Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model

Xingjian Shi[1] Zhihan Gao[1] Leonard Lausen[1] Hao Wang[1]
Dit-Yan Yeung[1] Wai-kin Wong[2] Wang-chun Woo[2]

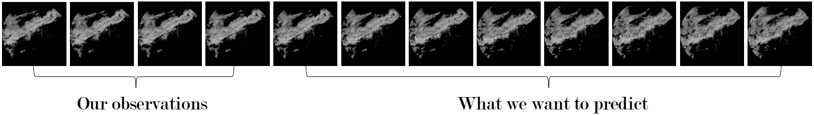
[1]Department of Computer Science and Engineering
Hong Kong University of Science and Technology

[2]Hong Kong Observatory
Hong Kong, China

Neural Information Processing System (NIPS) 2017
Poster: #110

What is precipitation nowcasting? Why is it important?

- Provide nowcast (0-6 hours) of the rainfall intensity in a local region based on radar echo maps, rain gauge and other data.



- Precipitation nowcasting IMPACTS our daily life.

(a) Predict road condition

(b) Weather guidance for aviation

(c) Urban rainstorm warnings

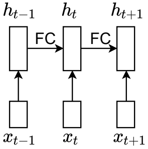
- Complexities of the atmosphere + real-time, large-scale, and fine-grained nowcasting → New challenges!

Previous Work

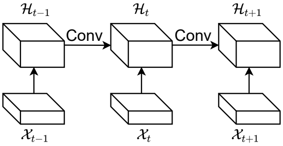
- Formulated as a spatiotemporal sequence forecasting problem and modeled by an encoding-forecasting structure [Shi *et al.*, 2015]:

$$\tilde{X}_{t+1}, \ldots, \tilde{X}_{t+K} = \operatorname{argmax}_{\mathcal{X}_{t+1},\ldots,\mathcal{X}_{t+K}} p(\mathcal{X}_{t+1}, \ldots, \mathcal{X}_{t+K} \mid \hat{\mathcal{X}}_{t-J+1}, \ldots, \hat{\mathcal{X}}_{t})$$
$$\approx g_{\text{encoder}}(f_{\text{forecaster}}(\hat{\mathcal{X}}_{t-J+1}, \ldots, \hat{\mathcal{X}}_{t})).$$

- Use *Convolutional LSTM* (ConvLSTM) in the encoder and forecaster. Unlike FC-LSTM, ConvLSTM uses convolution in the state-state connections and is better at capturing spatiotemporal correlations.



(a) States are fully-connected in FC-LSTM.

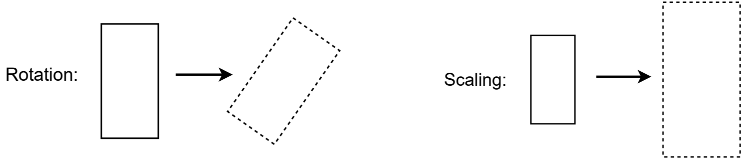


(b) State-state connection is convolutional in ConvLSTM.

Motivation

The previous work has two deficiencies

- Convolutional recurrence structure in ConvLSTM is *location-invariant*, which is not good at modeling *location-variant* spatiotemporal patterns.



- The evaluation method was far from real-world requirement. “Deep Learning for Precipitation Nowcasting” is still in its early stage!

Our solutions:

- *Trajectory GRU* (TrajGRU) that can actively learn the location-variant structure.
- The HKO-7 benchmark with new performance measures and a clear evaluation protocol.

ConvGRU

- ConvGRU: Replace full-connection with convolution in GRU.

$$\begin{aligned}\mathcal{Z}_t &= \sigma(\mathcal{W}_{xz} * \mathcal{X}_t + \mathcal{W}_{hz} * \mathcal{H}_{t-1}),\\ \mathcal{R}_t &= \sigma(\mathcal{W}_{xr} * \mathcal{X}_t + \mathcal{W}_{hr} * \mathcal{H}_{t-1}),\\ \mathcal{H}'_t &= f(\mathcal{W}_{xh} * \mathcal{X}_t + \mathcal{R}_t \circ (\mathcal{W}_{hh} * \mathcal{H}_{t-1})),\\ \mathcal{H}_t &= (1 - \mathcal{Z}_t) \circ \mathcal{H}'_t + \mathcal{Z}_t \circ \mathcal{H}_{t-1}.\end{aligned} \tag{1}$$

- Convolutional recurrence fixes the neighborhood set and is location-invariant

$$\mathcal{H}'_{t,:,i,j} = f(\mathbf{W}_{hh}\text{concat}(\langle \mathcal{H}_{t-1,:,p,q} \mid (p,q) \in \mathcal{N}^h_{i,j} \rangle)) \tag{2}$$

- Ideally, neighborhood sets should vary for different locations and timestamps!

