International Core Journal of Engineering Volume 7 Issue 6, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

Fast Wind Power Prediction Method Based on Time Convolution Network

Shaoyang Huang

School of Control and Computer Engineering, Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, Baoding 071003, Hebei, China.

Abstract

LSTM, GRU and RNN are all deep learning models of cycle structure, which are the main methods of wind power prediction. But the cyclic structure requires wind power time series data to be calculated one by one in chronological order, which leads to poor timeliness of the model, and is difficult to adapt to wind power forecast scenarios under the background of big data. This paper proposes a fast wind power forecasting method based on time convolutional network. In this method, the model discards the loop structure and improves the convolution operation through one-dimensional convolution, causal convolution and cavity convolution to achieve fast training and accurate prediction, which leads to the model can process data in parallel, has a robust gradient, and provides a flexible receptive field. This article uses the open data set provided by Kaggle, and carries out prediction accuracy comparison experiments, training convergence comparison experiments, and training time-consuming comparison experiments. The experimental results show that the prediction accuracy of the improved model in this paper is slightly higher than that of LSTM, GRU, and RNN, and the training process is easier to converge, and the training time is less than that of LSTM and GRU.

Keywords

Wind Power Forecasting; Time Convolution Network; Time Series Analysis; Fully Convolutional Network.

1. Introduction

Wind power prediction is of great significance to wind field management and power operators. Wind power prediction in wind field management is mainly for power market bidding. Wind power as grid-connected power needs to participate in market bidding. The accuracy of wind power prediction directly affects the economic benefits of wind field. The purpose of wind power prediction by power grid operators is to balance the power of the whole network, and accurate prediction can ensure the safe and stable operation of the power grid system. The difficulty of wind power prediction lies in the large volatility and randomness of wind resources, which directly leads to the instability of power generation. From the data set used in this paper, we can see that the difference between the maximum and minimum power generation in 2018 is more than 50 times. The unstable state of wind resources undoubtedly brings difficulty to wind power prediction.

Wind power prediction methods are usually divided into traditional methods and depth learning methods. Traditional Wind Power Forecast[1] Methods include statistical methods and classical machine learning methods. Common statistical methods include: differential integrated moving average autoregressive model[2] Monte Carlo model, Kalman filter, empirical mode decomposition, etc. Statistical methods are often used in the data preprocessing phase. Classic machine learning

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

methods include support vector machines[3] (SVM), artificial neural network (ANN), etc. By combining traditional algorithms, good prediction results can be obtained. Literature[4] The prediction accuracy of wind power is improved by combining three algorithms: ANN network, RBF network and SVM. The traditional wind power prediction method is easy to realize and explain, and is suitable for small samples.

With the advent of the big data era, the volume and dimension of data become larger and the advantages of deep learning methods become more and more prominent. In recent years, the rapid development of deep learning, the use of deep learning method for wind power prediction results. Among the many methods of deep learning, circular neural networks[5] As one of the classical algorithms, the input end of the RNNs receives the current wind power data and the state vector of the previous moment. This design enables the loss function to correlate the information of each time node of the sequence. RNNs in reverse propagation[6] Gradient dispersion / explosion is easy to occur in the process. In order to alleviate this phenomenon, the gated circulation unit is added to the original circulation structure to obtain LSTM. The[7] GRU algorithm. Literature[8] prove that using LSTM alone can achieve good results in short-term wind power prediction. Wind power prediction hopes to find the mapping relationship between each dimension information and power generation from historical data, so wind power prediction uses wind power historical data from the previous period to predict the power generation at the next moment. The original intention of RNNs design is naturally in line with wind power prediction.

RNNs model is usually used in wind power prediction, and there are three common ways to improve it.

(1) Optimization of feature extraction for input wind power timing data. Paper[9] EEMD algorithm is used to analyze and filter the wave of wind power sequence data. Improve the data quality in the data preprocessing link, so that the wind power sequence data is optimized before entering the LSTM.

(2) Reprocessing of RNNs output feature vectors. The prediction process is to map the feature vector to the specific generation power, and it is very important to deconstruct the feature space reasonably. Paper[10-12] All are based on LSTM output improvement.

(3) Introduction of attention mechanisms. In wind power prediction, the influence of each time node on prediction is obviously different. The attention mechanism can highlight the importance of key time nodes by assigning different weight values to different feature vectors. At the same time, the influence of non-critical time nodes is weakened. And the attention mechanism overcomes the problem of RNNs short distance dependence to some extent. Literature[13] GRU combined with attention mechanism is used to predict wind power, and good results are obtained. Literature[14] The attention mechanism and activation function were improved. However, as long as the cyclic structure is adopted, its recursive nature determines that the model can not process data in parallel. The cyclic structure requires that the current data must wait until the previous data processing is completed before it can be calculated. Convolution operation makes up for this shortcoming. Convolution operation is a calculation method of sliding superposition summation. The size of convolution kernel automatically matches the dimension of wind power timing data. Tensorflow, café, pytorch these three mainstream frameworks, the underlying code implementation of the convolution process is a matrix operation, so the whole convolution process is logically calculated in the form of a sliding window. But the calculation process at the code level is similar to the fully connected matrix operation, which makes the convolution operation very fast. At the same time, convolution operation has strong feature extraction ability. It is fast and efficient to mine the underlying information of wind power timing data. Convolution operation has been used in wind power prediction. Literature[15] A prediction method combining convolution operation, LSTM and attention mechanism is proposed. Literature[16] Combined with GRU and convolution operation, wind power prediction is carried out. In the above paper, convolution operation is only used as feature extractor or filter, and the prediction model has no farewell RNNs, which leads to the model always calculating wind power timing data in the form of queue.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

In order to process wind power timing data in parallel in wind power prediction task, the time convolution network designed in this paper[17] The wind power prediction model abandons the cyclic structure by one-dimensional convolution and causal convolution[18]And cavity convolution[19]The basic convolution operation is improved and the residual structure[20] is used The interlayer conveys information. The feature extraction layer of the network is realized by the full convolution network, and the output layer is composed of the full connection. Model prediction accuracy is slightly higher than RNNs, and fast training and prediction are realized.

