A deep-learning method for precipitation nowcasting

Wai-kin WONG

Xing Jian SHI, Dit Yan YEUNG, Wang-chun WOO

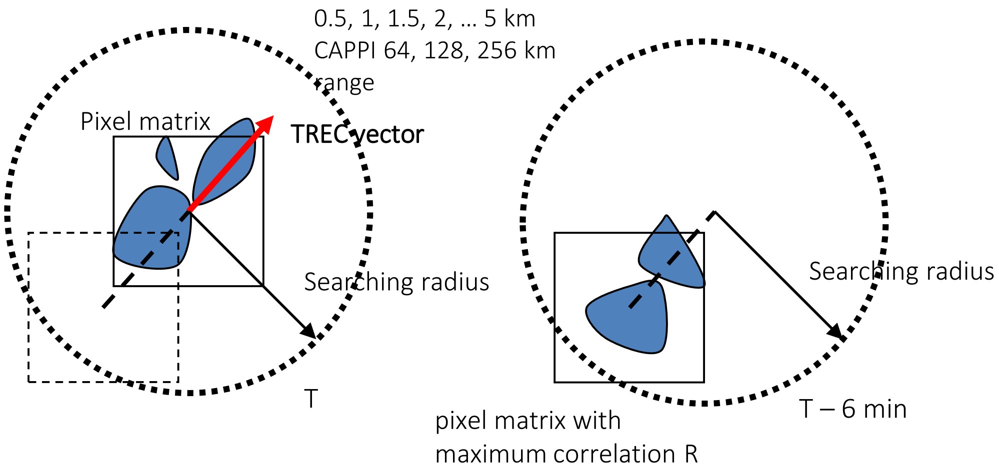
WMO WWRP 4th International Symposium on Nowcasting and Very-short-range Forecast 2016 (WSN16)

Session T2A, 26 July 2016

香港天文台
HONG KONG OBSERVATORY

Echo Tracking in SWIRLS Radar Nowcasting System

- Maximum Correlation (TREC)



where Z_1 and Z_2 are the reflectivity at T+0 and T+6min respectively

$$R = \frac{\sum_k Z_1(k)\times Z_2(k) - \frac{1}{N}\sum_k Z_1(k)\sum_k Z_2(k)}{\left[\left(\sum_k Z_1^2(k) - N\overline{Z_1}^2\right)\times\left(\sum_k Z_2^2(k) - N\overline{Z_2}^2\right)\right]^{1/2}}$$

- Optical Flow

MOVA – Multi-scale Optical-flow by Variational Analysis

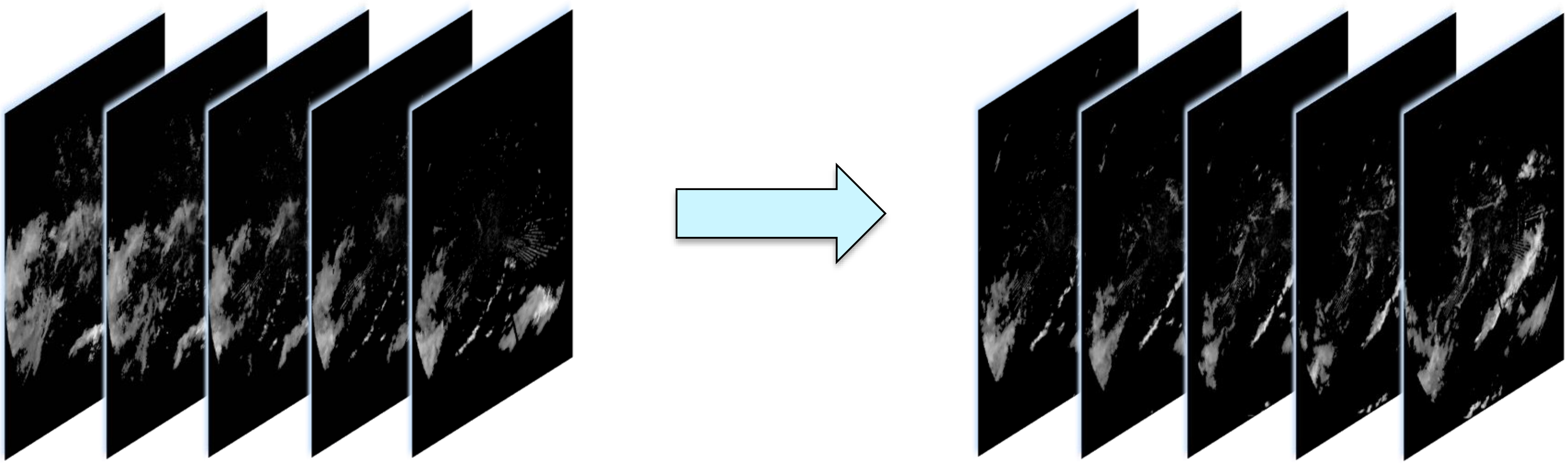
ROVER – Real-time Optical-flow by Variational method for Echoes of Radar

Given I(x,y,t) the image brightness at point (x,y) at time t and the brightness is constant when pattern moves, the echo motion components u(x,y) and v(x,y) can be retrieved via minimization of the cost function:

$$J = \iint\left[\frac{\partial I}{\partial t} + u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y}\right]^2 dxdy$$

Predicting evolution of weather radar maps

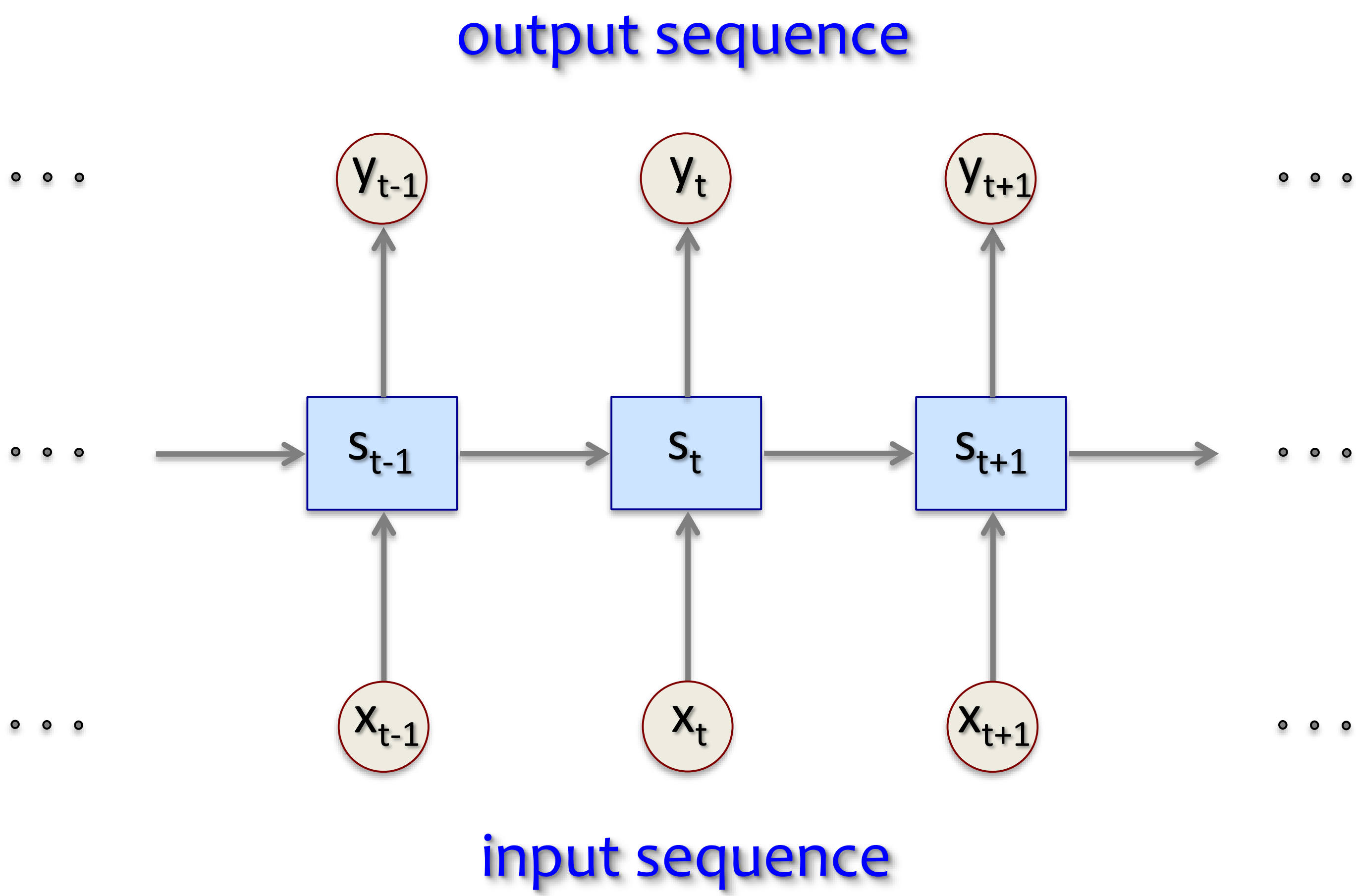
- Input sequence: observed radar maps up to current time step
- Output sequence: predicted radar maps for future time steps



