MCGRU: Multi-Channel Gated Recurrent Unit for Weather Precipitation

Dan Niu, Haorui Li, Hangyuan Yu

Abstract—With the goal of making high-resolution forecasts of regional rainfall, precipitation nowcasting has become an important and fundamental technology underlying various public services ranging from rainstorm warnings to flight safety. Recently, the Trajectory GRU (TrajGRU) model has been shown to outperform traditional optical flow based methods for precipitation nowcasting, suggesting that deep learning models have a huge potential for solving the problem. However, the TrajGRU models is only based on image sequence, ignoring the effect of multi-modeling data from weather area . Furthermore, TrajGRU has achieved SOTA performance in weather forcast area, we proposed a new model whose performace is no less than TrajGRU. Specifically, we go beyond it and propose the Multi-Channel Gated Recurrent Unit (MCGRU) model that can actively learn the location-variant structure from multi-modeling data.

Index Terms—Precipitation Nowcasting, Multimodal Deep Learning, GRU.

1 Introduction

Precipitation nowcasting refers to the problem of providing very short range (e.g., 0-6 hours) forecast of the rainfall intensity in a local region based on radar echo maps1 , rain gauge and other observation data as well as the Numerical Weather Prediction (NWP) models. It significantly impacts the daily lives of many and plays a vital role in many real-world applications. Among other possibilities, it helps to facilitate drivers by predicting road conditions, enhances flight safety by providing weather guidance for regional aviation, and avoids casualties by issuing citywide rainfall alerts. In addition to the inherent complexities of the atmosphere and relevant dynamical processes, the ever-growing need for real-time, large-scale, and fine-grained precipitation nowcasting poses extrachallenges to the meteorological community and has aroused research interest in the machine learning community [1].

The conventional approaches to precipitation nowcasting used by existing operational systems rely on optical flow [2]. In optical flow method, the movements of convective cloud is calculated using the observed radar echo maps and then the future radar echo maps are predicted by virtue of the Lagrangian advection method. However, this method is simply based on the differentiation of some observed pictures, which is a waste of plenty of historical data. Despite much calculation is done using this method, the result is not stable and not always accurate. Moreover, these methods are unsupervised from the machine learning point of view in that they do not take advantage of the vast amount of existing radar echo data.

Recently, progress has been made by utilizing supervised deep learning [3] techniques for precipitation nowcasting. Shi et al. [1] formulated precipitation nowcasting as a spatiotemporal sequence forecasting problem and proposed the Convolutional Long Short-Term Memory (ConvLSTM) model, which extends the LSTM [4] by having convolutional structures in both the input-to-state and state-to-state transitions, to solve the problem. Using the radar echo sequences for model training, the authors showed that ConvLSTM is better at capturing the spatiotemporal correlations than the fully-connected LSTM and gives more accurate predictions than the Real-time Optical flow by Variational methods for Echoes of Radar (ROVER) algorithm [2] currently used by the Hong Kong Observatory.

2 Related Work

For the precipitation nowcasting problem, the reflectivity factors in radar echo maps are first transformed to grayscale images before being fed into the prediction algorithm [1]. Thus, precipitation nowcasting can be viewed as a type of video prediction problem with a fixed "camera", which is the weather radar. Therefore, methods proposed for predicting future frames in natural videos are also applicable to precipitation nowcasting and are related to our paper.

There are three types of general architecture for video prediction: RNN based models, 2D CNN based models, and 3D CNN based models. Ranzato et al. [5] proposed the first RNN based model for video prediction, which uses a convolutional RNN with 1×1 state-state kernel to encode the observed frames. Srivastava et al. [6] proposed the LSTM encoder-decoder network which uses one LSTM to encode the input frames and another LSTM to predict multiple frames ahead. The model was generalized in [1] by replacing the fully-connected LSTM with ConvLSTM to capture the spatiotemporal correlations better. Later, Finn et al. [7] and De Brabandere et al. [3] extended the model in [1] by making the network predict the transformation of the input frame instead of directly predicting the raw pixels. Ruben et al. [8] proposed to use both an RNN that captures the motion and a CNN that captures the content to generate the prediction. Along with RNN based models, 2D and 3D CNN based models were proposed in [9] and [10] respectively. Mathieu et al. [9] treated the frame sequence as multiple channels and applied 2D CNN to generate the prediction while [10] treated them as the depth and applied 3D CNN. Both papers show that Generative Adversarial Network (GAN) [11] is helpful for generating sharp predictions.