2. Wind Power Prediction Based on Deep Learning

The wind power prediction based on deep learning takes the wind power timing data as the input data, and finally obtains the power generation at the next time through the model calculation. The model can be divided into three stages: data preprocessing, feature extraction and feature vector decoding.

2.1 Data preprocessing phase

Data preprocessing is the processing and transformation of raw wind power data, which makes it a data structure that meets the requirements of wind power prediction model. Wind power timing data is the product of raw wind power data after data preprocessing. as the raw wind power data, it should cover as many factors as possible that can affect the power generation (e.g., wind speed, wind direction, fan real-time data, etc.). the data from different sources are summarized and arranged in chronological order to form a two-dimensional table. the data structure can be represented as (times, features), where times represents the time node, features for each dimension information. The process of data preprocessing can be divided into four steps: data cleaning, feature analysis, data standardization, and construction of wind power timing data.

(1) Data cleaning. Wind power raw data is often accompanied by a large number of invalid data (null value, invalid value), take the data set used in this paper as an example, where invalid data accounts for 6.5% of the data, cleaning data is very necessary.

(2) Feature analysis. In order to better analyze the change law between wind resources and power generation, the original data of wind power will often be analyzed. Common methods such as data dimensionality reduction, wavelet analysis, data dimension expansion and so on.

(3) Standardization of data. Because of the difference of dimensionality between each dimension, each dimension will reflect obvious span numerically, and the span will have adverse effect on model prediction. Before constructing the timing data of wind power, it is necessary to perform scaling operation on the original wind power data. The common feature scaling methods are standardization, mean and normalization.

(4) Construction of wind power timing data. By sliding window, the original wind power data is divided into equal length, which is constructed into wind power timing data, and the data structure can be expressed as (samples, step, featrues). The samples is sample size, the step is the step size of wind power timing data, and the features is the information of each dimension.

2.2 Feature extraction phase

Wind power timing data contains important information to be explored. The degree of discovery of these information is directly reflected in the final effect of the model, and the feature extraction work can be regarded as the coding of wind power timing data. The feature extraction will map the wind power timing data to the high dimensional feature space. The calculation of each layer of neural network is a re-mapping of the previous layer of feature vectors. The feature vector is an abstraction of the high dimension of the original data. After the network calculation, the feature vector will show certain distribution characteristics in the new feature space.

Based on the characteristics of wind power timing data, the main difficulties in feature extraction are as follows.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

(1) Whether the selected feature extractor can effectively discover the information contained in the wind power timing data is related to the design of the feature extractor.

(2) In the process of feature extraction, some wind power timing data may be lost. Especially in the wind power timing data arranged behind the data, after many mapping, often easily forgotten. Different feature extractors have different working principles, but the problem of information loss is common.

(3) The new feature space has the problem of poor distribution structure or sparse distribution. The quality of feature vector is directly reflected in the quality of feature vector, and the poor quality of feature vector is very unfavorable to the subsequent decoding work.

(4) The structure of the feature extractor is not easy to reverse the gradient propagation, which makes the model prone to gradient dispersion / explosion.

2.3 Feature vector decoding stage

After the wind power timing data is extracted, a feature vector will be obtained, and the prediction is the decoding of the feature vector. In the decoding process of wind power prediction, the high dimensional feature vector is mapped to a specific generation power value. Comparing the space distance between the predicted power value and the actual power value is the standard to evaluate the advantages and disadvantages of the model, and it is also the basic idea of the design of the model loss function.

3. Wind Power Prediction Model Based on Time Convolution Network

3.1 Model structure

The wind power prediction model based on time convolution network designed in this paper includes data preprocessing, feature extraction layer and output layer, as shown in figure 1.

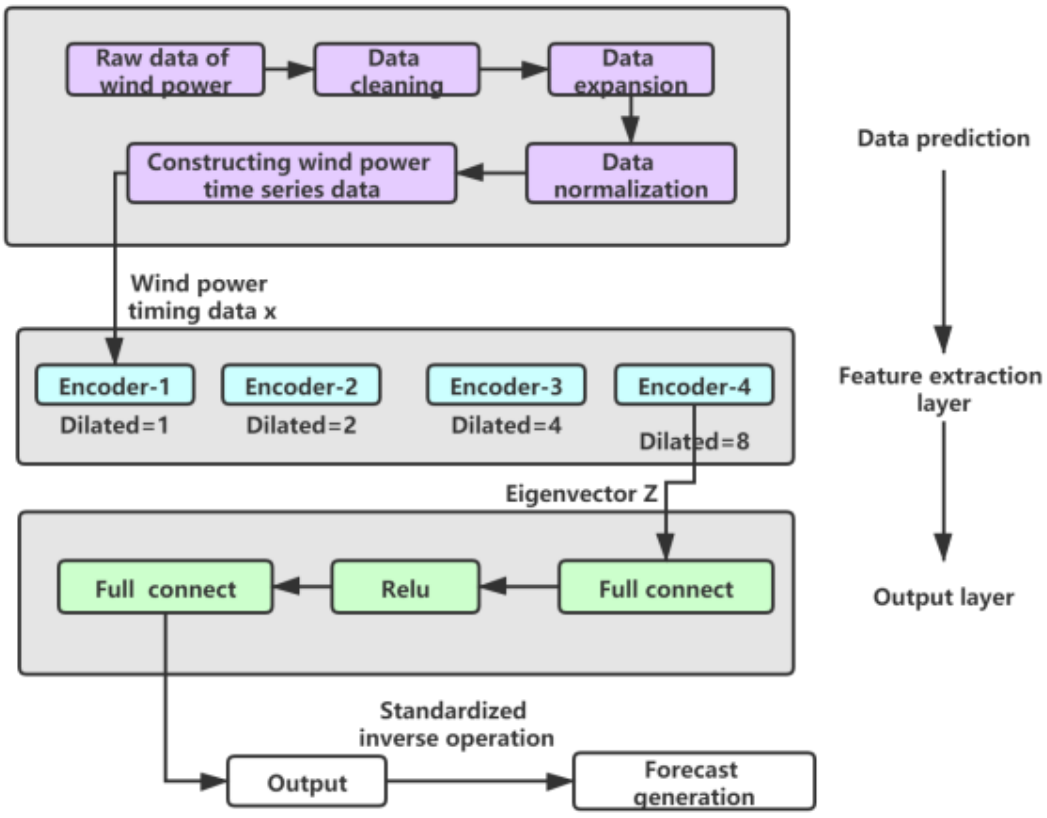


Figure 1. Wind power forecasting model based on time convolution network

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

The Encoder is a feature coding module; the Full connect is a full connection; the Relu is an activation function; the Dilated is an expansion coefficient; and the output is a model output. Wind power raw data will be calculated according to the flow shown in the diagram.

