A Prediction Approach of Hot Electric Power Researches for Heterogeneous Time Spans

Yan Ma1,*, Wei Qi2, Dali Qi1, Yufeng Chen1 and Lida Zou3,*

1State Grid Shandong Electric Power Research Institute, Jinan 250002, China.
2Shandong College of Electronic technology, Jinan 250200, China
3Shandong University of Finance and Economics, Jinan 250014, China.

*Corresponding author e-mail: zoulida@163.com, *yanpony@126.com

Abstract. Recently with the rapid growth of electric power literatures, it is hard to artificially track and process hot electric scientific researches. In the past most, professionals use simple statistics to get high-frequency words, which is time-consuming and ignores the similarity between words. Moreover, different researchers have different requirements for prediction time span. In the paper we propose a prediction system for hot electric scientific researches and gives its implementation. The proposed embedded RNN prediction model is flexible for heterogeneous time spans and can return prediction results rapidly and accurately. Our extensive experiments demonstrated that our approach has acceptable precision ratio as well as training time in comparison to SVM, RNN and linear regression algorithms. It also performs better when the embedded layers are multiple.

1. Introduction

Sci-tech information research plays an important role on the establishment and implementation of strategies and plans for states, society and enterprises [1]. The prediction of hot electric power researches is one of the newest applications in the fields of information science [2]. Scientific workers and project managers should have certain foresightedness during choosing topics and approving projects. They make decisions on potentially new theories or valuably new technologies.

At present, prediction methods of hot electric power researches severely rely on senior professionals in the fields. Moreover, they mostly use the time-consuming approaches, such as literature review and on-the-spot investigation, etc. As the development of cloud computing and artificial intelligence, predicting hotspots based on machine learning is feasible. The intuitive way is downloading the literatures in the database using crawler technology, converting text data into time series data through feature representation and feature extraction, and last obtaining the future hotspots based on time series prediction algorithms, such as RNN (recurrent neural network), LSTM [3], GRU [4] etc.

However, different users have various definitions for future time interval, i.e., time span of prediction is different. Some users need to predict the hotspots after a week and some other users need the results after a month or even a year. Aiming for the scenario, a prediction and push system for hot electric power scientific researches is necessary, which could make quick and accurate judgement in situations of different time spans, and push them to electric power researchers.
2. The Prediction Framework
To solving the above problem, we propose a prediction method of scientific hotspots in different time spans, which could quickly predict hot scientific researches for any future time span. Scientific hotspot prediction means to predict the possibly appeared hot electric power researches for some time to come. Different time span means to set different lengths of time according to user requirements.

We divide the system into five modules, as shown in Fig. 1. These modules are separately demand analysis, model generation, model training, feature extraction and result prediction. In the figure, the dashed lines show control flow and full lines represent data flow.

![Figure 1. The prediction framework](image)

After giving the prediction framework, we introduce the whole flow of generating the hot electric scientific researches in different time spans, as shown in Fig. 2. First the system prepares for data input. It crawls enough data from literature database and time series data are obtained after feature representation and feature extraction. Then for different user demands, the system analyzes the prediction period and generates the structure of model. Based on part of time series data, the model is trained and modified for many times in order to get the accurate prediction results. Last, we get a relatively stable and fine model. Based on the newly input data, user demand and the prediction model, it is quick to get and return a series of hot scientific researches.

3. The Concrete Prediction Approach
In the section, we concretely introduce each module of the framework. Since we periodically collect electric power information, the input data are time series data. RNN is a time series forecasting model and we have done some preliminary work [5]. In the paper we further improve our previous research to adapt to the scenario of indefinite time span.

3.1. Demand Analysis Module
The module is mainly responsible for providing user-friendly interface and interacting with users. It analyzes user required time span for hotspot prediction and transmits time span data to model generation module.

