Combining the Traditional Methods and Deep Learning Approach for Precipitation Nowcasting

Zhang Rui
Chinese Academy of Sciences
P.O. Box 8718
China
501798619@qq.com

Qiao Fengchun
Chinese Academy of Sciences
P.O. Box 8718
China
2435089303@qq.com

Guo Ran
Chinese Academy of Sciences
P.O. Box 8718
China
leslieamber@163.com

ABSTRACT

In this competition, we make use of radar data and rain gauge data to predict short-term rainfall amount. Firstly, we process raw data by “percentile” method and “wind” method. Secondly, we clean the processed data by the similarity of train data and test data. Thirdly, we use the ensemble model which is consisted of Random Forestry, XGBoost and Bidirectional Gated Recurrent Units(GRUs) to predict short-term rainfall amount. In the first phrase, we got the third place. In the second phrase, we got the fourth place.

KEYWORDS

Rainfall amount prediction, Machine Learning, Deep Learning

1 INTRODUCTION

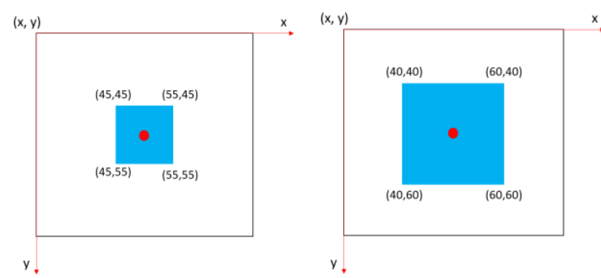
Nowcasting rainfall prediction has long been an important problem in the field of weather forecasting. An accurate rainfall prediction service can support casual usages such as outdoor activity and even provide early warnings of floods or traffic accidents. The goal of this challenge is to give precise rainfall amount on the ground between future 1-hour and 2-hour for each target site. The data provided by the contest organizer is a set of radar maps at different time spans where each radar map covers radar reflectivity of a target site and its surrounding areas.

From the perspective of techniques, traditional machine learning methods such as Random Forestry, gradient boosting methods have been widely used in a number of competitions and achieve satisfying performance. Recent advances in deep learning, especially recurrent neural networks (RNNs) and long short-term memory (LSTM) models have been demonstrated their effectiveness in the area of sequential data modeling [1] [2]. In this challenge, we combine Random Forestry, XGBoost and Bidirectional Gated Recurrent Units (GRUs) into an ensemble model to tackle this problem and achieve satisfying result.

2 The body of the paper

2.1 data process

2.1.1 Percentile. A statistical method was applied to reduce the dimension of radar data. For a single radar map, we pick the 25th, 50th, 75th, 100th percentile of reflectivity values in various scales of neighborhood around the target site from center to the whole map. The scale gradually expands with the step size of 10km. As result, we get 4*10 features from a single radar map. The detailed process of feature selection is illustrated as Fig1. This method enables us to view a single radar map in various scales of receptive fields.



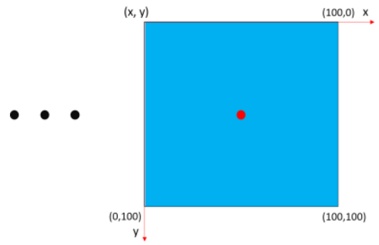


Figure 1: Various scales of neighborhood around the target site.

2.1.2 wind. We first handle the original data (15*4*101*101) into a small size of data (15*4*10*10). Then shrink the data into 15*4*6*6 features through judging the wind direction. The entire preprocess learns from the idea of CNN, especially the convolutional calculation and max polling.

Convolution

For every layer of each time span, average each 5*5 frame from 100*100 (up to 100 columns and rows) original data, resulting in data of 15*4*20*20.

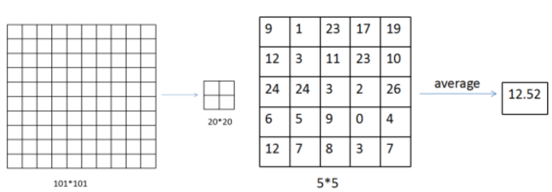


Figure 2: represention of convolutional calculation.

Max pooling

We quarter the 20*20 frame into a 10*10 data, remaining the max one of each 2*2 unit.

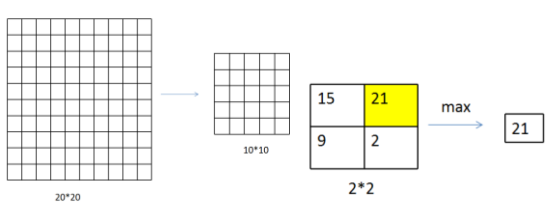


Figure 3: represention of pooling calculation.

The wind direction

Representative Data Selection:

We take the fourth layer of data to determine the wind direction. Then, in order to calculating the resulting wind direction, we carry out two ways of choosing representative data. The first one uses the maximum value in each 10*10 frame as the representation. The second one takes the average of the largest five data instead. After selecting the representative data, we determine the wind direction by calculating the deviation between the initial position and the following frames, voting the moving direction, finally get the maximum votes as the resulting wind direction based on the given thresholds.

Method One

For the 10*10 frame of time span 0-14, we get the 15 representative max points (xx[i], yy[i]), i=0, 1, 2...14, and average first three xx and yy as the average initial position. Then adjust four direction values through comparing the rest 12 points with the initial position.

Method Two

Instead of selecting the max points of each frame, we average the largest five points as representation. For more specific, for the 10*10 frame of time span 0-14, we average them as (xx[i] yy[i]), i=0..14. Then work out the deviation between the time 1-14 with xx[0] and yy[0]. Noticeably, in this method, we choose only the first point as the initial position, not the first three due to the value of points are already the average ones. The voting process is similar with Method One.