Data preprocessing includes cleaning, expanding, standardizing and constructing wind power timing data.

The feature extraction layer is responsible for the feature extraction of the input wind power timing data. It consists of four feature coding modules with the same structure but different parameters. Encoder layers are input and output, different Encoder layers will give different expansion coefficient, and the expansion coefficient of each layer will increase exponentially by 2. If the expansion coefficient is larger, the larger the receptive field is, the smaller the expansion coefficient is used in the bottom Encoder, and then the receptive field increases gradually, which to some extent prevents the loss of information. The feature extraction layer is a deep neural network based on convolution, which is a full convolution network. Convolution operation is a logical sliding calculation of data, which seems to process wind power timing data in time order, but in the bottom operation, the moving process of convolution kernel is integrated into a matrix operation. when it is necessary to synchronize the calculation of multiple batches of wind power timing data, the pytorch depth learning framework will configure the convolution kernel size according to the size of the input data tensor, so the convolution operation realizes the parallel processing of the data. For wind power timing data, the feature extraction layer of this model can realize synchronous calculation of each timing node.

The output layer decodes the output vector of the feature extraction layer, which consists of two full connections and an activation function. Because the data is standardized in the preprocessing stage, the model output needs to be mapped to the predicted power generation by standardized inverse operation.

3.2 Feature extraction layer

Encode r is a feature coding module based on convolution principle, and multiple Encoder are combined into a feature extraction layer. Encoder structure is shown in Figure 2.

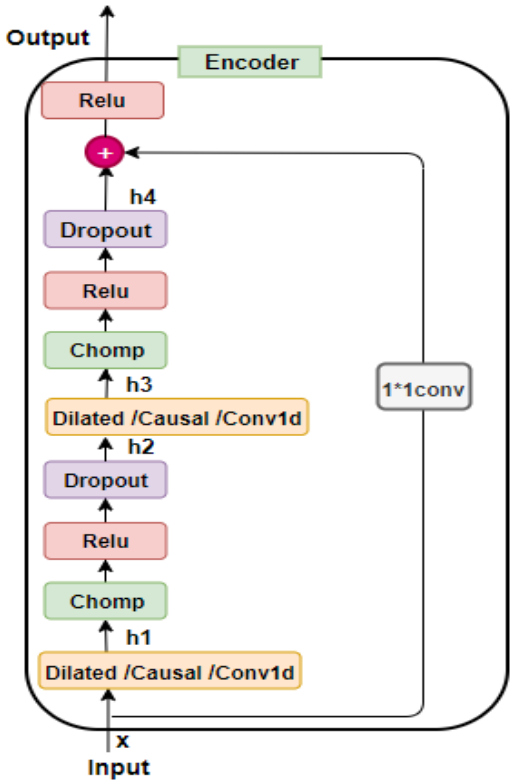


Figure 2. Structure of Encoder

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

Within the Encoder are convolution layer (Dilated/Causal/Conv1d), clipping layer (Chomp), activation function layer (Relu), Dropout layer () and residual structure (1*1 conv).

Encoder calculation flow is expressed as formula group (2-7):

$$h_1 = \ （X_d * f）(s) = \sum_{i=1}^{k-1} f(i) \bullet P(X)_{s-d*i} \tag{1}$$

$$h_2 = Dropout(Re\,L\,U(C\,hom\,p\,(h_1))) \tag{2}$$

$$h_3 = \ （X_d * f）(s) = \sum_{i=1}^{k-1} f(i) \bullet P(h_2)_{s-d*i} \tag{3}$$

$$h_4 = Dropout(Re\,L\,U(C\,hom\,p\,(h_3))) \tag{4}$$

$$h_5 = \sum_{k=0}^{2}\sum_{h=0}^{2} F(X,\ f)X(1-h,1-k) \tag{5}$$

$$Z = Re\,l\,u(h_4 + h_5) \tag{6}$$

Formula group: P() for convolution filling operation; Dropout() as Dropout function; Relu() as activation function; Chomp() is a clipping function; d coefficient of expansion; k is filter size; i sequence subscript; X as input vector; f as convolution kernel; The h1, h2, h3, h4 is an intermediate hidden vector; the h5 is a 1/1 convolution result; the Z is an output vector.

The feature vector loops through the convolution layer, the clipping layer, the activation function layer and the Dropout layer, and the input feature vector is transmitted to the output through the convolution kernel residuals of 1*1. finally, the Relu activation is carried out again.

3.2.1 Convolutional layers

The convolution layer plays a key role in feature extraction, and its structure is composed of one-dimensional convolution, causal convolution and cavity convolution.

(1) One-dimensional convolution means that the data structure of the convolution kernel is one-dimensional. Two-dimensional convolution can process three-dimensional data structure, wind power timing data is two-dimensional structure, so one-dimensional convolution is used. One-dimensional convolution operation can play the role of feature extraction of wind power timing data. Taking wind power timing data as input and convolution calculation through multi-channel one-dimensional convolution kernel, the internal characteristics of multi-dimensional data can be extracted. Convolution operation has strong information capture ability. The values of convolution kernel are obtained by neural network learning. Adaptive convolution kernel can better adapt to specific application scenarios.