Our work is based on Trajectory GRU model. That paper have provided the first large-scale benchmark for precipitation nowcasting and have proposed a new TrajGRU model with the ability of learning the recurrent connection structure. That

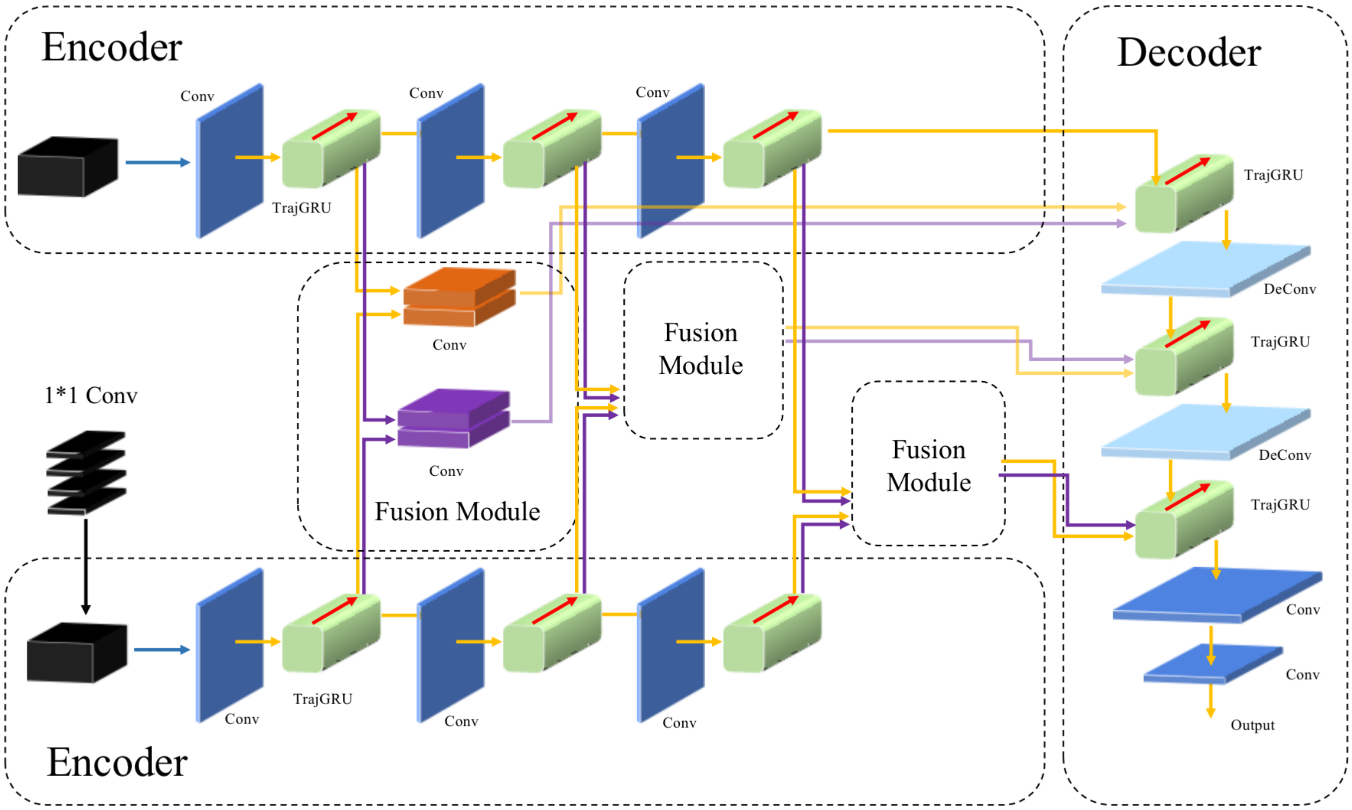

Fig. 1. The network chart of MCGRU

paper have shown TrajGRU is more efficient in capturing the spatiotemporal correlations than ConvGRU.

3 Approach

The radar echo map as well as the atmospheric data is preprocessed before being fed into our network. The Multi-Channel Gated Recurrent Unit(MCGRU) is based on multi-modal data and up-and-down sampling gated recurrent unit(GRU). MCGRU is not only conducive to model training and improving the accuracy of short-term precipitation prediction, especially the prediction accuracy of heavy rain, but also can solve the precipitation existing in the prior art Technical problems such as unbalanced data, low rainstorm prediction accuracy, and fewer model fusion features.