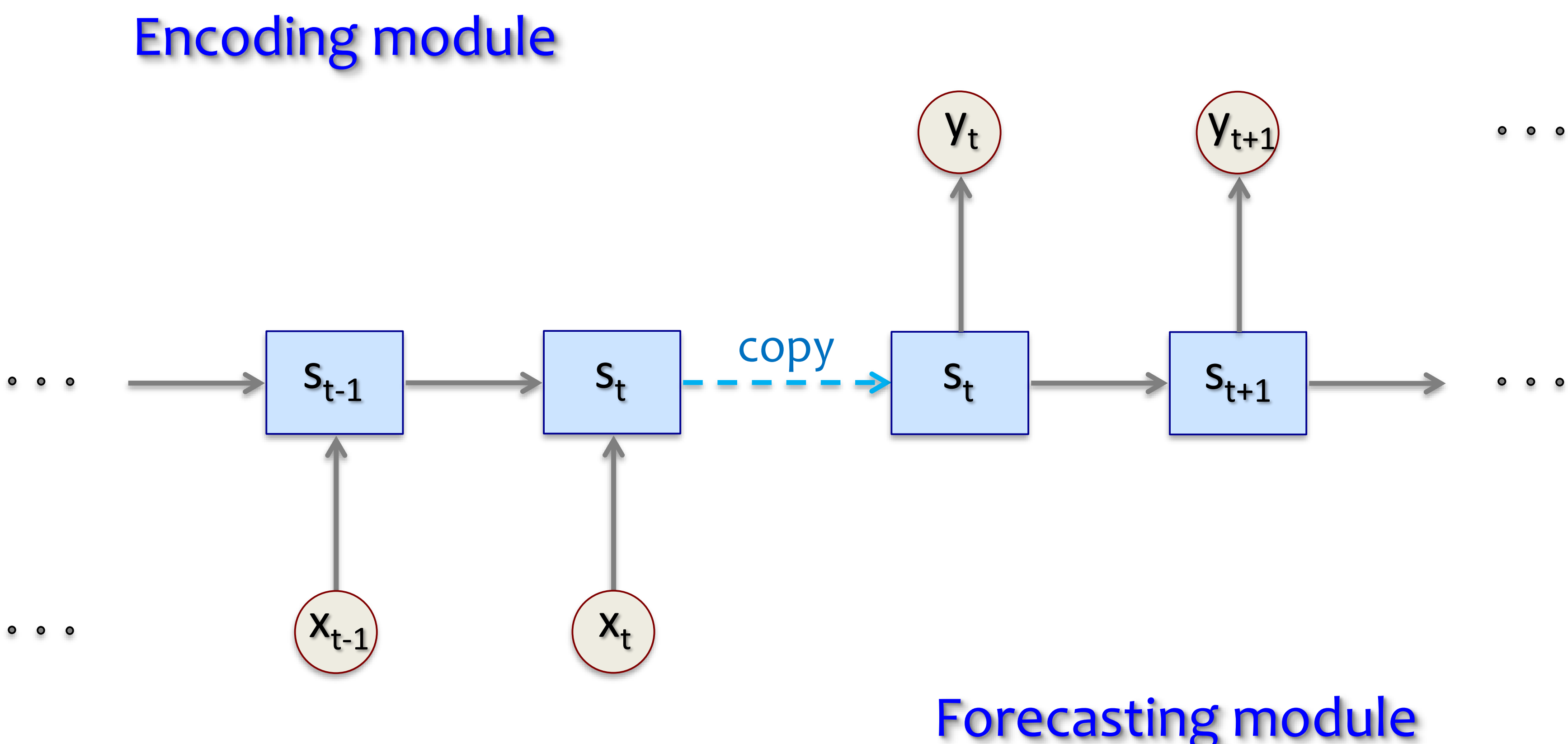
$$\tilde{\mathcal{X}}_{t+1}, \ldots, \tilde{\mathcal{X}}_{t+K} = \underset{\mathcal{X}_{t+1}, \ldots, \mathcal{X}_{t+K}}{\arg\max} \; p(\mathcal{X}_{t+1}, \ldots, \mathcal{X}_{t+K} \mid \hat{\mathcal{X}}_{t-J+1}, \hat{\mathcal{X}}_{t-J+2}, \ldots, \hat{\mathcal{X}}_{t})$$

Maximize posterior pdf of echo sequence across K time levels based on previous J time levels of observations

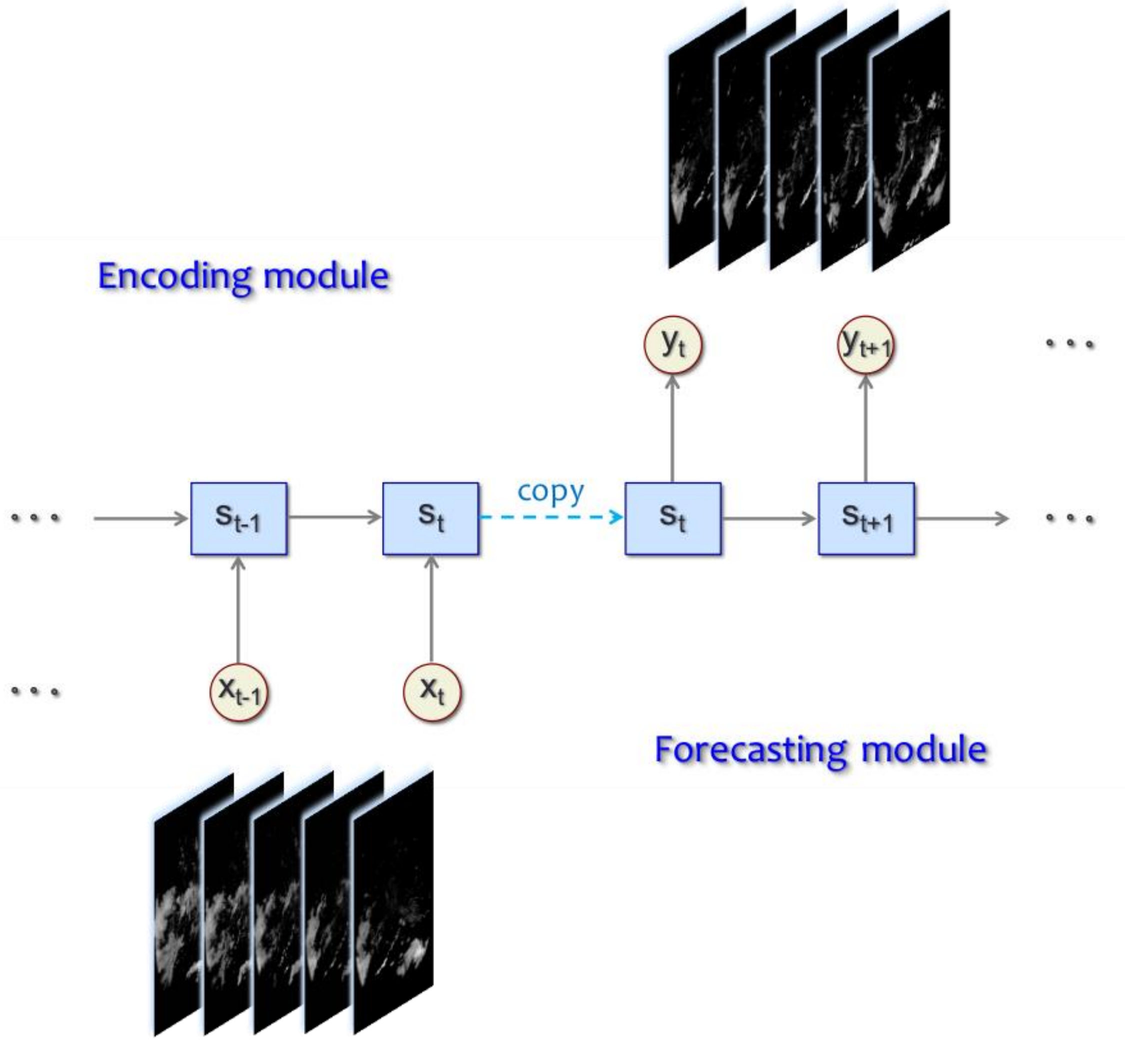
Sequence-to-sequence learning



Encoding-forecasting model



Spatiotemporal encoding-forecasting model



ConvLSTM model

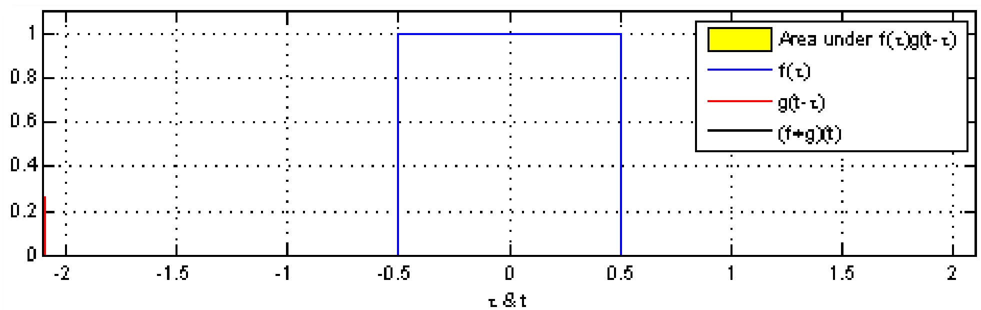
- Convolutional long short-term memory (ConvLSTM) model