(2) The use of causal convolution ensures the delay in the time dimension, and the principle of causal convolution is shown in formula (7):

$$（F * X）_{(x_t)} = \sum_{k=1}^{K} f_k x_t - K + k \tag{7}$$

The formula is: $F = (f_1, f_2, \ldots, f_K)$ filter; K is the number of convolution kernels; k is convolution kernel; t is time node; X is wind power timing data.

Causal convolution is an application of one-dimensional convolution. Usually, according to the different correlation between nodes before and after the data, the sequence data can be divided into unidirectional association and bidirectional association. Wind power timing data is arranged in chronological order and is a strict one-way dependency. The use of causal convolution ensures this one-way dependency.

(3) Cavity convolution refers to the insertion of zero values at certain intervals in convolution kernels. The working principle of cavity convolution is shown in figure 3.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

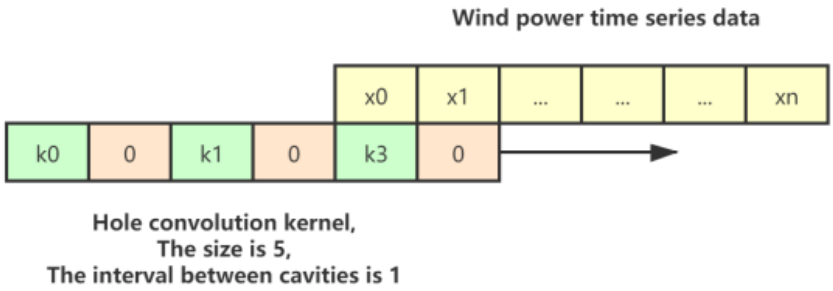


Figure 3. Dilated Convolution

The interval between inserted zero values is a parameter that can be adjusted, called the coefficient of expansion. Encoder the same expansion coefficient is used in the model, the first layer expansion coefficient is 1, and then the expansion coefficient of each layer increases exponentially with 2. Cavity convolution enables convolution kernels to be applied to regions larger than the length of the filter itself. The frequent use of convolution operations in neural networks will lead to the loss of internal data structure, spatial hierarchical information loss, small target information can not be reconstructed and so on. Cavity convolution operation can make the model have a very large receptive field when the number of layers is small. Paper [21] It shows that cavity convolution makes the model have flexible receptive field.

3.2.2 Cutting layer

Before the convolution operation of this model, the feature vector is filled, so the size of the output vector will change after the wind power timing data is convolved. The clipping layer cuts the tail of the output vector, and after the clipping layer, The input vector is consistent with the output vector size.

3.2.3 Activation function and Dropout

Activation functions and Dropout operations are routine operations for deep learning. The activation function realizes nonlinear transformation, and the Relu activation function is easier to calculate than the sigmoid, tanh activation function. When the value is large, the Relu has obvious advantages as unsaturated activation function. The working principle of Dropout is to randomly shield some neural network nodes. On the one hand, the use of Dropout can prevent the synchronization dependence of nodes, on the other hand, it can also prevent overfitting.

3.3 Residual Deconstruction

The residual structure can optimize the gradient conduction in reverse propagation, and at the same time, it will transmit the lower layer information to the upper network, and the information transfer layer can alleviate the loss of the bottom information to some extent. If the neural network simply increases the depth, it will lead to gradient dispersion / explosion. The gradient problem can be alleviated to some extent by regularization, but the accuracy saturation will inevitably decrease.

Residual network realizes cross-layer connection through Skip Connection operation. Residual transmission can avoid the problem that the gradient is too small to return in the process of reverse propagation. Taking the lower layer features to the upper layer can enhance the accuracy, but connecting the lower layer feature map jump layer directly to the upper layer will lead to the inconsistency of the corresponding feature map channel number. This model uses 1/1 convolution to reduce dimension.

3.4 Output Layer

The input of the output layer is the output of the feature extraction network, which is realized by two full connections and one nonlinear transformation. The output layer formula is shown in formula (8):

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

$$y = w_2 \left(\sigma(w_1 * z + b_1)\right) + b_2 \quad (8)$$

σ The w_1, w_2 is a parameter matrix, the b_1, b_2 is a bias term, and the activation function Relu; y the prediction value of the final generation power. The core operation of full connection is matrix vector product, which plays the role of mapping from feature vector to power generation.

After data preprocessing, the data structure needed by the model can be constructed. The feature extraction layer will mine the feature information of wind power timing data, and the output layer realizes the nonlinear mapping between the feature vector and the predicted power generation.

4. Experiment

4.1 Loss function and optimizer

Model training is the process of updating model parameters. The design of loss function and the selection of optimizer are very important.

(1) The loss function is the optimization direction of model training. The loss function reflects the relationship between the predicted value and the actual value. The smaller the loss value, the better the prediction result. When the training model propagates in reverse, the loss function moves in the direction of decreasing the loss value. The gradient descent method updates the parameters in the model by iteration. A classical mean square error (MSE) loss function is used in this model. The mean square error function expression is shown in (9) as follows:

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \quad (9)$$

The n is the number of samples of wind power sequence data, the actual generation power value, the generation power prediction value and the Loss loss value. $y_i \hat{y_i}$

MSE values can intuitively reflect the spatial distance between predicted and actual power generation.

(2) The optimizer acts to update parameters and calculate gradients. The model finds the local optimal solution through the optimizer, which minimizes the loss function. Adam of Model Training Selection [22] The model is updated as an optimizer. Adam the algorithm is different from the traditional stochastic gradient descent, the traditional stochastic gradient descent maintains a single learning rate. according to the product of the learning rate and the gradient, the weight is updated, and the learning rate will not change during the training process. and Adam design independent adaptive learning rates for different parameters by calculating the first and second moment estimates of the gradient.

4.2 Data sources and processing

The data used in this paper are provided by Kaggle and collected from the scada system of wind turbines operating and generating electricity in Turkey. The data set sampling time interval is 10 min, a total of 50530 time section data, including real-time generation power, wind speed, theoretical power curve, wind direction four dimensions information. The four dimensions of information from the physical level are the main factors affecting wind power[23].