3.1 Preprocessing

We use bilinear interpolation to fill the missing data in the radar echo graph

$$
\begin{aligned}
f(x,y) \approx & \frac{f(Q_{11})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y)+ \\
& \frac{f(Q_{21})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y)+ \\
& \frac{f(Q_{12})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1)+ \\
& \frac{f(Q_{22})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1)
\end{aligned}
$$

Afterwards, we use hard threshold wavelet transform to denoise atmospheric grid data.

3.2 Convolution

The count of channels is adjusted by applying 1×1 kernel:

$$y = \sum_c w(c)x(c)$$

Then we adjust the space size of the new feature map through a deconvolution operation, so that the scale of the feature map is consistent with the scale of the radar echo image

$$O = K * I$$

where $*$ is the convolution operation of matrices, I is the matrix obtained after the channels are modified and K is the deconvolution kernel.

3.3 MCGRU

In MCGRU, different sources of data are fed into separate GRUs, and finally they are concatenated to jointly work for prediction. The radar echo map data and grid air data are input into an encoder network containing two down-sampling loop gated units for encoding, and the two obtained feature maps are spliced. The encoder network is composed of convolutional layers, cyclic gating unit layer, down-sampling layer, cyclic gating unit layer, down-sampling layer, cyclic gating unit layer. The encoding of radar echo maps and the encoding of atmospheric grid data are concatenated in terms of feature channels.

3.4 Decoding and Precipitation

The concatenated data are fed into a decoder network containing two-layer up-sampling GRU for decoding to obtain a predicted radar echo image. By using ZR transformation, the precipitation prediction is available in terms of area. The decoder network is composed of a loop gating unit, cyclic gating unit, deconvolution up-sampling, cyclic gating unit, and deconvolution up-sampling.

4 Models

In the work of Ranzato et al. [5], RNN was first introduced to modeling image sequences. In RNNs, we have as many hidden states as inputs and each hidden state is computed based on the previous hidden state and the input on that step. However, it is difficult for us to access information from many steps back and vanishing gradients turn out to be a severe problem in vanilla RNNs. This problem can be solved by adding separate memory to RNNs, which is the motivation for GRU. GRU adds update gate and reset gate to the original RNN cell. In our task, when used for capturing spatiotemporal correlations, the deficiency of ConvGRU and other ConvRNNs is that the connection structure and weights are fixed for all the locations. As the hyperparameter of convolution is fixed, the neighborhood set stays the same for all locations, which is a bad phrnomenon in our precipitation nowcasting task. Hence, the trajGRU model was proposed to replace ConvGRU.

4.1 Trajectory GRU

We know that most motion patterns have different neighborhood sets for different locations. For example, rotation and scaling generate flow fields with different angles pointing to different directions. Based on that, TrajGRU uses the current input and previous state to generate the local neighborhood set for each location at each timestamp. Since the location indices are discrete and non-differentiable, we use a set of continuous optical flows to represent these "indices". The main formulas of TrajGRU are given as follows:

$$
\begin{aligned}
&\mathcal{U}_t, \mathcal{V}_t = \gamma(\mathcal{X}_t, \mathcal{H}_{t-1}) \\
&\quad \mathcal{Z}_t = \sigma\left(\mathcal{W}_{xz} * \mathcal{X}_t + \sum_{l=1}^{L} \mathcal{W}_{hz}^{l} * \mathrm{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})\right) \\
&\quad \mathcal{R}_t = \sigma\left(\mathcal{W}_{xr} * \mathcal{X}_t + \sum_{l=1}^{L} \mathcal{W}_{hr}^{l} * \mathrm{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})\right) \\
&\quad \mathcal{H}_t' = f\left(\mathcal{W}_{xh} * \mathcal{X}_t + \mathcal{R}_t \circ \left(\sum_{l=1}^{L} \mathcal{W}_{hr}^{l} * \mathrm{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})\right)\right) \\
&\quad \mathcal{H}_t = (1 - \mathcal{Z}_t) \circ H_t' + \mathcal{Z}_t \circ H_{t-1}
\end{aligned}
$$

Here, L is the total number of allowed links. $\mathcal{U}_t, \mathcal{V}_i \in \mathbf{R}^{L \times H \times W}$ are the flow fields that store the local connection structure generated by the structure generating network γ. The $\mathcal{W}_{hz}^{l}, \mathcal{W}_{hr}^{l}, \mathcal{W}_{hh}^{l}$ are the weights for projecting the channels, which are implemented by 1×1 convolutions. The $\mathrm{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})$ function selects the positions pointed out by $\mathcal{U}_{t,l}, \mathcal{V}_{t,l}$ from $\mathcal{H}_{t-1}$ via the bilinear sampling kernel.