3.2. Model Generation Module
To adapt to prediction task for indefinite time span, we propose an embedded RNN prediction model, which could effectively accelerate training speed of prediction model. Our method is improved based on RNN model, as shown in Fig. 3. Its bottom layer is an RNN model and a sub RNN model is embedded in each cycle. The cycle of sub model can also embed a grandson RNN model. For example, each cycle of G0 embeds G1, and each cycle of G1 embeds G2. If the times of iterative embedding is $\sigma - 1$, the model is called $\sigma$-layer embedded RNN model. The right part in Fig. 3 is three-layer BP neural network.
Figure 3. The structure of embedded RNN prediction model

Figure 4. Forward propagation and error back propagation of model training
3.3. Model Training Module

The embedded RNN prediction model is trained layer by layer. It first trains the top RNN and then successively trains the lower layer. Fig. 4 shows its main steps.

1) First determine training data of each layer. Assume that our system continuously recorded data at previous arbitrary time. a) We use \( h \) to denote period coefficient of training data, which means how long to train a layer of RNN model. b) Training data of each layer is \( \{ x_i | i \in [t - hK, t], i \in K = 0 \} \), where \( k \) is the layer number starting from 0. For example, if \( h = 2, k = 2 \) stand and the current time is 8, the training data of the first layer is \( <5, 6, 7, 8> \) and that of the second layer is \( <2, 4, 6, 8> \).

2) The three-layer BP neural network in the right uses the general method to train and predict. During the prediction it accepts forward propagation feature of RNN. When solving gradient, it delivers error terms to RNN model.

3) The standard forward propagation method is as follows. We use \( S_k \) to denote the value of cyclic layer at time \( k \) and \( S_k = G(\text{net}_k) \) stands, where \( G() \) is activation function. The vector \( \text{net}_k \) denotes the weighted input of cyclic neural cell at time \( k \), which is computed as \( \text{net}_k = Ux + WS_{k-1} \), where \( W \) is cyclic weight.

4) Now we introduce multiple layer forward propagation approach. For convenience of description without loss of generality, we present our training method taking two-layer embedded RNN prediction model as an example. Fig. 4 shows the process of model training. When the upper layer \( G_1 \) finishes training, the lower layer \( G_0 \) are then trained. The layer \( G_0 \) not only uses standard forward propagation method, also uses the following equations when involving layer \( G_1 \).

\[
\text{net}_1 = U\bar{x}_0 + \text{net}_{k-1}, \quad \hat{S}_{k-1} = G(\text{net}_{k-1}), \quad \text{net}_k = U\hat{x}_0 + WH_{k-1}, \quad G(\text{net}_k) = \hat{S}_k = G(\text{net}_k),
\]

where the dotted symbols denote the corresponding parameters in layer \( G_1 \), and \( \hat{x}_0 \) uses the average value of series data, i.e., \( \hat{x}_0 = \{ \bar{x}_i | x_i \in X \} \).

5) Now we present multiple layer error back propagation method. We use standard back propagation way of RNN error item \([5]\) when training layer \( G_1 \). While training layer \( G_0 \), we not only use the method in paper \([5]\), also utilize the following equations to propagate errors when involving layer \( G_1 \).

\[
(\delta_{k-1})^T = (\delta_k)^T W \cdot \text{diag}[G'(\text{net}_1)], \quad (\delta_{k-1})^T = (\delta_k)^T W \cdot \text{diag}[G'(\text{net}_k)], \quad (\delta_k)^T = (\delta_k)^T W \cdot \text{diag}[G'(\text{net}_k)],
\]

where \( \delta \) is error item of each cyclic period and \( \text{diag}[x] \) denotes a diagonal matrix created by vector \( x \).

6) Next, we compute the gradient between error function \( E \) and weight matrix of period \( k \), which is denoted as \( W_k \). Layer \( G_1 \) use the classical equation in our paper \([5]\), and the gradient of layer \( G_0 \) is computed as

\[
\nabla_{W_k} E = \begin{bmatrix}
\delta_{k,1} \hat{S}_{1,1} & \delta_{k,1} \hat{S}_{1,2} & \cdots & \delta_{k,1} \hat{S}_{1,n} \\
\delta_{k,2} \hat{S}_{1,1} & \delta_{k,2} \hat{S}_{1,2} & \cdots & \delta_{k,2} \hat{S}_{1,n} \\
\vdots & \vdots & \ddots & \vdots \\
\delta_{k,n} \hat{S}_{1,1} & \delta_{k,n} \hat{S}_{1,2} & \cdots & \delta_{k,n} \hat{S}_{1,n}
\end{bmatrix}
\]
where $\delta_{i,t}$ denotes the $i$-th component of error item $\delta_t$, and $S_{j,t-1}$ denotes the output valued of the $j$-th neural cell of $S_{t-1}$. Note that $S_{t-1}$ is the sub module of $\delta_t$ in the left.