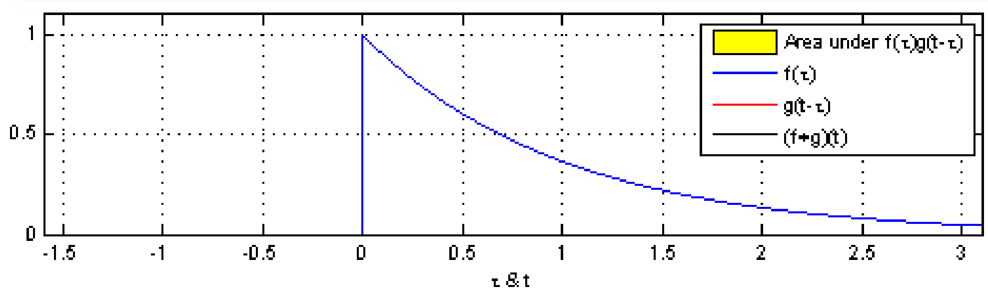
X. Shi, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, and W.C. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *NIPS 2015*.

- Two key components:
  - Convolutional layers
  - Long short-term memory (LSTM) cells in recurrent neural network (RNN) model

Convolution

- An operation on two functions
- Produces a third function which gives the overlapped area of the two functions as a function of the translation of one of the two functions





Convolution

- Continuous domains:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau = \int_{-\infty}^{\infty} f(t-\tau)\, g(\tau)\, d\tau = (g * f)(t)$$

- Discrete domains:

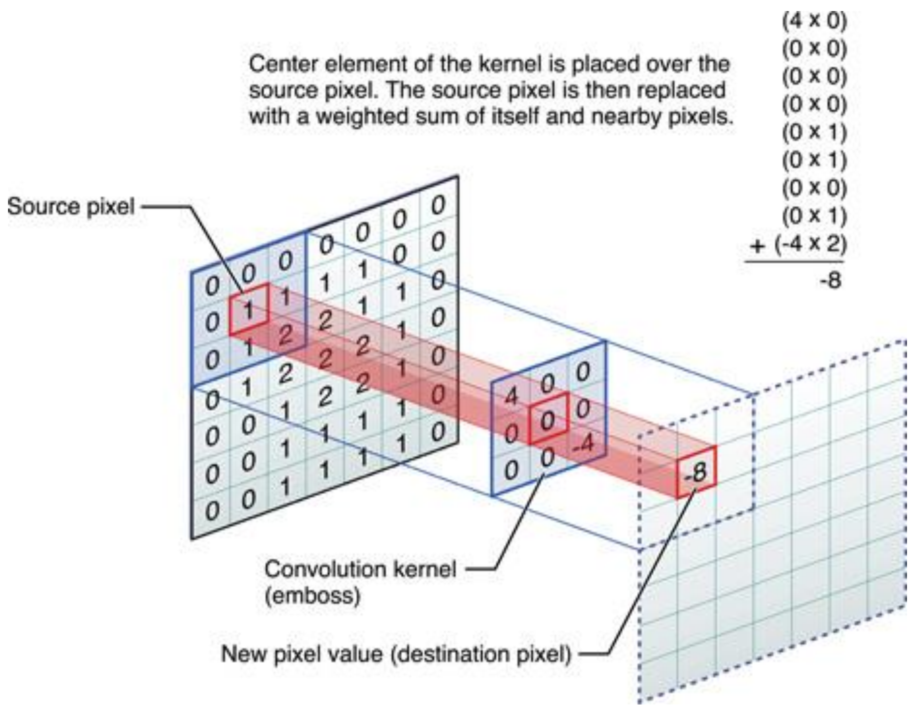
$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]\, g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]\, g[m] = (g * f)[n]$$

- Discrete domains with finite support:

$$(f * g)[n] = \sum_{m=-M}^{M} f[n-m]\, g[m]$$

2D convolution

- 2D convolution (a.k.a. spatial convolution) as linear spatial filtering



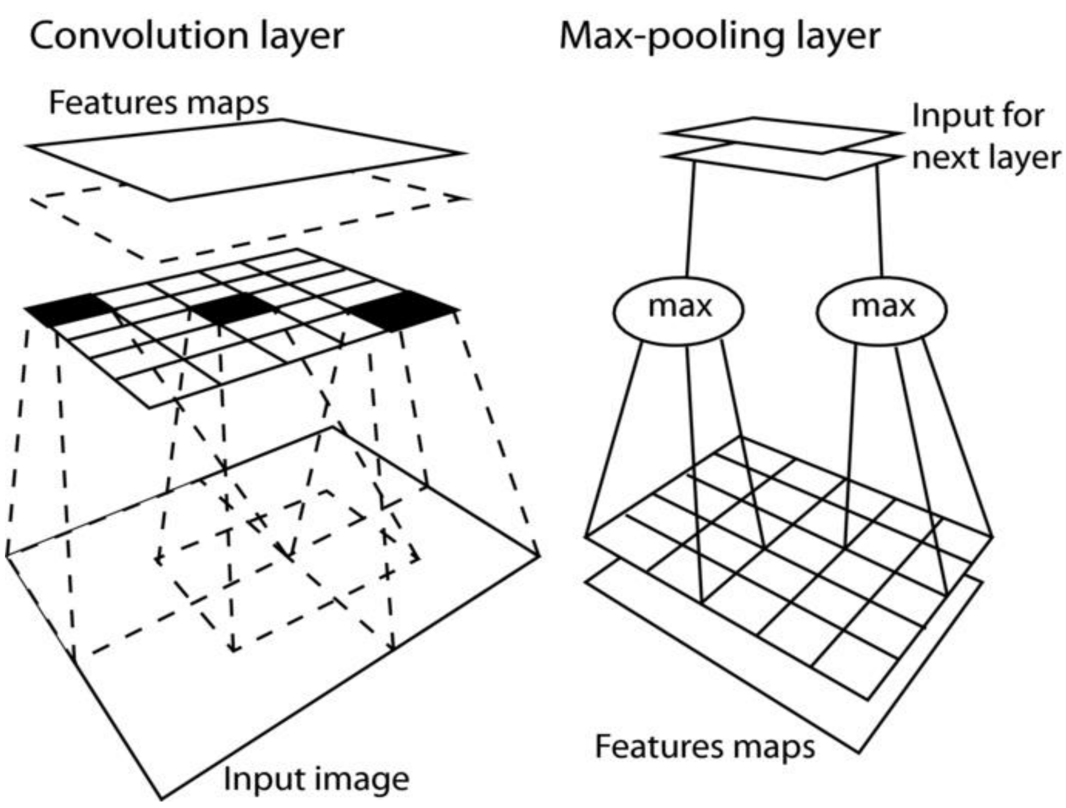
- Multiple feature maps, one for each convolution operator