After the wind power raw data clears the empty value and the invalid value value value, the valid data is 47268, all the data form a two-dimensional table shape (472684), which is divided into two dimensions. Part of the wind power raw data in the dataset on March 1,2018 is shown in figure 4.

Figure 4 shows that the original wind power data because of different dimensions, the numerical span of each dimension data is large. The average power generation in figure 4 is 1476.4, while the average wind speed is 7.5. If the dimension is not taken into account, the power generation is 197 times the wind speed from a numerical point of view. If the wind power raw data is not unified dimensional operation, it will lead to the model training difficult to converge.

In order to better reflect the fluctuation of wind resources and power generation, the model expands the mean and variance of wind speed and power generation according to different time series steps. Based on wind power raw data, A total of 16 dimensions were expanded. Two dimensions of electricity generation and wind speed, Moving the cutting sequence to a window value of 5,10,15,20,

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

Then the mean and variance are calculated for the cut sequence, Put it into the corresponding position of wind power timing data. These 16 dimensions can be expressed as: power generation mean _5, electricity quantity mean _10, electricity quantity mean _15, electricity quantity mean _20, electricity quantity variance _5, electricity quantity variance _10, electricity quantity variance _20, wind speed mean _5, wind speed mean _10, wind speed mean _20, wind speed variance _5, wind speed variance _10, wind speed variance _15, wind speed variance _20. After expansion, Wind power raw data from 4 to 20 dimensions.

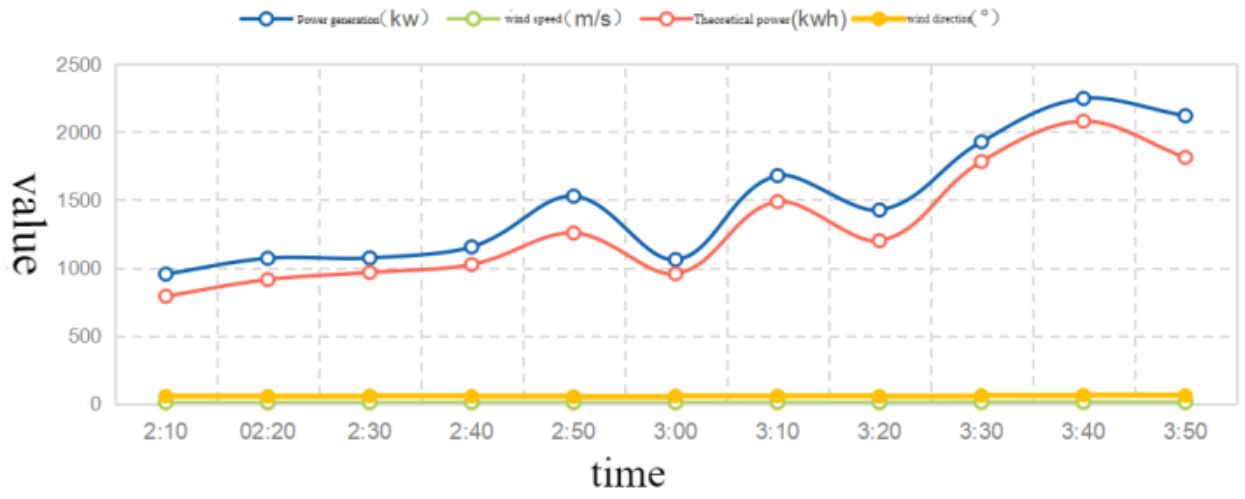


Figure 4. Wind power raw data

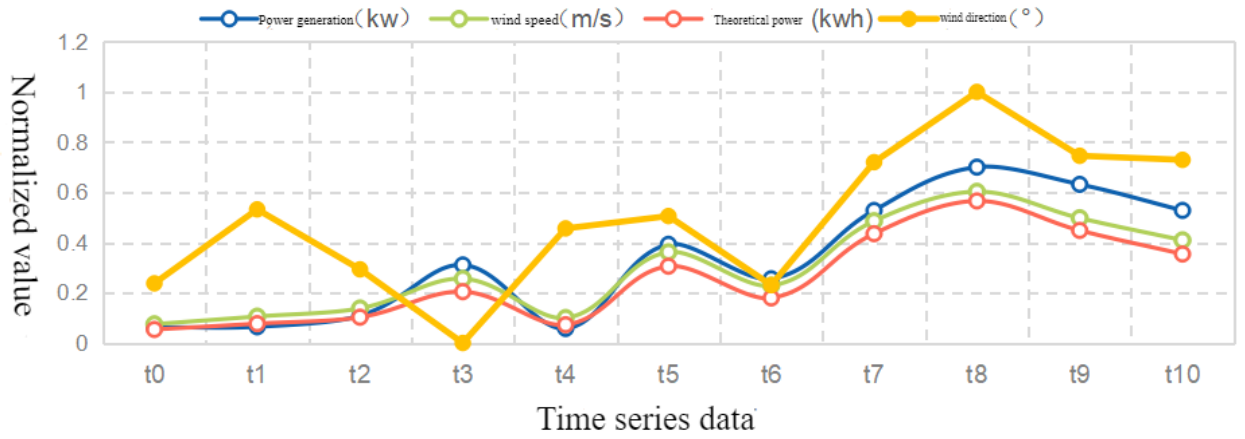


Figure 5. Standardization data

This model adopts min-max standardization, and through min-max standardization, wind power data from different sources are unified to similar scales. min-max standardization is shown in formula (10):

$$x^{\sim} = (x - x_{min})/(x_{max} - x_{min}) \quad (10)$$

Formula: x~ Value after standardization; x wind power raw data; x_{max} and x_{min} The maximum and minimum values, respectively.

min-max standardization is a linear transformation, which is essentially the scaling and translation of the data. It does not change the numerical ranking of the original data. After standardized processing, the data are all mapped to the unified scale range. The original wind power data are standardized as shown in figure 5. Figure 5 shows that each dimension of data is compressed between [0,1].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

Finally, the wind power timing data sample set is constructed by sliding window, and the wind power timing data samples are randomly scrambled, divided into training sample set and prediction sample set according to 8:2 ratio, and the data of January 1,2018 is selected as the verification set.

