change amount of values rather than the ranking of the values in the process of association rule transformation. Thus, it can be concluded that different association rule transformations have different effects on the prediction accuracy and direction. This is because the time is not sufficient for the model's learning and adjusting behaviors to take effect, rendering the use of fine-tuning inefficient. In the one-month prediction, the use of fine-tuning effectively enhances the directional prediction. The dynamically fine-tuned model increases the DS accuracy by 0.5% in comparison with the model without such process. Table 3 shows the performances of all the cases in terms of accuracy and execution time.

Table 3. Accuracy and execution time of cases

No.	Initial Training time (sec)	Average Dynamic Fine-tuning time (sec)	NMSE	DS
1	112.5471	0	0.00141162	0.779762
2	198.5657	0	0.00107861	0.767857
3	309.4416	0	0.00081201	0.797619
4	192.7309	0	0.00142664	0.711111
5	214.5558	0	0.0007538	0.708333

6	380.6261	0	0.00086337	0.731944
7	112.5471	5.84644997	0.00141162	0.779762
8	198.5657	21.6370609	0.00107861	0.767857
9	309.4416	83.1199062	0.00081201	0.797619
10	192.7309	5.94193332	0.00125021	0.719444
11	214.5558	23.6020907	0.00075219	0.7112
12	380.6261	85.388829	0.00085914	0.7341
Avg.	234.7445	18.7946892	0.00104249	0.750551

5. Conclusions and Future Works

In the DDDAS, the association rule analysis is used to explore the data characteristics and it transforms the input data values. Furthermore, the DDDAS concept is introduced to the proposed deep learning method, which helps the identification of the relation between the system's prediction values and true observation values. This combination assists the continuous self-learning and self-adjusting of DDDAS and leads to a more accurate and efficient dynamic prediction process. The proposed method aims to solve the problem of weather forecasting. The experiment results show that the method reduces the average error rate by 87% and improves the maximum DS by 0.8%. Furthermore, the method requires only 0.5% of fine-tuning time. In conclusion, this study proposes a DDDAS based on deep learning. The experiment also proves that the system is suitable for dynamic data environment and exhibits more efficient and accurate performances.

In this study, association rules are applied to enhance the speed of computation, while maintaining the accuracy simultaneously. Owing to the differences between the data association results, the form of data after transformation also varies. The quality and quantity of attributes obtained are influenced by the values of support and confidence. The form of transformation required for depicting data association is determined by the characteristics of the data themselves. Therefore, the association of data should be observed after the exploration of the relation between the data and prediction target. For example, the definition of association in the experiment depends on the possible variation and average variance of the issue to be predicted. Hence, in the process of transformation of Phase 1–Step 1 in Fig. 1, original data are converted to the associated form by their tendency and value.

This study considers the issue of weather prediction for example, and adopts an auto-encoder as the stack layer of a deep learning network. Including the bottom input

layer and the supervised learning layer on the top, there are four layers in the proposed network. However, this network setting is not likely to fit all the prediction problems. When the amount of data attributes is not sufficient, the number of network layers should be reduced accordingly. Attempts to use a network with excessive layers may cause unnecessary resource consumption. Moreover, some substitution or hybrid in the stack layer may enable neural network training in the field of multimedia, such as the CNN in graphics or audio recognition or fully connected neural network in certain applications.

In the future, the transformation process would be further improved to better fit different data sets or for predicting issues individually, such as PM2.5 pollutions and price fluctuations, whose prediction target values are numeric data and are affected by other increasing and decreasing factors. Moreover, the threshold values of association rules may be changed to determine better accommodation of operation scenarios. Instead of using the greedy layer-wise pre-training strategy, the classification layer can be joined to the training process to achieve model accuracy [33]. In order to implement this model in the real world, online-learning could be considered in the future [34], [35]. For the dynamic tuning of parameters, fuzzy inference systems [36], [37] and extreme learning machines [38] may also be used to further improve the proposed system.

Acknowledgment

This research work was supported by the Ministry of Science and Technology, Taiwan under the grant 106-2410-H-033-012. This work is partially supported by EU H2020 programme (Project NOESIS under grant no 769980).

References

[1] Malensek, M., Pallickara, S., & Pallickara, S. Autonomous Cloud Federation for High-Throughput Queries over Voluminous Datasets. IEEE Cloud Computing, 3(3) (2016) 40–49.

[2] Lin, S. Y. Reinforcement learning-based prediction approach for distributed Dynamic Data-Driven Application Systems. Information Technology and Management, 16(4) (2015) 313–326.

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems. (2012) 1097-1105.

[4] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. Stacked denoising autoencoders: Learning useful representations in a deep network with

a local denoising criterion. Journal of Machine Learning Research, 11 (2010) 3371-3408.

[5] Deng, L., &Platt, J. C. Ensemble deep learning for speech recognition. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. (2014) 1915–1919.

[6] Liu, J. N., Hu, Y., He, Y., Chan, P. W., & Lai, L. Deep neural network modeling for big data weather forecasting. In Information Granularity, Big Data, and Computational Intelligence. (2015) 389-408.

[7] Dalto, M., Matusko, J., & Vasak, M. Deep neural networks for ultra-short-term wind forecasting. In Proceedings of the IEEE International Conference on Industrial Technology. (2015) 1657–1663.