Convolutional and pooling layers

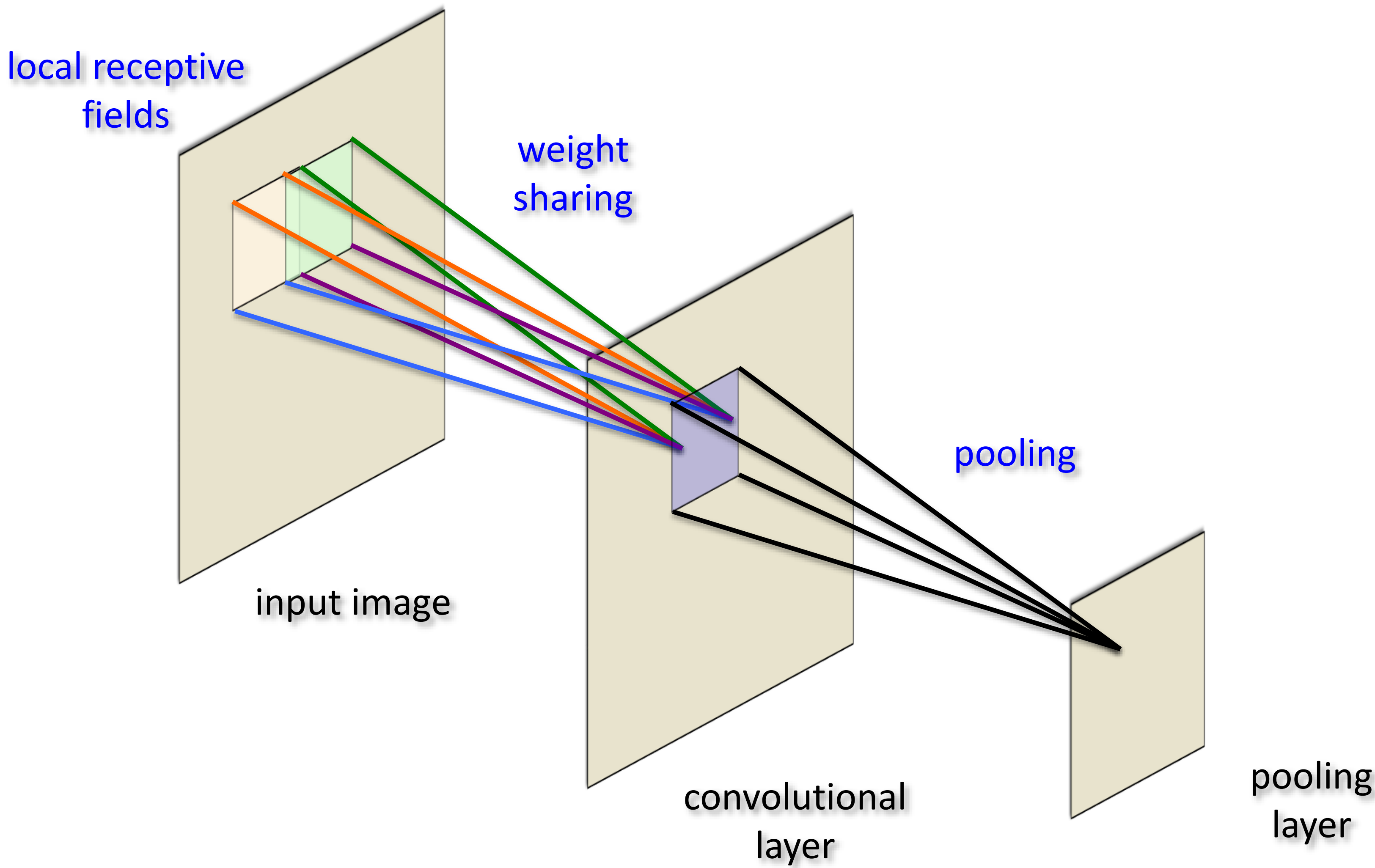
- Convolution: feature detector
- Max-pooling: local translation invariance

determines the future state of a certain cell in the grid by the inputs and past states of its local neighbors

Size of state-to-state convolutional kernel for capturing of spatiotemporal motion patterns



Convolutional and pooling layers



NN and Fully-connected Recurrent NN

Feed-forward NN

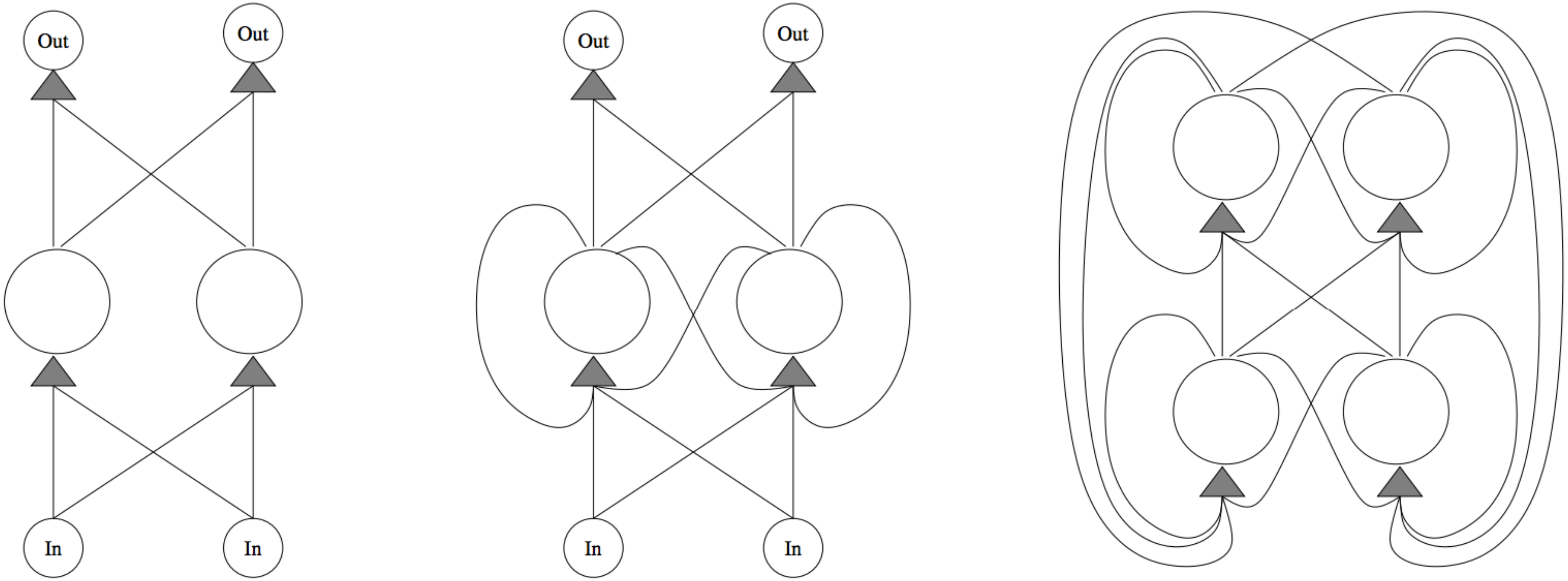


Figure 1.1: Left: Feed-forward neural network. Middle: Layered network with an input layer, a fully recurrent hidden layer and an output layer. Right: Fully connected recurrent network.

From RNN to LSTM

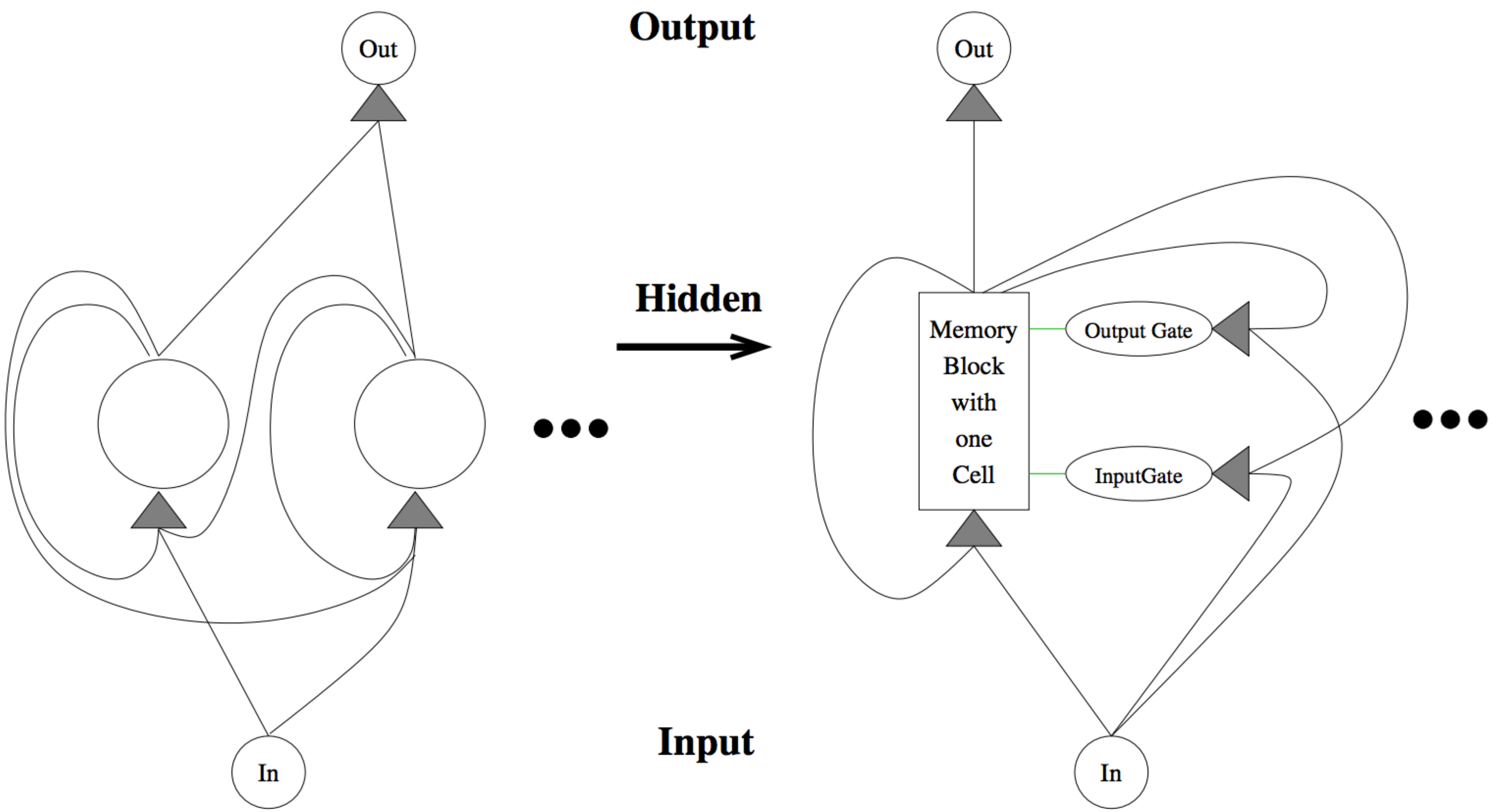
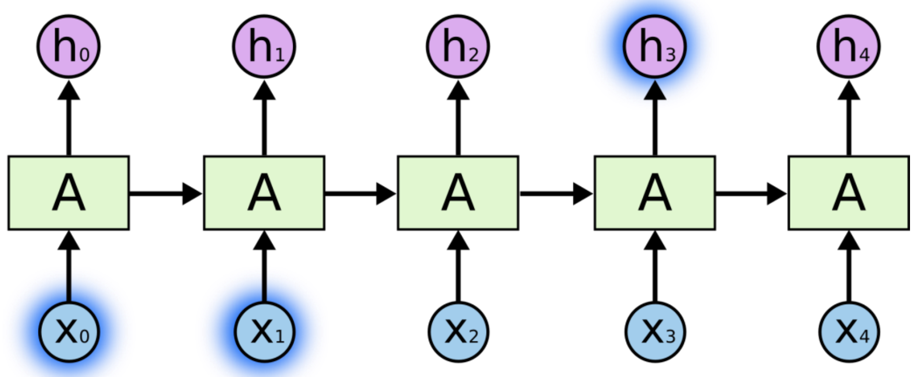