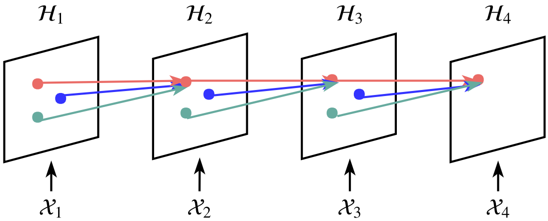
TrajGRU

- In TrajGRU, we use a subnetwork to output these neighborhood sets.

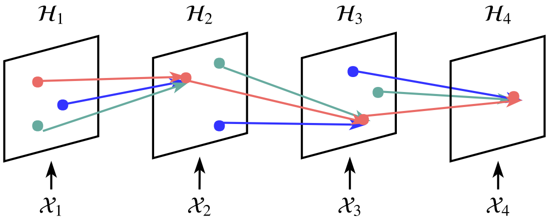
$$
\begin{aligned}
\mathcal{U}_t, \mathcal{V}_t &= \gamma(\mathcal{X}_t, \mathcal{H}_{t-1}), \\
\mathcal{Z}_t &= \sigma(\mathcal{W}_{xz} * \mathcal{X}_t + \sum_{l=1}^{L} \mathcal{W}_{hz}^{l} * \text{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})), \\
\mathcal{R}_t &= \sigma(\mathcal{W}_{xr} * \mathcal{X}_t + \sum_{l=1}^{L} \mathcal{W}_{hr}^{l} * \text{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l})), \\
\mathcal{H}_t' &= f(\mathcal{W}_{xh} * \mathcal{X}_t + \mathcal{R}_t \circ (\sum_{l=1}^{L} \mathcal{W}_{hh}^{l} * \text{warp}(\mathcal{H}_{t-1}, \mathcal{U}_{t,l}, \mathcal{V}_{t,l}))), \\
\mathcal{H}_t &= (1 - \mathcal{Z}_t) \circ \mathcal{H}_t' + \mathcal{Z}_t \circ \mathcal{H}_{t-1}.
\end{aligned} \tag{3}
$$

- Use optical flow to represent the "indices" [Jaderberg *et al.*, 2015].

Illustration of ConvGRU and TrajGRU



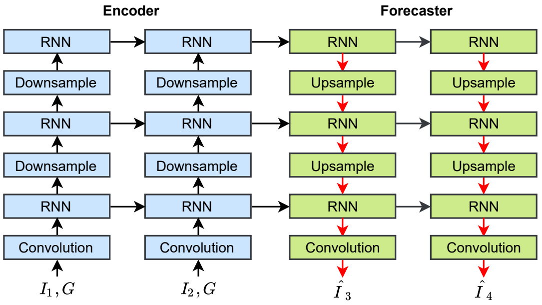
(a) ConvRNN: Links are fixed over time/location.



(b) TrajRNN: Links are dynamically determined.

New Encoding-forecasting Structure

We reverse the link of the forecaster. Both ConvGRU and TrajGRU can be used as the RNN.

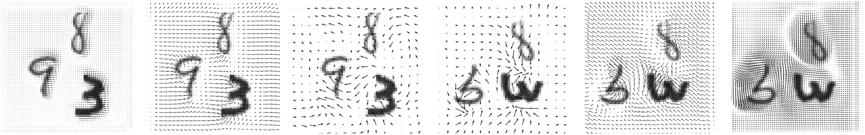


Encoder: local $\rightarrow$ global, Forecaster: global $\rightarrow$ local

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



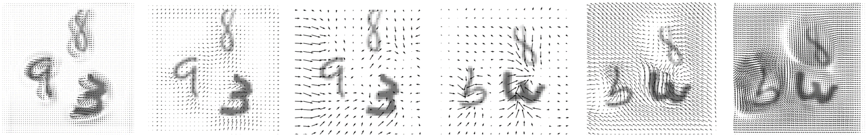
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=0

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



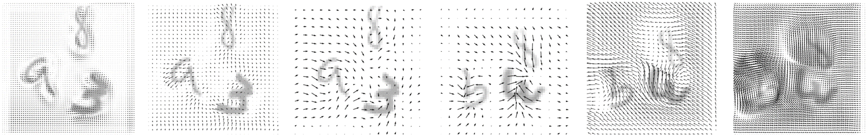
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=1

