Machine Learning Specialist

Convolutional LSTM & PredRNN Models

Stanford University

Teaching

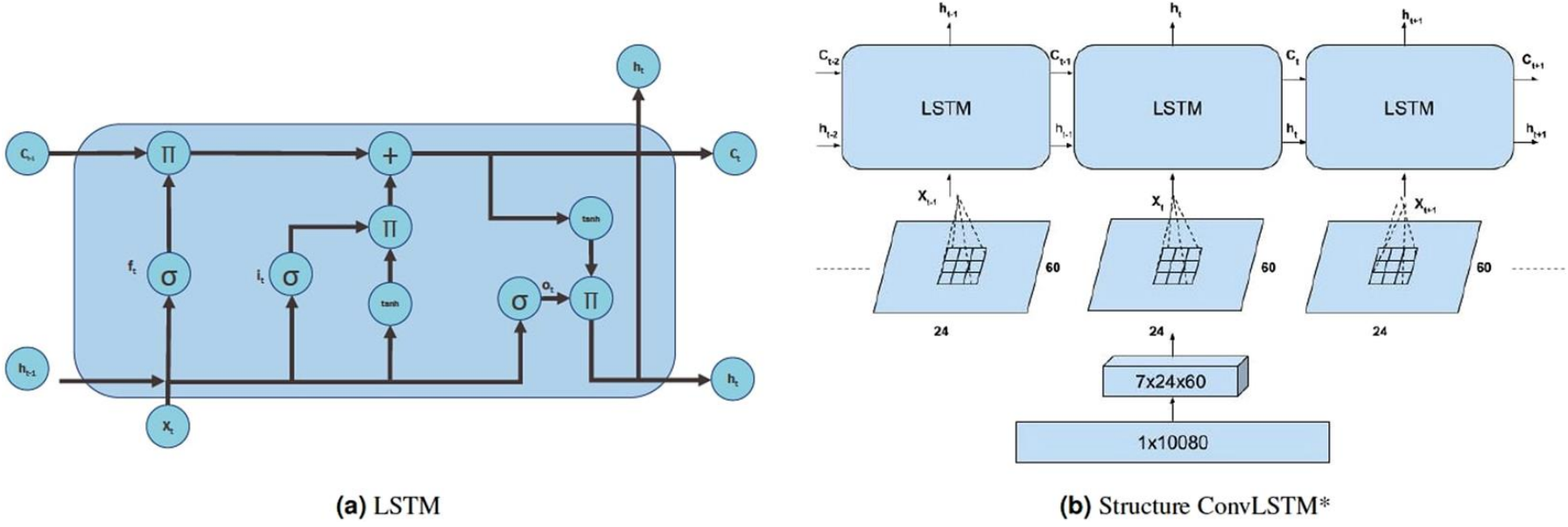
Model Investigation:

- Convolutional LSTM

Convolutional LSTM (ConvLSTM) is a neural network architecture which combines the convolutional operations with the Long Short-Term Memory (LSTM) units to process spatiotemporal data effectively.

Spatiotemporal Data??
A Spatiotemporal data refers to information that combines both spatial (space) and temporal (time) information

- Convolutional LSTM



(a) LSTM **(b)** Structure ConvLSTM*

(**a**) Basic LSTM cell. Π denotes multiplication, + denotes addition, *σ* is sigmoid function, and *tanh* calculates hyperbolic tangent, (**b**) Transformation and inner structure of ConvLSTM*.

Rahman, S.A., Adjeroh, D.A. Deep Learning using Convolutional LSTM estimates Biological Age from Physical Activity. *Sci Rep* **9**, 11425 (2019). https://doi.org/10.1038/s41598-019-46850-0

- Convolutional LSTM

The architecture of the ConvLSTM model consists of four components:

- **Convolutional Layers**
- **Forget Gate**
- **Input Gate**
- **Output Gate**

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot \mathcal{C}_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot \mathcal{C}_{t-1} + b_f)$$

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

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

$$\mathcal{H}_t = o_t \odot \tanh(C_t)$$

Please note: * denotes the convolution operator and ⊙ represents the Hadamard product.

- Convolutional LSTM

1. Input Gate

The input gate controls how much of the new input is allowed into the cell state:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$

Where:

- i_t is the input gate at time step t,
- σ is the sigmoid activation function,
- W_{xi} and W_{hi} are convolutional weights for the input X_t and the previous hidden state H_{t-1},
- W_{ci} is a convolutional weight for the previous cell state C_{t-1},
- b_i is the bias for the input gate.