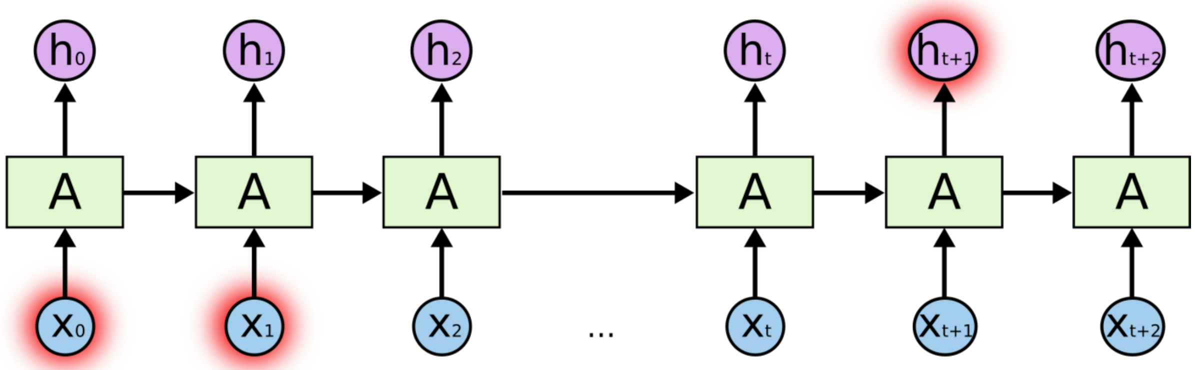
Figure 2.1: Left: RNN with one fully recurrent hidden layer. Right: LSTM network with memory blocks in the hidden layer (only one is shown).

Dependencies between events in RNNs

- Short-term dependencies:

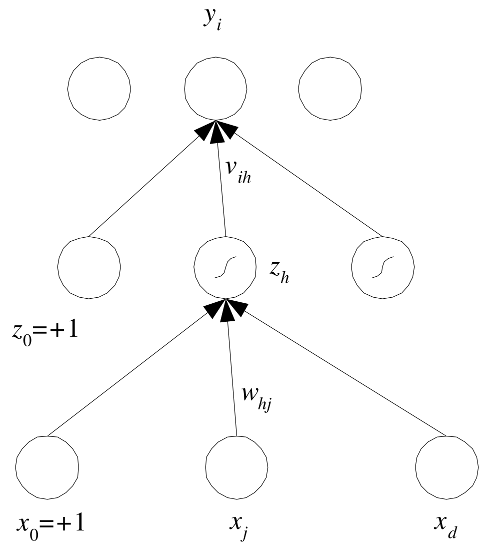


- Long-term dependencies:



Ordinary hidden units in multilayered networks

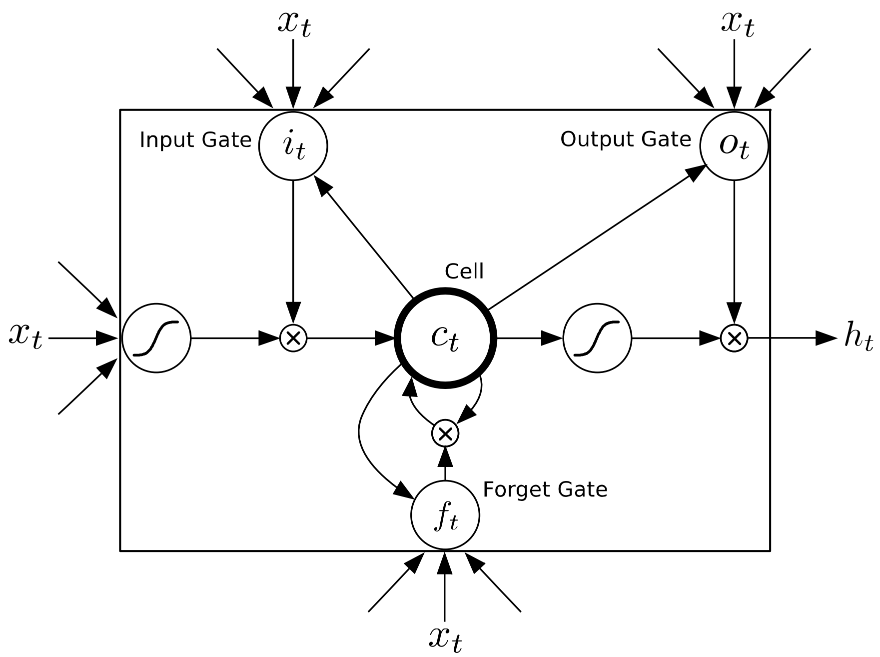
- Nonlinear function (e.g., sigmoid or hyperbolic tangent) of weighted sum

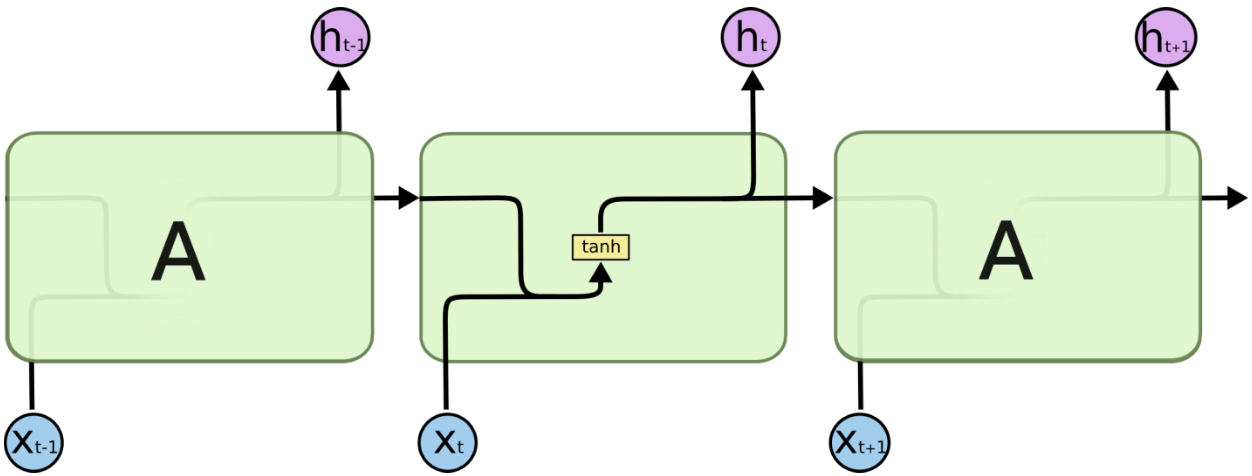


- RNNs, like deep multilayered networks, suffer from the vanishing gradient problem