4.2 Multi-Channel GRU

We propose multi-channel GRU because in actual world, numerous meteorological parameters are available in addition to the radar echo maps(M). We have Simulated Micro-Physical Parameters(G), Simulated Radar Reflectivity(R), Simulated Maximum Vertical Velocity(V) and so on. Each parameter is represented in a graph the same size as the radar echo map. In the Multi-Channel GRU model architecture, we have two encoders for two source of data input.

4.2.1 MAP encoder

The MAP encoder is responsible for encoding the weather forecasting simulation data$(M, G, R$ and $V)$. We reduce the spatial resolution of M through a trajGRU

$$\hat{M}_t = trajGRU(M_t)$$

For other parameters, we calculate the average level of the parameters along the z direction(altitude) and then stack all these data together to get the final input $P = P[t]_{t=0}^{h}$, where P_t is defined as:

$$P_t = \mathrm{stack}^{z\_avg}(\hat{M}_t, G_t, R_t, V_t)$$

Afterwards, P is sent into an LSTM network to obtain two feature tensors

$$[C_P, H_P] = LSTM(P)$$

where C is the sequence of all cell states while H is the sequence of all hidden states.

4.2.2 OBSERVATION encoder

The OBS encoder is responsible for encoding the observed raining intensity data I. To make it consistent with the MAP encoder, we also make it pass an LSTM network to obtain two feature tensors

$$[C_I, H_I] = LSTM(I)$$

5 Experiment

This method verifies that the data set is the radar echo map, gridded temperature and total precipitation provided by Guangdong Meteorological Bureau.The time span is from March 2017 to December 2018.The resolution is 1 km and the matrix size is 300*300*1.The data interval was 12 minutes.The z-r relation represents the relation between reflectivity Z and precipitation intensity R (mm/h), where, a and b are the parameters of the radar itself, and are values in this experiment.DBZ is commonly used to describe the precipitation. In general, the greater the value, the greater the reaction precipitation. Grid meteorological data (including 9 humidity levels at elevations, 12 wind speeds at elevations, and 2 rainfall intensities, in 13 characteristic channels) is provided for GRAVETS in South China with a matrix size of 100*100*13 with a resolution of 3 km and 1 hour. Combined with the experimental experience, the radar echo of the first 10 moments and the total precipitation of gridded temperature were used in this experiment to predict the radar echo of the last 10 moments.

We then calculate the TP (prediction=1, truth=1), FN (prediction=0, truth=1), FP (prediction=1, truth=0), and TN (prediction=0, truth=0). The evaluation index of precipitation prediction in meteorological field is CSI score is:

$$CSI = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

In this method, the historical radar echo map at time t and the gridded meteorological data (including 9 kinds of humidity at altitude, 12 kinds of wind speed at altitude and 2 kinds of rainfall intensity, a total of 13 characteristic channels) were input, and the missing values were supplemented and denoised.Then, to solve the problem of different spatial and temporal resolutions, the 1*1 convolution kernel is set as 1*1*1, and the parameters of the deconvolution kernel are set as:

Name	Kernel	Stride	Pad	Ch I/O	In Res	Out Res
econv1	7*7	5*5	1*1	1/4	300*300	60*60
eRNN1	3*3	1*1	1*1	4/8	60*60	60*60
edown1	5*5	3*3	1*1	8/8	60*60	20*20
eRNN2	3*3	1*1	1*1	8/64	20*20	20*20
edown2	3*3	2*2	1*1	64/64	20*20	10*10
eRNN3	3*3	1*1	1*1	64/64	10*10	10*10
fRNN1	3*3	1*1	1*1	128/128	10*10	20*20
fup2	4*4	2*2	1*1	128/128	20*20	20*20
fRNN2	3*3	1*1	1*1	128/128	20*20	60*60
fup3	5*5	3*3	1*1	128/128	60*60	60*60
fRNN3	3*3	1*1	1*1	128/64	60*60	300*300
fconv2	7*7	5*5	1*1	64/8	300*300	300*300
fconv3	1*1	1*1	0*0	8/1	300*300	300*300

TABLE 1

The details of the MCGRU model. The ‘In Kernel‘, ‘In Stride‘ and ‘In Pad‘ are the kernel, stride and padding in the input-to-state convolution.