4.3 Experimental results and assessment

In order to comprehensively evaluate the performance of this model, the prediction accuracy, training convergence and training time are compared and tested. Finally, the prediction effect of the data set is demonstrated. The experiment adopts the pytorch framework of python language, which is GPU as the

4.3.1 Comparison experiment prediction accuracy of models

Prediction accuracy is an index to evaluate the prediction effect of the model. Finally, the root mean square error (R_{MSE}) between the predicted power generation and the actual power generation is calculated by using the wind power timing data in the test set as input Value. the experiment uses RNN, LSTM, GRU to compare with this model and sets four different sets of timing steps (step) with lengths of 5,10,15,30. RMSE adopted To evaluate the prediction results, the mathematical expression of the mean square error is shown in formula (11):

$$\varepsilon_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[p\hat{}\ (i) - p(i)]^2} \tag{11}$$

$p(i)$ $p\hat{}\ (i)$ Formula: the actual generation power and the predicted value, the n is the number of predictive verification data, and the prediction point sequence number. i R_{MSE} The smaller the value, the better the prediction effect. The model prediction effects are shown in Table 1:

Table 1. RMSE errors were predicted with different time steps m/s

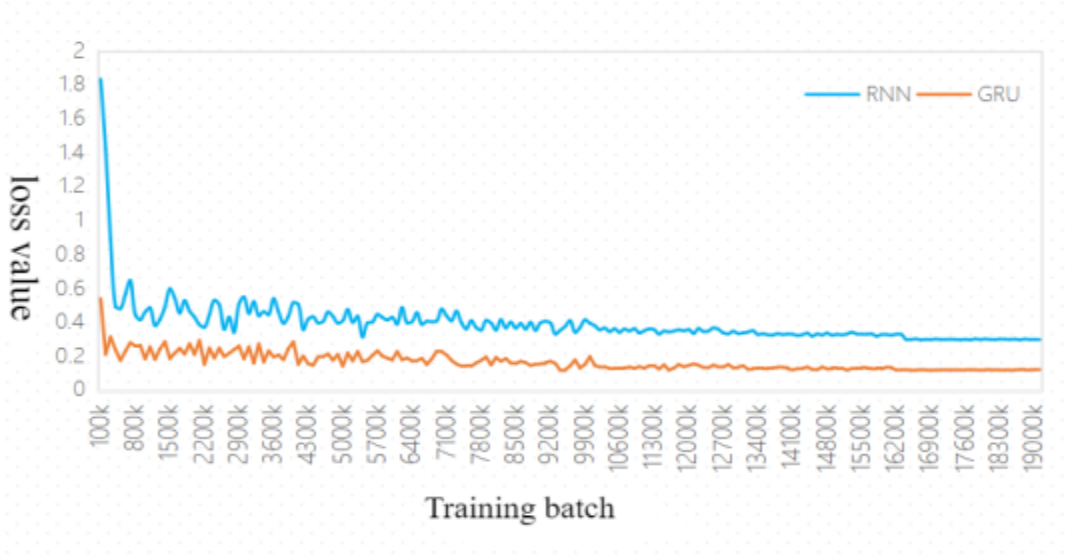
Timing step	GRU	LSTM	DNN	Model
5	99.28	109.43	119.39	127.18
10	49.48	46.51	48.04	61.48
15	48.52	47.16	50.53	44.91
30	48.28	46.2	44.92	45.11

Experimental results show that the prediction error of the proposed model is the lowest in the 4 model, and the R_{MSE} of the model in this paper Reduced GRU, LSTM and RNN by 10.16%, 9.83%, 59.27%. When the order step is 10:00, LSTM and GRU reach the best value. As the length of the sequence increases, The RMSE value of the LSTM, GRU increases, This is related to the short memory cycle of the circular structure, Can judge RNNs do not have the ability to deal with long timing. And the R_{MSE} of the model used in this paper However, the value increment is not obvious, which indicates that the model has better ability to deal with long time series than LSTM, GRU.

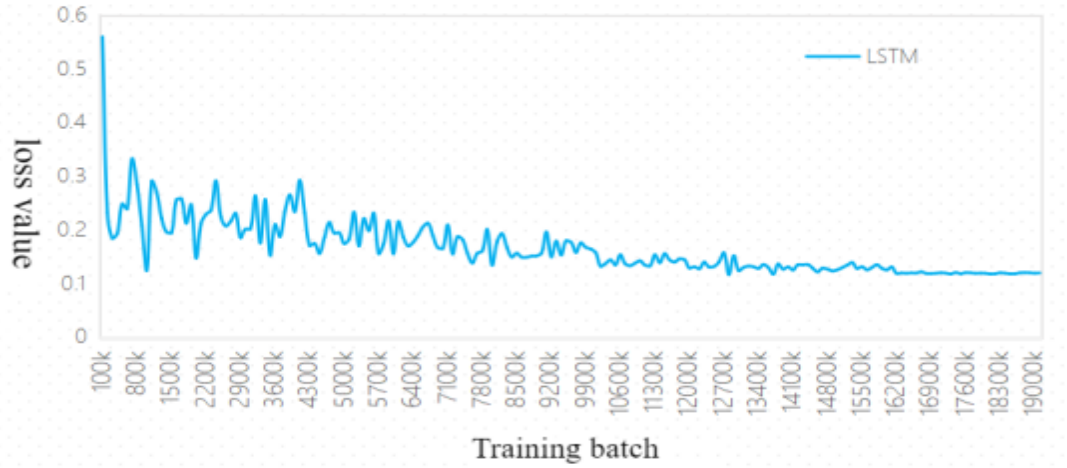
The loss curve can evaluate the convergence of the loss value of the model during training. According to the loss curve, we can see whether the model has overfitting or not fitting. The loss curves of each model are shown in figure 6.