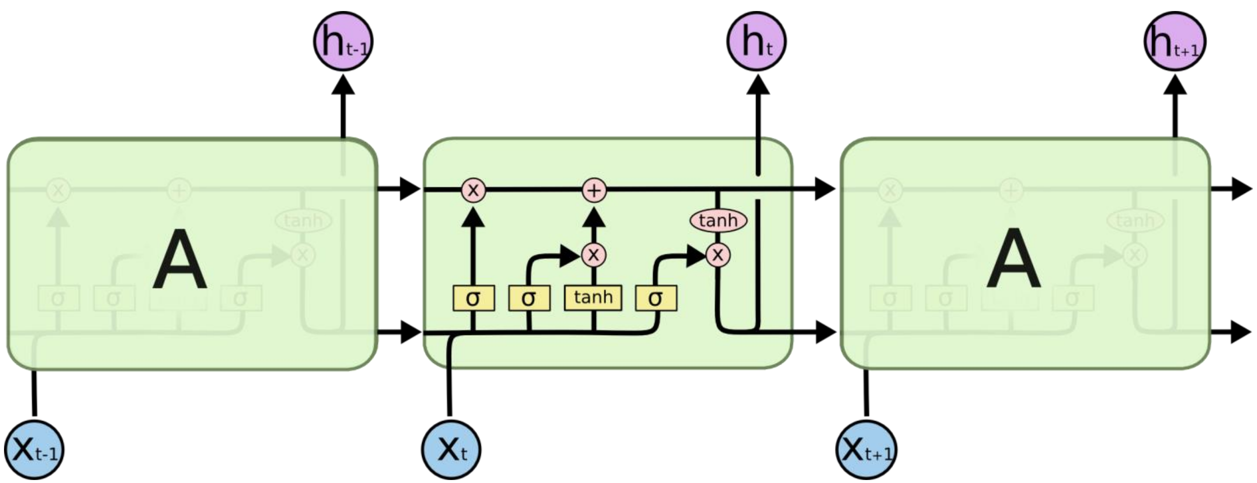
LSTM units

- LSTM units, which are essentially subnets, can help to learn long-term dependencies in RNNs
- 3 gates in an LSTM unit: input gate, forget gate, output gate





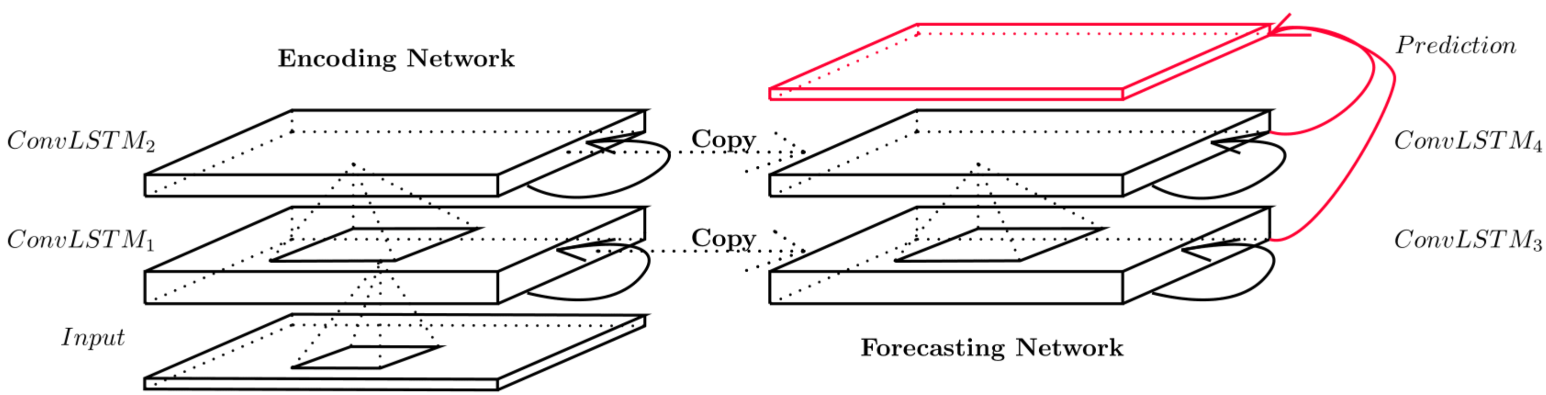
RNNs with ordinary unit



RNNs with LSTM units

Encoding-forecasting ConvLSTM network

- Last states and cell outputs of encoding network become initial states and cell outputs of forecasting network
- Encoding network compresses the input sequence into a hidden state tensor
- Forecasting network unfolds the hidden state tensor to make prediction



ConvLSTM governing equations

Accumulator of state information

Inputs

Memory cell

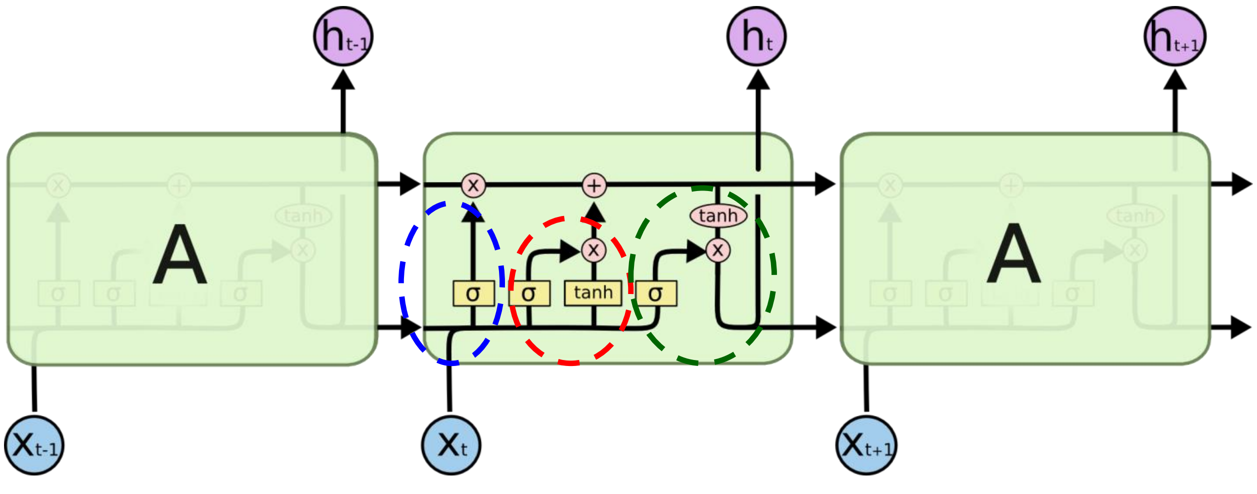
input gate $i_t = \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i)$

forget gate $f_t = \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f)$

Cell outputs $\mathcal{C}_t = f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c)$

output gate $o_t = \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o)$

Hidden states $\mathcal{H}_t = o_t \circ \tanh(\mathcal{C}_t)$



Training and preprocessing of radar echo dataset

- 97 days in 2011-2013 with high radar intensities

- Preprocessing of radar maps:
  - Pixel values normalized
  - 330 x 330 central region cropped
  - Disk filter applied
  - Resized to 100 x 100
  - Noisy regions removed

Data splitting

- 240 radar maps (a.k.a. frames) per day partitioned into six 40-frame blocks

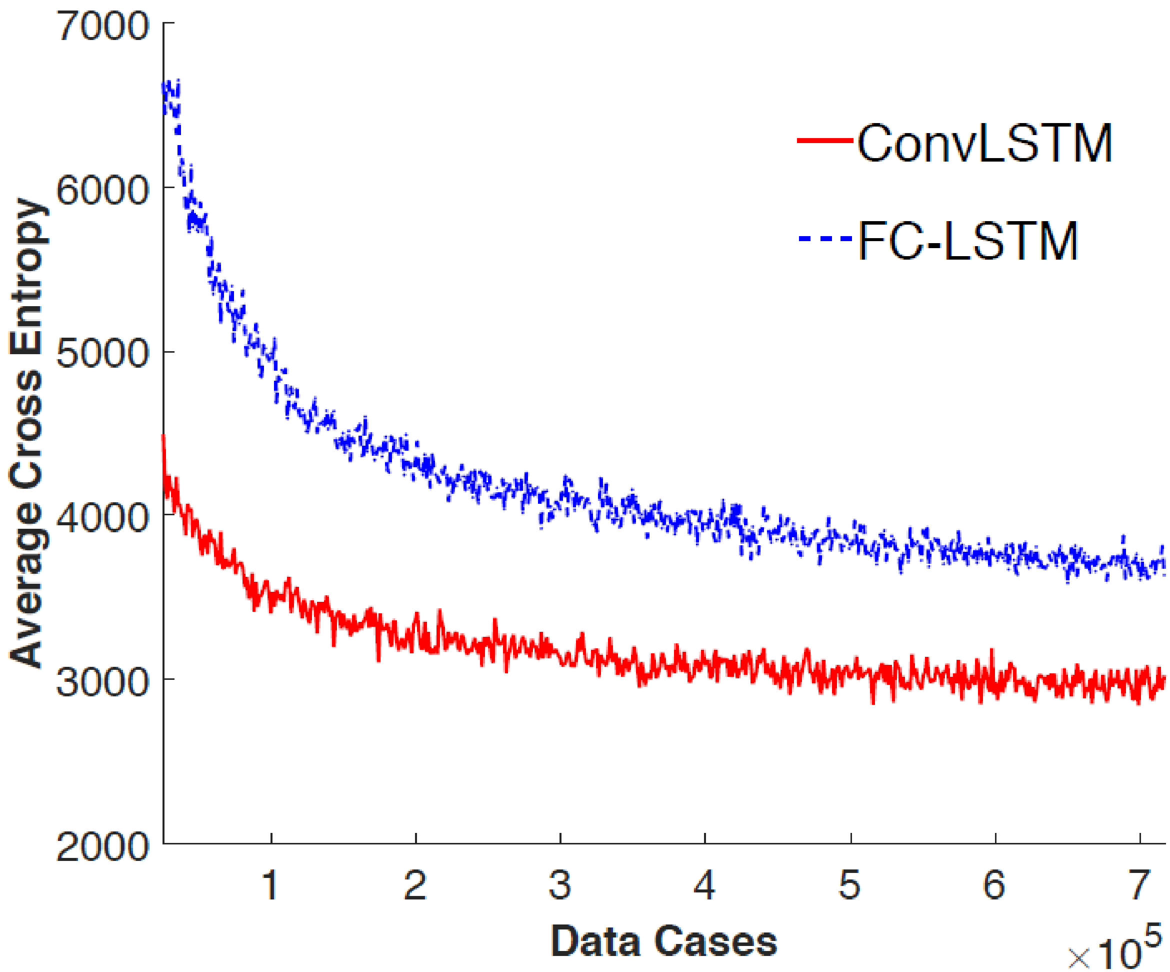
- Random data splitting:
  - Training: 8148 sequences
  - Validation: 2037 sequences
  - Testing: 2037 sequences