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern $\rightarrow$ High-level pattern. Forecaster: Coarse $\rightarrow$ Fine



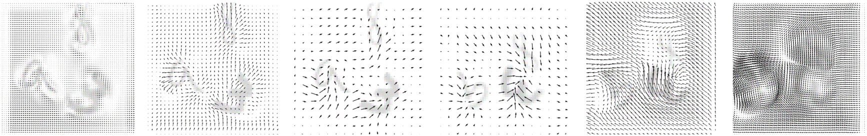
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=2

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



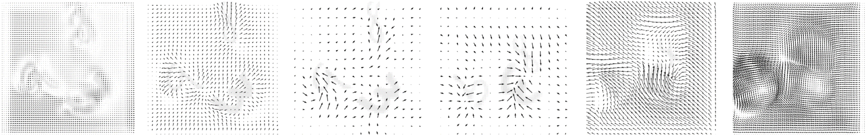
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=3

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



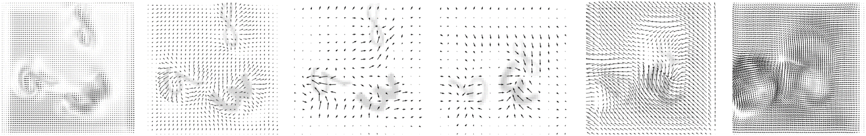
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=4

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



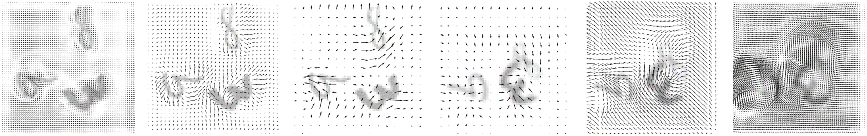
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=5

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



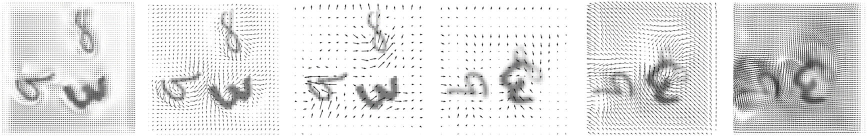
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=6

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



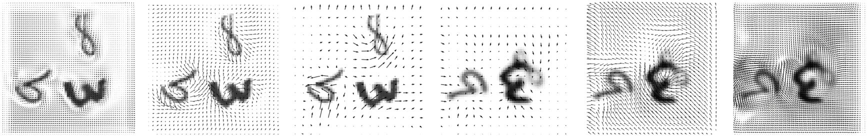
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=7

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



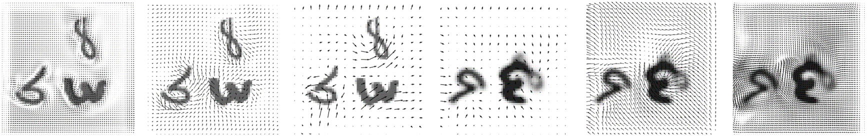
E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=8

Experiments on MovingMNIST++

- MovingMNIST++: MovingMNIST with random rotation, scale changes and illumination changes.

- Results show that TrajGRU performs better than ConvGRU with less parameters.
- Also, the learned recurrent connection structure is meaningful. Encoder: Low-level pattern → High-level pattern. Forecaster: Coarse → Fine



E-L1 E-L2 E-L3 F-L3 F-L2 F-L1

t=9

Highlights of HKO-7 Benchmark

- We use the 7-year radar data from Hong Kong Observatory.
- Heavier rainfall occurs LESS often but has LARGER real-world impact.
  - *Balanced MSE* (B-MSE) and *Balanced MAE* (B-MAE) that assign more weights to heavier rainfall.
- In real application, the nowcasting algorithm can use online learning to adapt to newly emerging patterns.
  - Evaluation protocol has both the offline setting and the online setting.
- Evaluated 3 classical models and 4 deep learning based models (ConvGRU, TrajGRU, 2D CNN and 3D CNN).

Our Findings

- ALL the deep learning models outperform the optical flow based models.
- TrajGRU attains the BEST overall performance among all the deep learning models.
- With online fine-tuning, models CONSISTENTLY perform better.

Come check it out!

Poster #110, Pacific Ballroom, Today

Code: https://github.com/sxjscience/HKO-7