- Convolutional LSTM

2. Forget Gate

The forget gate decides how much of the previous cell state will be retained:

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f)$$

Where:

- f_t is the forget gate at time step t,
- W_{xf}, W_{hf}, and W_{cf} are the convolutional weights for the input, hidden state, and cell state, respectively,
- b_f is the bias for the forget gate.

- Convolutional LSTM

3. Cell State Update

The cell state C_t is updated by combining the new candidate cell state $\tilde{C}_t$ and the previous cell state:

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t$$

Where:

- C_t is the new cell state,
- $\circ$ denotes the Hadamard (element-wise) product,
- $\tilde{C}_t = \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c)$ is the candidate cell state.

- Convolutional LSTM

4. Output Gate

The output gate determines how much of the cell state will be output as the hidden state:

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o)$$

Where:

- o_t is the output gate at time step t,
- W_{xo}, W_{ho}, and W_{co} are the convolutional weights for the input, hidden state, and cell state, respectively,
- b_o is the bias for the output gate.

- Convolutional LSTM

5. Hidden State

The hidden state is updated based on the new cell state:

$$H_t = o_t \circ \tanh(C_t)$$

Where:

- H_t is the new hidden state at time step t,
- tanh is the hyperbolic tangent activation function applied to the cell state.

Applications:

- Video Prediction: Forecasting future frames in video sequences.
- Precipitation Nowcasting: Predicting short-term rainfall patterns.
- Impulsive Sound Detection: Identifying sudden, short-duration sounds in urban environments.
- Activity Recognition: Generating textual descriptions of activities from image sequences.
- Image and Video Description: Creating captions for images and videos.

Teaching

Model Investigation:

- PredRNN

PredRNN is a recurrent neural network model which is designed for spatiotemporal predictive learning, particularly focusing on predicting the future frames in a sequence of videos. The model was formulated to address the challenges in video prediction tasks by combining techniques from convolutional neural networks (CNNs) and recurrent neural networks (RNNs)

The core of this network is a novel Spatiotemporal LSTM (ST-LSTM) unit that extracts and memorizes spatial and temporal representations simultaneously.

Model Investigations:

- PredRNN

Key features of PredRNN:

- Zigzag Memory Flow: Memory states are allowed to flow in two directions - vertically across stacked RNN layers and horizontally through all RNN states.
- Unified Memory Pool: It combines spatial and temporal information in a single memory cell and is better for spatiotemporal modelling.
- State-of-the-Art Performance: PredRNN has achieved top performance on various video prediction datasets, and thus has outperformed previous models.

Applications:

- Video Frame Prediction: The primary use is predicting future frames in video sequences (It is a fundamental problem in computer vision). It is also applied for situations such as video surveillance, autonomous driving, weather forecasting, and others.

- Traffic Flow Prediction: Traffic flow is a spatiotemporal problem (evolving in both space and time), PredRNN is used to predict traffic conditions based on historical data.

- Climate Forecasting: The ability to model spatiotemporal patterns makes PredRNN suitable for climate prediction, where changes occur across both spatial and temporal dimensions.

Similarity Discussions:

Similarities between Convolutional LSTM and PredRNN:

- Spatiotemporal Modelling: Both the models are designed for spatiotemporal data (i.e., video prediction and others). They aim to capture both the spatial and temporal dependencies in sequential data.

- Use of Convolutional Layers: Both Convolutional LSTM and PredRNN integrate convolutional operations with recurrent architecture which makes them ideal for spatial data like video frames. The convolutional layers allows them to process image-like data while capturing local spatial patterns.

- Sequential Processing: Both architectures process sequences of data step by step, updating hidden states and memory cells over time which allows for capturing the temporal dynamics in the data.

Comparison Discussions:

Comparison discussions between Convolutional LSTM and PredRNN:

Memory Mechanism:

- Convolutional LSTM: The Convolutional LSTM is a standard LSTM with convolutional operations instead of fully connected layers. It only considers the temporal dependencies through time and spatial information via convolution in each time step. It has a single memory cell that is updated at each time step.

- PredRNN: PredRNN introduces the Spatiotemporal LSTM (ST-LSTM), which extends the Convolutional LSTM by maintaining not just a temporal memory but also spatial memory. This means that PredRNN can model interactions not only over time but also across different spatial locations.

Handling Spatiotemporal Correlations:

- Convolutional LSTM: The Convolutional LSTM models temporal dependencies in the sequence of convolutional features. It processes frames in sequential manner.

- PredRNN: PredRNN enhances the handling of spatiotemporal correlations by introducing a separate memory state for spatial information. This enables the model to propagate both spatial and temporal memory states, making it more effective at capturing complex spatiotemporal patterns compared to Convolutional LSTM.