- 20-frame sequence :
  - Input sequence: 5 frames
  - Output sequence: 15 frames (i.e., 6-90 minutes)

Comparison of performance

- ConvLSTM network:
  - 2 ConvLSTM layers, each with 64 units and 3 x 3 kernels
- Fully connected LSTM (FC-LSTM) network:
  - 2 FC-LSTM layers, each with 2000 units
- **ROVER:**
  - Optical flow estimation
  - 3 variants (ROVER1, ROVER2, ROVER3) based on different initialization schemes

Comparison of ConvLSTM and FC-LSTM



the loss of entropy for ConvLSTM decreases faster than FC-LSTM across all the data cases

➔ a better matching with training datasets

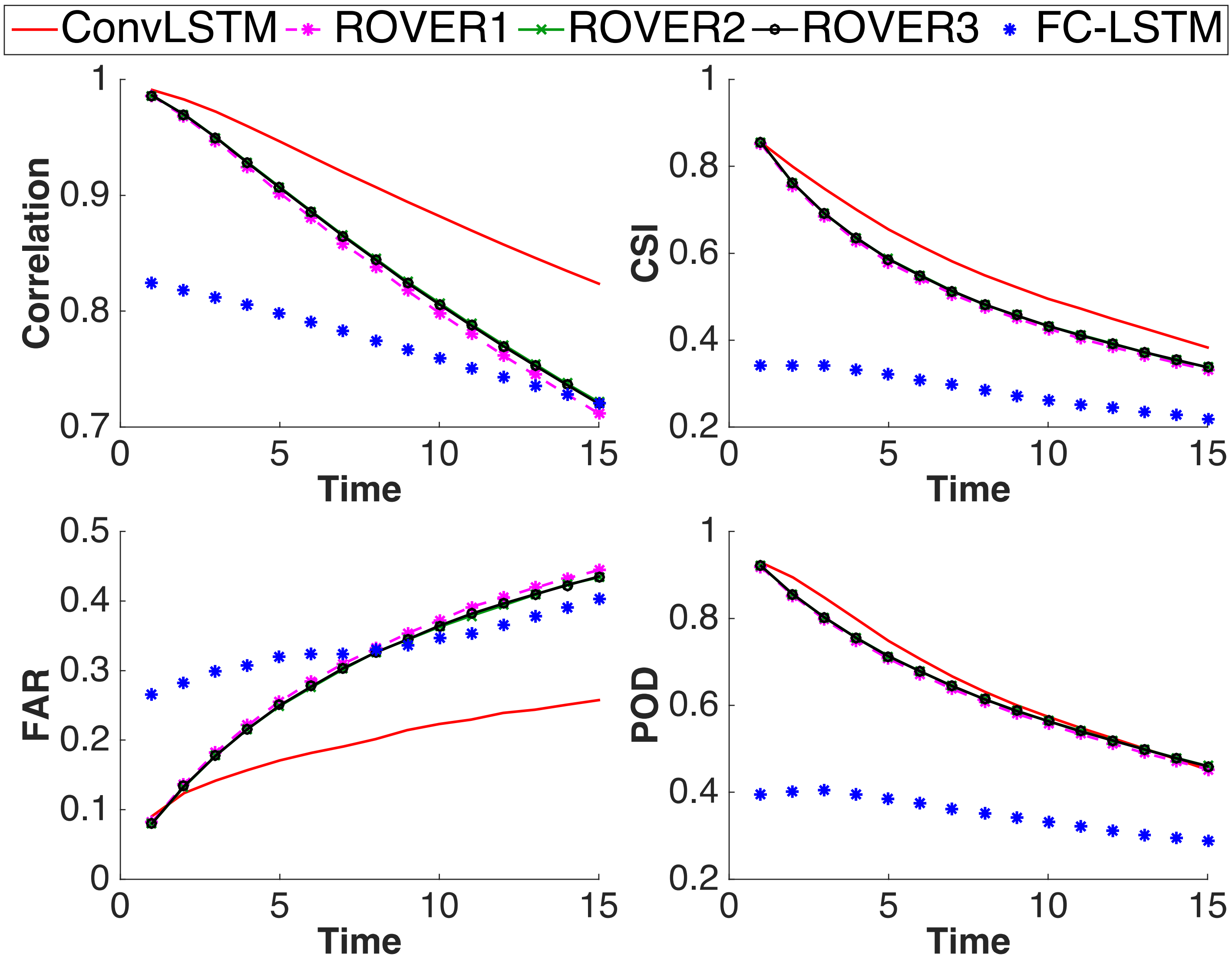
Comparison based on 5 performance metrics

- Rainfall mean squared error (Rainfall-MSE)
- Critical success index (CSI)
- False alarm rate (FAR)
- Probability of detection (POD)
- Correlation

Model	Rainfall-MSE	CSI	FAR	POD	Correlation
ConvLSTM(3x3)-3x3-64-3x3-64	**1.420**	**0.577**	**0.195**	**0.660**	**0.908**
Rover1	1.712	0.516	0.308	0.636	0.843
Rover2	1.684	0.522	0.301	0.642	0.850
Rover3	1.685	0.522	0.301	0.642	0.849
FC-LSTM-2000-2000	1.865	0.286	0.335	0.351	0.774

Threshold = 0.5 mm/h

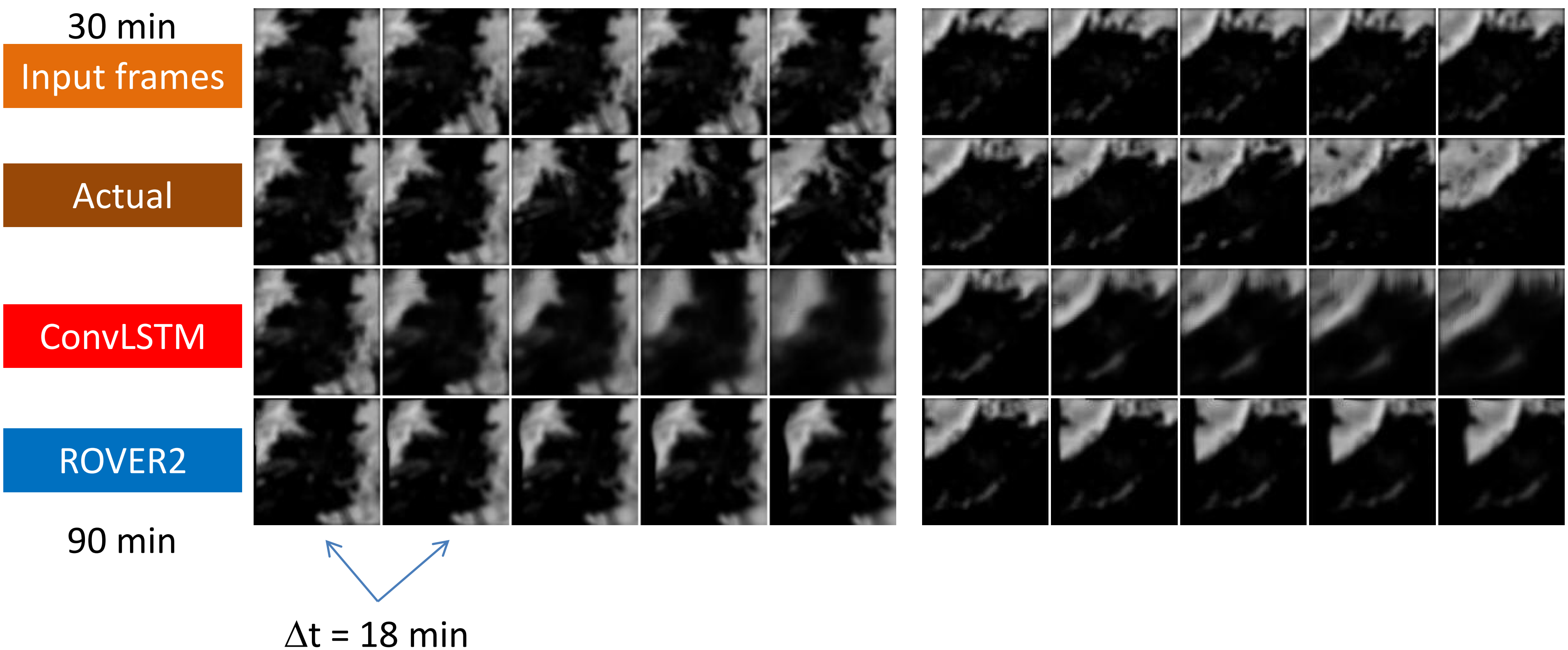
Prediction accuracy vs prediction horizon