[8] Grover, A., Kapoor, A., & Horvitz, E. A deep hybrid model for weather forecasting. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2015) 379-386.

[9] Zhang, C., & Zhang, S. Association rule mining: models and algorithms. Springer-Verlag, 2002.

[10] Cheung, D. W., Han, J. H. J., Ng, V. T., Fu, A. W., &Fu, Y. F. Y. A fast distributed algorithm for mining association rules. Fourth International Conference on Parallel and Distributed Information Systems. (1996) 31–42.

[11] Kuo, C. H., Chou, Y. H., & Chang, P. C. Using deep convolutional neural networks for image retrieval. Electronic Imaging. 2 (2016) 1-6.

[12] Hinton, G. E., Osindero, S., & Teh, Y. W. A fast learning algorithm for deep belief nets. Neural Computation. 18(7) (2006) 1527–1554.

[13] Bengio, Y., Lamblin, P., Popovici, D., &Larochelle, H. Greedy Layer-Wise Training of Deep Networks. Advances in Neural Information Processing Systems. 19(1) (2007) 153-160.

[14] Ranzato, M. A., Poultney, C., Chopra, S., & Lecun, Y. Efficient Learning of Sparse Representations with an Energy-Based Model. Advances In Neural Information Processing Systems. 19 (2007) 1137–1134.

[15] Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. ICML Unsupervised and Transfer Learning, (2012) 37–50.

[16] Chandra, B., & Sharma, R. K. Fast learning in Deep Neural Networks. Neurocomputing, 171 (2016) 1205–1215.

[17] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning - ICML ’08 (2008) 1096–1103.

[18] Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining - WSDM ’16 (2016) 153–162.

[19] Zhang, Y. D., Zhang, Y., Hou, X. X., Chen, H., &Wang, S. H. Seven-layer deep neural network based on sparse autoencoder for voxelwise detection of cerebral microbleed. Multimedia Tools and Applications. (2017) 1–18.

[20] Jia, W., Muhammad, K., Wang, S. H., & Zhang, Y. D. Five-category classification of pathological brain images based on deep stacked sparse autoencoder. Multimedia Tools and Applications. (2017) 1–20.

[21] Jia, W., Yang, M., & Wang, S. H. Three-Category Classification of Magnetic Resonance Hearing Loss Images Based on Deep Autoencoder. Journal of medical systems. 41(10) (2017) 165.

[22] Lin, S. Y., Chiang, C. C., Hung, Z. S., & Zou, Y. H. A Dynamic Data-Driven Fine-Tuning Approach for Stacked Auto-Encoder Neural Network. In 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE) (2017) 226–231.

[23] Bengio, Y. Deep Learning of Representations for Unsupervised and Transfer Learning. In JMLR: Workshop and Conference Proceedings Vol. 7. (2011) 1–20.

[24] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning. (2014) 647-655.

[25] Käding, C., Rodner, E., Freytag, A., & Denzler, J. Fine-tuning deep neural networks in continuous learning scenarios. In Asian Conference on Computer Vision (2016) 588-605.

[26] Erhan, D., Bengio, Y., Courville, A., Manzagol, P. A., Vincent, P., & Bengio, S. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research. 11 (2010) 625-660.

[27] Agrawal, P., Girshick, R., & Malik, J. Analyzing the performance of multilayer neural networks for object recognition. In European conference on computer vision. (2014) 329-344.

[28] Girshick, R., Donahue, J., Darrell, T., &Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2014) 580–587.

[29] Reyes, A. K., Caicedo, J. C., & Camargo, J. E. Fine-tuning deep convolutional networks for plant recognition. In CEUR Workshop Proceedings (Vol. 1391) (2015).

[30] Zhou, Z., Shin, J., Zhang, L., Gurudu, S., Gotway, M., & Liang, J. Fine-Tuning Convolutional Neural Networks for Biomedical Image Analysis: Actively and Incrementally. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 7340-7349.

[31] Kim, M., & Smaragdis, P. Adaptive denoising autoencoders: A fine-tuning scheme to learn from test mixtures. In International Conference on Latent Variable Analysis and Signal Separation. (2015) 100-107.

[32] Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. Dynamic itemset counting and implication rules for market basket data. ACM SIGMOD Record, 26(2) (1997) 255–264.

[33] Sahoo, D., Pham, Q., Lu, J., & Hoi, S. C. Online Deep Learning: Learning Deep Neural Networks on the Fly. arXiv preprint arXiv:1711.03705. (2017).

[34] Hettinger, C., Christensen, T., Ehlert, B., Humpherys, J., Jarvis, T., & Wade, S. Forward thinking: Building and training neural networks one layer at a time. arXiv preprint arXiv:1706.02480. (2017).

[35] Mutanu, L., & Kotonya, G. A self-learning approach for validation of runtime adaptation in service-oriented systems. Service Oriented Computing and Applications. (2017) 1–14.

[36] Jang, J. S. R. ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Systems, Man and Cybernetics, 23(3) (1993) 665–685.

[37] Dorn, C., & Dustdar, S. Weighted fuzzy clustering for capability-driven service aggregation. Service Oriented Computing and Applications, 6(2) (2012) 83–98.

[38] Huang, G. B., Zhu, Q. Y., & Siew, C. K. Extreme learning machine: theory and applications. Neurocomputing, 70(1-3) (2006) 489-501.