Wind Direction Determination:

To further specify the four direction, we regard the direction which has max value as the main direction, then observe the values in crossed directions. If both are lower than threshold, then take the main direction as resulting direction. Otherwise, add the direction whose value is larger than threshold as auxiliary direction. As result, every main direction with its two auxiliary directions results in three specific directions for four directions. Thus, we have twelve wind directions, west, west-north, west-south, east, east-north, east-south, north, north-west, north-east, south, south-west, south-east.

Feature Extraction

In intuitive thinking, winds blow the rainy clouds to their directions, leading to the rainfall wherever they go, which means if we use the radar data (which is directly related to the rainfall) of wind source direction in the past 15 time spans, then prediction will be more precise. Here, we hence extract 6*6 matrices as features from 10*10 matrices.

For twelve wind directions, we have different ways to extract respectively. [2:8,0:6] from 10*10 matrix for the west wind, [1:7,0:6] for the west-north, [3:9,0:6] for the west-south, [2:8,4:10] for the east wind, [1:7,4:10] for the east-north, [3:9,4:10] for the east south, [0:6,2:8] for the north wind, [0:6,1:7] for the north-west, [0:6,3:9] for the north-east, [4:10,2:8] for the south wind, [4:10,1:7] for the south-west, and [4:10,3:9] for the south-east.

Since we have 15 time spans and 4 layers, we get 15*4*6*6 features. Fig. 4 and Fig. 5 displays the way of extracting features when wind direction is west、west-north 、west-south、 north、north-west and north-east .

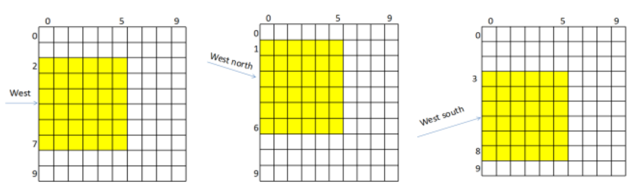


Figure 4: the way of extracting features when wind direction is west、west-north or west-south .

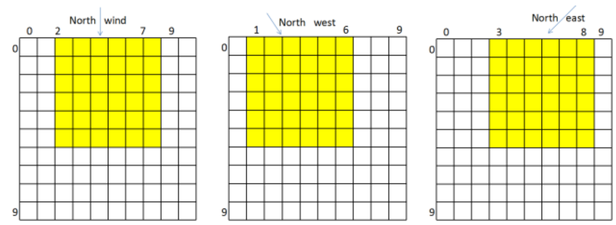


Figure 5: the way of extracting features when wind direction is north、north-west or north-east .

2.2 XGBoost model

We predict the rainfall using the XGBoost model which is a highly efficient, flexible and portable distributed gradient boosting library. We first extract features using the method mentioned in 2.1. Then we flip ten thousand train data to eighty thousand. At last, we use five fold cross validation to train model and predict the result. In addition, We trained two models using two sets of parameters, and weighted average the two results as the ultimate result of XGBoost model.

2.3 Random Forestry Model

The train set of Random Forestry model is the cleaned data, which some train samples are filtered out because of the low similarity of the train samples and test set. And train samples are consisted of "percentile" data and "wind" data, which we focus on the data of the second heights and last 12 time slots. As for Random Forestry model, We only adjust the n_estimators parameter. When n_estimators is set to 1100, this model is best.

2.4 Bi-GRUs model

Recurrent neural networks have been widely used in the area of sequential data modeling. Compared with long short-term memory (LSTM) which are capable of learning long-term dependencies, Gated Recurrent Units (GRUs) have fewer parameters and are more suitable for limited data [3]. The detailed architecture of GRUs are represented as (1) (2) (3).

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{3}$$

Where x_t denotes input vector, h_t denotes output vector, z_t denotes output gate vector and r_t denotes reset gate vector. W_* , U_* are matrices and b_* are vectors. σ_* denote activation functions.

The basic idea of bidirectional RNNs is to combine a forward RNN layer and a backward RNN layer into a shared output layer. This kind of architecture encodes contextual information of input data and takes advantage of both the past and future data. The architecture of our model is a two-layer stacked bidirectional GRUs, and a dense layer is added between the layers. Average pooling is performed at the output of the model and mean squared error is utilized as the loss function. There are 128 cells utilized in the GRU layers, 32 nodes in the first dense layer and 1 node in the last dense layer. The maximum training epoch is 100. Dropout regularization technique and batch normalization are performed in all layers except the output layer. Dropout rate is 0.5.

3 Conclusion

In summary, we conduct a series of experiments on data processing and model combination. As a result, convolution-based, wind-based and percentile based methods are selected for data processing. Random Forestry, XGBoost and Bidirectional GRUs are utilized for model ensemble. Our best result on the first stage is RMSE 12.99671 and we get the 3rd place. On the second stage is RMSE 13.20062 and we get the 4th place in this challenge among 1395 teams.

A HEADINGS IN APPENDICES

A.1 Introduction

A.2 The body of the paper

A.2.1 data process

percentile

wind

A.2.2 Xgboost model

A.2.3 Random Forestry Model

A.2.4 Bi-GRUs model

A.3 Conclusions

A.4 References

ACKNOWLEDGMENTS

This work was partially supported by the Tianchi CIKM AnalytiCup 2017 big data contest and the Alibaba company and Shenzhen meteorologic bureau .

REFERENCES

[1] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Advances in neural information processing systems. 2014: 3104-3112.

[2] Donahue J, Anne Hendricks L, Guadarrama S, et al. Long-term recurrent convolutional networks for visual recognition and description[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 2625-2634.

[3] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.