When this model calculates the loss value curve, the wind power timing data in the training set is used as the input, the parameters are updated continuously through gradient descent and the loss value is calculated according to the loss function, and the loss value is recorded once every five samples are calculated. By observing figure 6, it can be found that the RNN has the largest loss value in the three models. The LSTM and GRU loss curves are similar. Compared with other models, the loss curve of this model fluctuates less and converges earlier. It shows that the model is easier to converge and more robust in training.

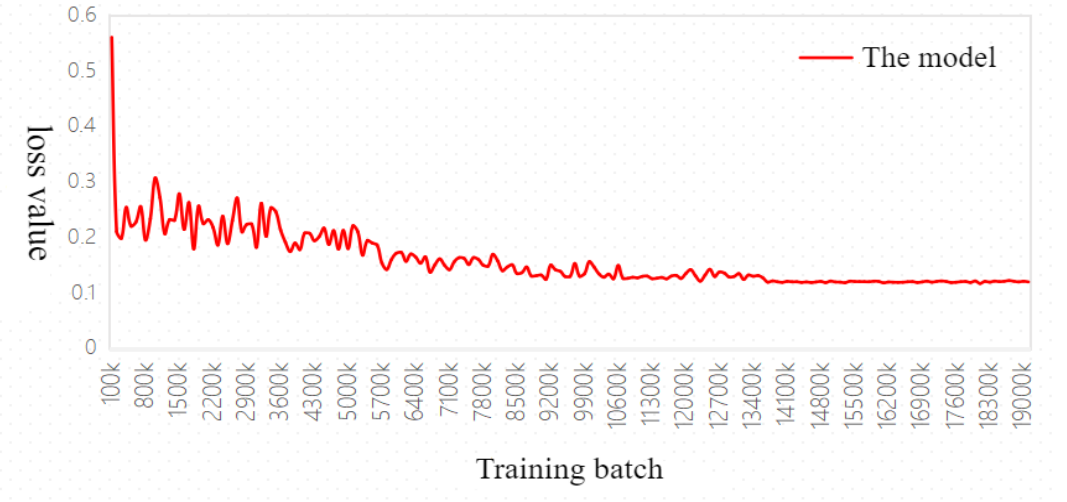
ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031



(a) RNN and GRU



(b) LSTM



(c) The model

Figure 6. Comparison of loss curve of each model

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

4.3.2 Comparison of Training Time Models

The training time comparison experiment is to compare the time-consuming comparison between parallel processing data and queue processing data. Two contrast methods are designed. The first is to compare the training time of the model with different time series steps. The second is to compare the training time of each model under the large amount of data. The step and ba t ch parameters are modified respectively, in which the batch is batch size. experiments were performed with the same amount of data and the same number of cycles.

(1) Set the timing step size to 10,20,30,50, and 100 groups of contrast experiments, using training time-consuming as the evaluation index and comparing with LSTM, GRU. The experimental results are shown in Table 2.

Table 2. Comparison of training time for different time series step size of each model (s)

Timing step	GRU	LSTM	This model
10	14.5	14.8	11.1
20	17.3	18.3	11.4
30	22	21.8	11.9
50	29.3	29.1	12.3
100	46.5	46.9	12.5

From the experimental results, the time-consuming of LSTM in different sequence lengths is 2.91 times, 2.55 times, 1.96 times, 1.64 times, 1.33 times, respectively. GRU is 2.88 times, 2.57 times, 1.98 times, 1.55 times, 1.33 times of this model. The experiment shows that the training time of this model is better than that of LSTM and GRU. at different time series steps With the increase of timing step size, the training time of LSTM and GRU increases exponentially. Compared with this model, the length of time series is insensitive.

(2) The data structure of wind power timing in this experiment is :(batch, step, info), in which batch is batch quantity. Each model pair is shown in Table 3. The batch size is set to be 1,5,10,20,50.

Table 3. Comparison chart of training time for different data volume of each model (s)

Data volume	LSTM	GRU	Model
1	18.3	17.4	11.1
5	97	96	11.8
10	192	194	12.4
20	388	379	13.6
50	990	983	14.5

Batch simulate data from different sources in the case of big data. Because the cyclic structure can only receive two-dimensional data, LSTM, GRU can only input the data into the neural network in a queue.

Experimental results show that the LSTM training time is 1.64 times, 8.22 times, 15.48 times, 28.52 times and 68.27 times respectively. GRU training time is 1.55 times, 8.13 times, 15.64 times, 27.86 times, 67.79 times, respectively. The experiment shows that the training time of this model is obviously better than that of LSTM and GRU. in the experiment At different batch values, the time-consuming growth rate of this model is 6.3,11.7,22.5 and 30.6. It shows that when the model processes the data in large quantities, the training time does not increase significantly with the increase of the quantity.

4.3.3 Experimental prediction

Using the data of the reserved verification set to display the prediction effect, the raw data of wind power in that day are preprocessed to obtain the wind power timing data, and the predicted power

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

generation value is calculated by each model. The real power generation is compared with each predicted power generation. The prediction results are shown in figure 7.

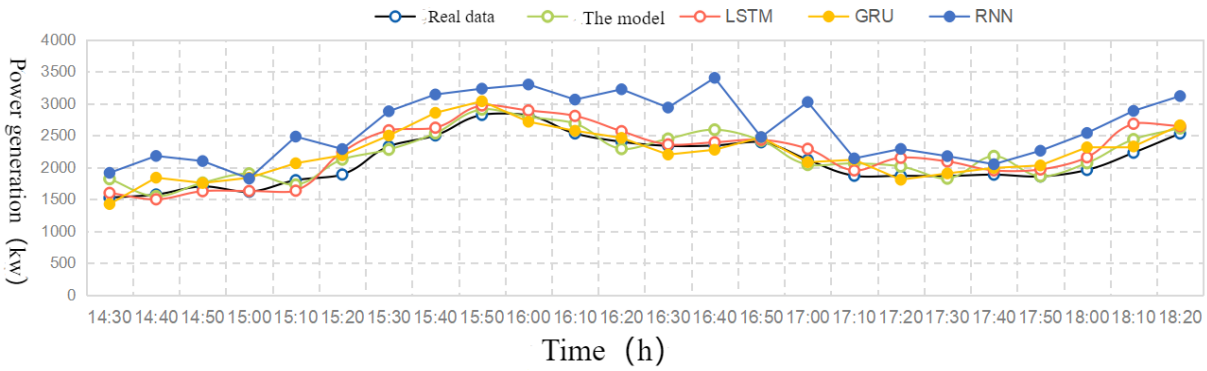


Figure 7. Comparison of prediction results of each mode

By comparing the real power generation with the predicted power generation curve of different models, we can see that the curve of the model used in this paper is smoother and better suited to the real data.