The gradient of cyclic weighted matrix $W$ is the sum of the gradient at each moment, and computed as $$\frac{\partial E}{\partial W} = \nabla_w E = \sum_{i=1}^{k} \nabla_{w_i} E.$$ 

After finishing model training, the model is sent to result prediction module.

### 3.4. Feature Extraction Module

In the module web crawler technology is used to crawl sci-tech literature from web and database. Then we use our proposed weighted TF-IDF algorithm and deep Boltzmann machine method [5] to vectorize text data and extract their features. At last we get a series data set $X$, where $X$ is the input of model training and result prediction modules.

### 3.5. Result Prediction Module

The module inputs the newest data, uses the trained model to predict hot researches and send the results to users, which is the same to all the prediction models.

### 4. Experiments

In order to validate the performance of our proposed embedded RNN model. We use three servers to deploy our environment. In view of the particularity of electricity industry, our internet is divided into internal and external power networks, and they are physically isolated. One server is to crawl intranet data, one server is to crawl extranet data and the other server is to deploy the system and the prediction algorithm. We collect the data from Jan. 2015 to Dec. 2018. The data from Jan. 2015 to June. 2017 are used with labels and the other data are used to predict. We set two comparison metrics, which are precision ratio and training time.

We first compare our embedded RNN model with SVM, RNN and linear regression algorithms. Fig.5 shows their comparison results with different lengths of prediction period. Experiments demonstrate that our proposed embedded RNN algorithm has efficient training efficiency while ensuring precision ratio. For RNN algorithm, it has a high precision ratio. However, when time span is 200 days, its training time is too long to return prediction results to users in time.

We next analyze the performance with different numbers of embedded layers. In Fig. 6, 1L, 2L and 3L respectively denote the number of model layers. The prediction time spans change from 100 to 400. Assuming that time span is set to $s$, the period of 1L RNN is $s$, the period of 2L RNN is $\sqrt{s}$ and the period of 3L RNN is $\sqrt[3]{s}$. From the figure we can see that they have similar precision ratio. For training time, 2L and 3L embedded RNN are better than 1L.

![Figure 5. The comparison results of different algorithms with prediction time spans](image-url)
Figure 6. The comparison results of different numbers of layers with prediction time spans

5. Conclusion
In the paper we propose an embedded RNN prediction model, which has a quick training efficiency. It trains the real-time model using the newest data and could returns prediction results rapidly and accurately. The proposed approach could also predict electric power hotspots for arbitrary time ranges, which is flexible and elastic. Its resilience provides researching directions for electric power researches of different demands. It is experimentally shown that RNN has the most accurate precision ratio and linear regression has the least training time, but their other performance indicator is the worst. Our proposed embedded RNN model is proper for heterogeneous time span scenario.

Acknowledgments
This work was supported by Science and Technology Plan for Colleges and Universities in Shandong Province (KJ2018BAN046).

References
[1] Su Xinning. The Return of Informatics and Intelligence Work in Big Data Era [J], Journal of the China Society for Scientific and Technical Information, 36 (4): 331-337, April 2017.
[2] Wu Lulu. Research Hotspot Situation Analysis Based on Topic Model [D], University of Electronic Science and Technology of China, 2018
[3] Hochreiter s, Schmidhuber J. Long Short-Term Memory Neural Computation [J], 1997, 9(8): 1935-1970.
[4] Sainath T. N., Vinyals O., Senior A., etc. Convolutional long short-term memory, fully connected deep neural networks [C], Proc of IEEE International Conference on Acoustics, Speech and signal Processing. Brisbane:IEEE, 2015:4580-4584.
[5] Ma Yan, Qi Dali, Yu Naihai, Zou Lida. A Prediction and Push Method of Scientific Researching Issue Based on RNN [C], IOP Conf. Series Materials Science and Engineering, 452 (2019) 042079