Comparison Discussions:

Comparison discussions between Convolutional LSTM and PredRNN:

Prediction Capabilities:

- Convolutional LSTM: It is good for tasks where both temporal and spatial patterns are simple or do not interact in complex ways.
- PredRNN: With its enhanced memory flow and the introduction of ST-LSTM, PredRNN is better suited for more complex video prediction tasks, where the spatial patterns evolve dynamically over time.

Architecture Complexity:

- Convolutional LSTM: The architecture is simple, and it focuses on modifying the LSTM unit to work with convolutional data.
- PredRNN: The architecture is complex due to the use of dual memory mechanisms (temporal and spatial) and the added complexity of the ST-LSTM.

Prediction examples on the radar echo test set, in which 10 future frames are generated from the past 10 observations.

Comparison:

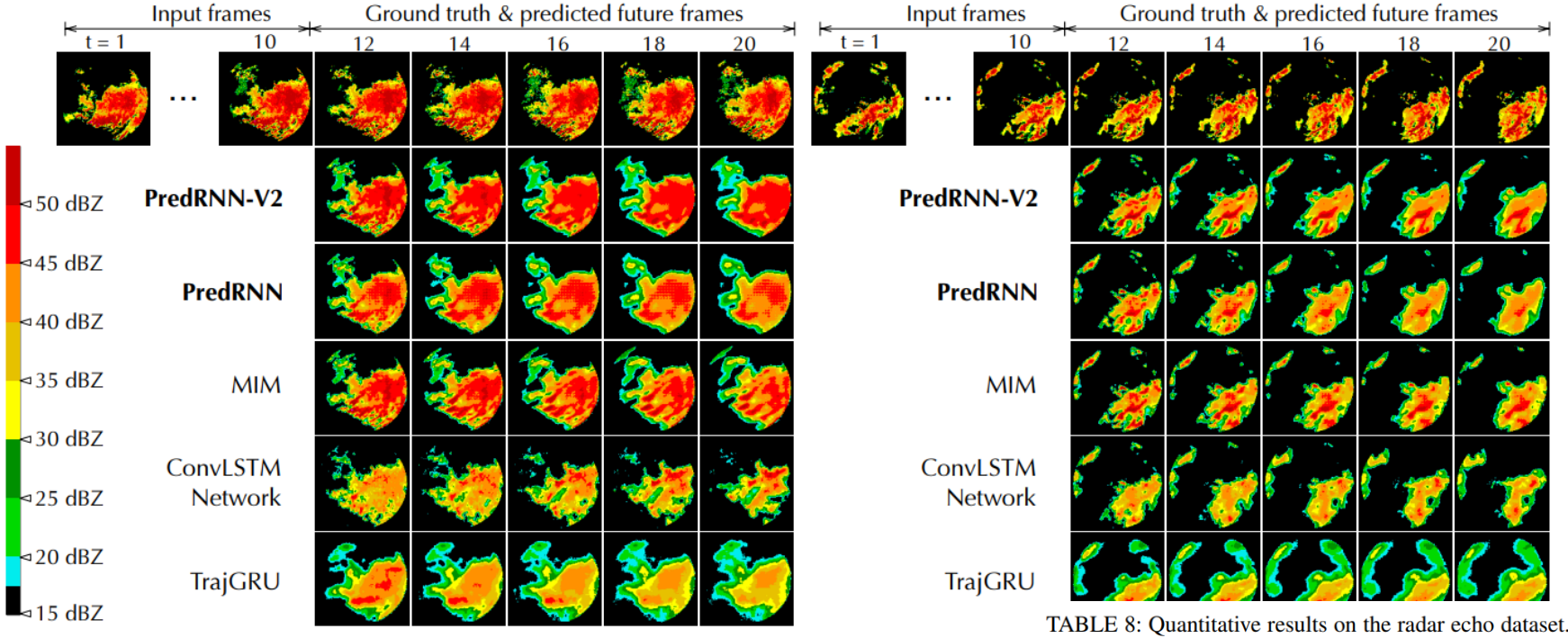


TABLE 8: Quantitative results on the radar echo dataset.

Model	MSE (↓)	CSI-30 (↑)	CSI-40 (↑)	CSI-50 (↑)
TrajGRU [25]	68.3	0.309	0.266	0.211
ConvLSTM [1]	63.7	0.381	0.340	0.286
MIM [4]	39.3	0.451	0.418	0.372
PredRNN	39.1	0.455	0.417	0.358
PredRNN-V2	**36.4**	**0.462**	**0.425**	**0.378**

Wang Y, Wu H, Zhang J, Gao Z, Wang J, Philip SY, Long M. Predrnn: A recurrent neural network for spatiotemporal predictive learning. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022 Apr 5;45(2):2208-25.

Thank you