Different parameters are used in ROVER1,2,3 optical flow estimators

Two squall line cases

- Radar location (HK) at center (~ 250 km in x- and y- directions)
- 5 input frames are used and a total of 15 frames (i.e. T+90 min) in forecasts



30 min

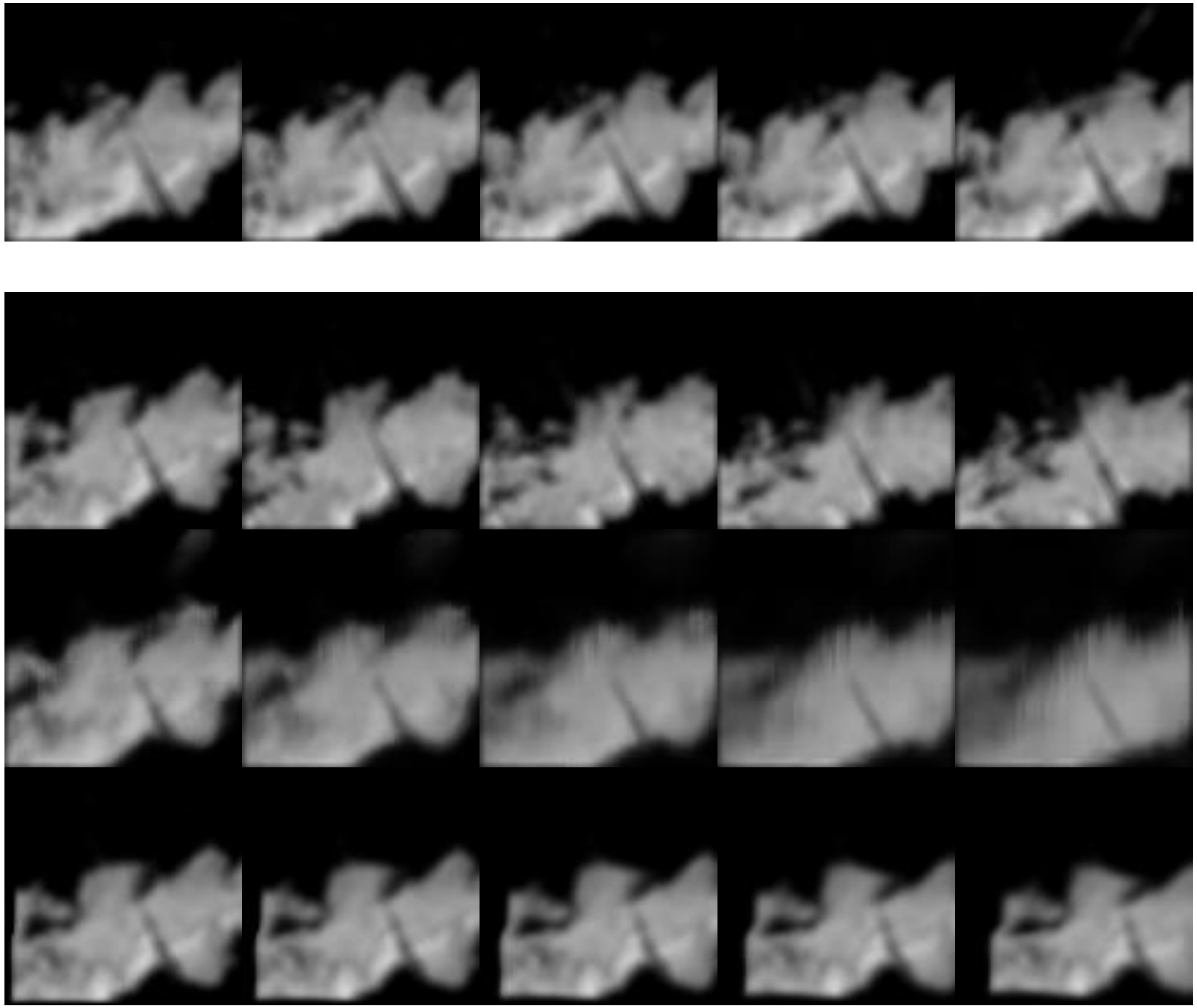
Input frames

90 min

Actual

ConvLSTM

ROVER2



30 min

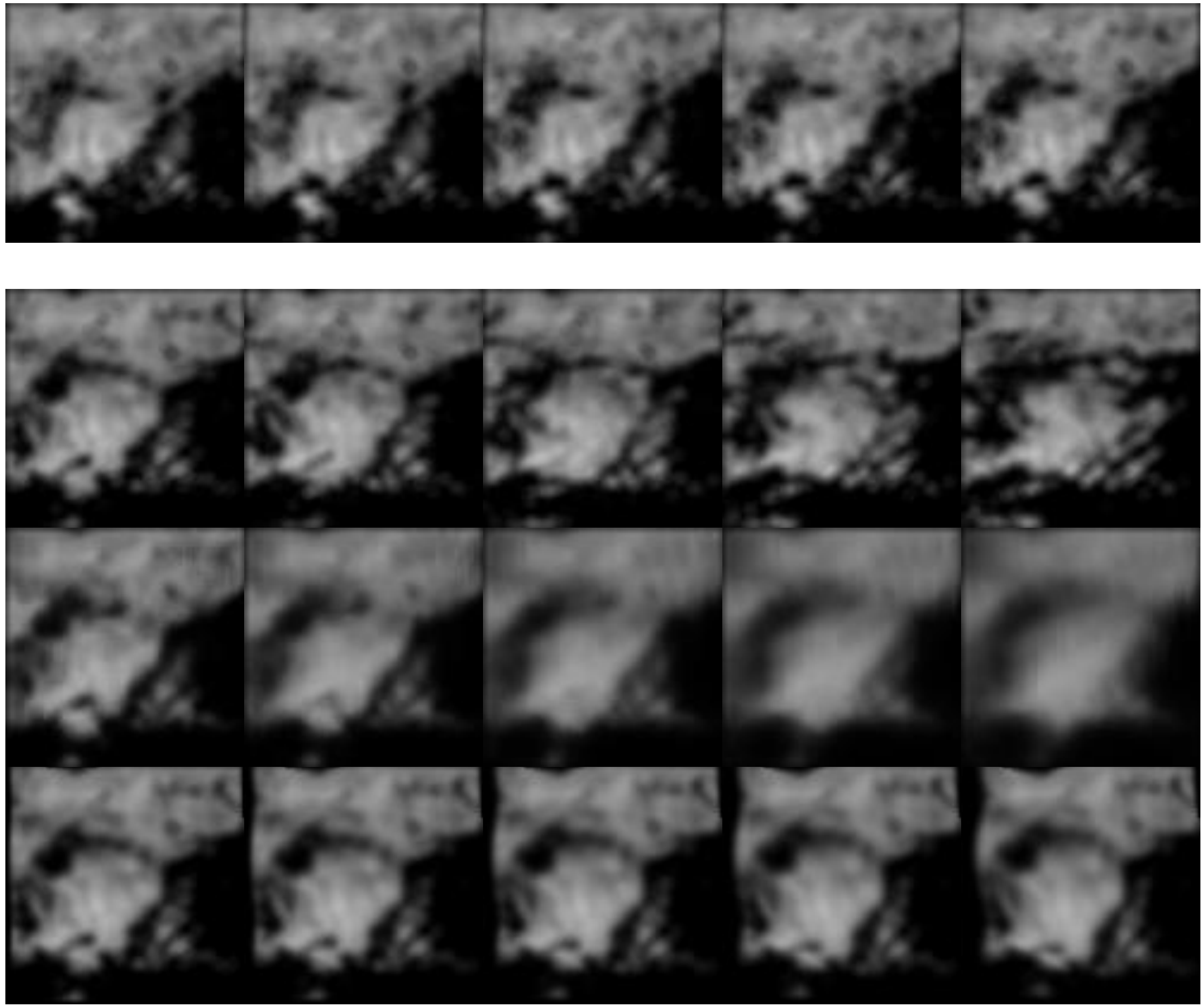
Input frames

90 min

Actual

ConvLSTM

ROVER2



Ongoing Development

- Longer training dataset (~ 10 years data)
- Adaptive learning to cater for multiple time scale processes
- Optimizing performance for higher rainfall intensity based on different convolutional and pooling strategies
- Extend learning process to extract stochastic characteristics of radar echo time sequence, features of convective development from mesoscale/fine-scale NWP models