5. Conclusion

Aiming at the problem that the RNNs model can not be calculated in parallel in wind power prediction, a fast wind power prediction method based on time convolution network is proposed in this paper. In this model, the cyclic structure is abandoned and the wind power timing data are extracted by full convolution network. Experimental tests were conducted on Kaggle open data sets and the following conclusions were obtained:

(1) A feature extraction network built in a full-volume approach has good results, the prediction accuracy of this model is slightly higher LSTM, GRU and RNN.

(2) This model has the ability to process data in parallel, can process data quickly, and has good timeliness. Both long time series data and large number of times data are processed, and their training time is better than that LSTM, GRU, Therefore, this model is more suitable for large-scale wind power prediction scenarios.

(3) The model is easy to train and the gradient converges fast. Attention mechanism assigns different weight values to feature vectors to highlight important information and secondary information. When wind resources fluctuate greatly, attention mechanism can be used to deal with sudden situations. Attention mechanism can be added to the model in the future to further improve the prediction accuracy of the model.

References

[1] Han Zifen, Jing Qianming, Zhang Yankai,et al.Summary of wind power forecasting methods and new trends [J]. Power system protection and control. 2019,47(24):178-187.

[2] ELDALI F A, HANSEN T M, SURYANARAYANAN S, et al. Employing ARIMA models to improve wind power forecasts: a case study in ERCOT [C]// 2016 North American Power Symposium (NAPS), September18-20, 2016, Denver, CO, USA: 1-6.

[3] ZHANG Y, WANG P, NI T, et al. Wind power prediction based on LS-SVM model with error correction [J]. Advances in Electrical & Computer Engineering, 2017, 17(1): 3-8.

[4] LIUChun, FAN Gaofeng, WANG Weisheng, et al. A combination forecasting model for wind farm output power[J]. Power System Technology, 2009, 33(13): 74-79

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0031

[5] Qu, XY, Kang, XN, Zhang, C, Jiang, S, Ma, XD, et al.Short-Term Prediction of Wind Power Based on Deep Long Short-Term Memory [C]// 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC) IEEE, 2016.

[6] Zhu Qiaomu, Li Hongyi, Wang Ziqi, et al. Ultra short term prediction of wind farm power generation based on long and short term memory network [J].Power System Technology, 2017(12):68-73.

[7] ALH, AHJ, BRZ, et al. Wind power forecast based on improved Long Short Term Memory network[J]. Energy, 189.2019

[8] Zhu Qiaomu, Li Hongyi, Wang Ziqi, et al.Ultra short term prediction of wind farm power generation based on long and short term memory network [J].Power System Technology, 2017(12):68-73.

[9] HUANGY, LIU S, YANG L. Wind speed forecasting method using EEMD and the combination forecasting method based on GPR and LSTM[J]. Sustainability, 2018,10(10):3693-3707.

[10] YU C, LI Y, BAO Y, et al. A novel framework for wind speed prediction based on recurrent neural networks and support vector machine[J]. Energy Conversion and Management,2018.178: 137-145.

[11] LÓPEZ E, VALLE C, ALLENDE H, et al. Wind power forecasting based on echo state networks and long short-term memory[J]. Energies, 2018,11(3):526-547.

[12] OEHMCKE S, ZIELINSKI O, KRAMER O. Input quality aware convolutional LSTM networks for virtual marine sensors[J]. Neurocomputing, 2018,275:2603-2615.

[13] Zhao Bing, Wang zengping, Ji Weijia, Gao Xin, Li Xiaobing. Cnn-gru short-term load forecasting method based on attention mechanism [J]. Power System Technology, 2019,43 (12): 4370-4376

[14] LÓPEZ P R, DORTA D V, PREIXENS G C, et al. Pay attention to the activations:a modular attention mechanism for fine-grained image recognition[J]. IEEE Transactions on Multimedia, 2020,22(2):502-514.

[15] Zhao Jianli, Bai Geping, Li Yingjun, Lu Yao. Short term wind power prediction based on cnn-lstm [J]. Automatic instrumentation, 2020,41 (05): 37-41

[16] Xue Yang, Wang Lin, Wang Shu, Zhang Yafei, Zhang Ning. An ultra short term wind power prediction model combining CNN and Gru networks [J]. Renewable energy sources, 2019,37 (03): 456-462

[17] Shaojie Bai, J. Zico Kolter, Vladlen Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling,arXiv:1803.01271v2[cs.LG] 19 Apr 2018

[18] Oord A V D, Dieleman S, Zen H, et al. WaveNet: A Generative Model for Raw Audio[J]. 2016.

[19] Kudo Y, Aoki Y. [IEEE 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA) -Nagoya, Japan (2017.5.8-2017.5.12)] 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA) -Dilated convolutions for image classification and object localization[C]// Fifteenth Iapr International Conference on Machine Vision Applications. IEEE, 2017:452-455.

[20] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.

[21] Wang P, Chen P, Yuan Y, et al. Understanding Convolution for Semantic Segmentation[J]. 2017.

[22] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.

[23] Jing Huitian, Han Li, Gao Zhiyu. Wind power ramp prediction based on convolutional neural network feature extraction [J/OL]. Power system automation: 1-13 [2020-07-30]