Rain Rate (mm/h)	Rainfall Level
$r \geq 0.5$	No / Hardly noticeable
$r \geq 2$	Light
$r \geq 5$	Light to moderate
$r \geq 10$	Moderate to heavy
$r \geq 30$	Rainstrom

TABLE 2

R value and corresponding rain rate

$$s = 3$$
$$k = 3$$
$$p = 0$$

The grid meteorological data can be mapped to the same scale as radar echo data.Then, the two feature images are input into the GRU network containing two layers of down-sampling respectively for coding, and the two feature images obtained are stitched together to obtain the fusion feature image.Finally, the fusion feature image is input into the GRU network containing two-layer upper sampling for decoding, and the predicted radar echo image is obtained. The future regional precipitation prediction is obtained through Z-R transformation, and the prediction results of short-imminent precipitation are output.

Table 1 is the model parameter table of the GRU network with up and down sampling.Combined with the experimental experience, the radar echo of the first 10 moments and gridded meteorological data were used in this experiment to predict the radar echo of the last 10 moments.

R value is the rainfall intensity grade obtained after Z-RC transformation, and the corresponding relationship with precipitation broadcast is shown in Table 2

The traditional methods usually adopt 2D CNN or 3D CNN. Table 3 below shows the average scores predicted by several models on the CSI index of the experimental data at different levels. 2D-CNN is a model that uses 2D convolutional neural network to carry out time sequence images.3D CNN is a time series image model that uses 3D-CNN to represent 3D convolutional neural network. The higher the CSI index, the higher the accuracy of precipitation prediction.As can be seen from the table below, this method has an improvement in predicting scores compared with traditional methods.

Model	$r \geq 0.5$	$r \geq 2$	$r \geq 5$	$r \geq 10$	$r \geq 30$
BaseLine	0.548	0.466	0.352	0.214	0.071
MCGRU	0.549	0.473	0.372	0.279	0.177

TABLE 3

CSI results of different methods

6 Discussion and Conclusion

This paper proposes a novel model called MCGRU, which has great advantages in now-casting CSI value. It can improve the accuracy of multi-scale precipitation prediction and integrate the image features of various meteorological data.

Future work should mainly focus on improving long-term precipitation. We know that the prediction of radar echo maps is similar to a seq2seq problem. Hence, we may apply attention mechanism or the transformer architecture to our algorithms to perform video tasks better. It remains to be seen whether the multi-head convolutional self-attention could efficiently exploit the long-range dependence within a video-frame sequence. What's more, in order to generate pictures closer to actual situations, the Generative Adversarial Network can be applied to make our model better to interpret. We can use our original architecture as a generator and use RNN to train another discriminator to tell whether the pictures we generated are reasonable.

The proposed MCGRU architecture is concise, compact and efficient. Extensive quantitative and qualitative evaluations indicate that the proposed solution performs favorably against existing frame extrapolation and interpolation methods. Our successful implementation may shed light for some other video tasks.

References

[1] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” 2015, cite arxiv:1506.04214. [Online]. Available: http://arxiv.org/abs/1506.04214

[2] R. Krishnamurthy, P. Moulin, and J. W. Woods, “Optical flow techniques applied to video coding.” in *ICIP*. IEEE Computer Society, 1995, pp. 570–573. [Online]. Available: http://dblp.uni-trier.de/db/conf/icip/icip1995-1.htmlKrishnamurthyMW95

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, 2017. [Online]. Available: https://www.worldcat.org/title/deep-learning/oclc/985397543referer=brief$_results$

[4] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, “Video (language) modeling: a baseline for generative models of natural videos.” *CoRR*, vol. abs/1412.6604, 2014. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1412.htmlRanzatoSBMCC14

[6] N. Srivastava, E. Mansimov, and R. Salakhutdinov, “Unsupervised learning of video representations using lstms,” 2015.

[7] C. Finn, I. J. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction." in *NIPS*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 64–72. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2016.htmlFinnGL16

[8] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction." in *ICLR (Poster)*. OpenReview.net, 2017. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2017.htmlVillegasYHLL17

[9] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error." in *ICLR (Poster)*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://dblp.uni-trier.de/db/conf/iclr/iclr2016.htmlMathieuCL15

[10] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics." in *NIPS*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 613–621. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2016.htmlVondrickPT16

[11] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, cite arxiv:1411.1784. [Online]. Available: http://arxiv.org/abs/1411.1784