Preparing your manuscript

Florian Börgel Sven Karsten

9/29/23

The abstract (1) states the nature of the investigation and (2) summarizes the important conclusions. The abstract should be suitable for indexing. Your abstract should:

Introduction

River runoff is an important component of the global water cycle as it comprises about one third of the precipitation over land areas (Hagemann, 2020). Moreover, accurate runoff forecasting, especially over extended periods, is pivotal for effective water resources management, as highlighted by studies such as Yang et al. (2018), Tan et al. (2018), and Fang et al. (2019).

Over the past decades, models for long-term runoff forecasting have been primarily bifurcated into physically based models and data-based models. While the former attempts to emulate intricate and nonlinear physical hydrological processes, the latter hinges on establishing statistical models that delineate the relationship between large-scale climate patterns and catchment runoff.

Machine Learning (ML) models, such as those employing artificial neural networks, support vector machines, adaptive neuro-fuzzy inference systems, and notably, Long Short-Term Memory (LSTM) neural networks, have gained traction for long-term hydrological forecasting due to their commendable performance (Humphrey et al 2016, Huang et al 2014, Ashrafi et al 2017, Yuan et al 2018, Xu et al 2021).

LSTM networks, an evolution of the classical Recurrent Neural Networks (RNNs), have shown stability and efficacy in sequence-to-sequence predictions, such as using climatic indices for rainfall estimation or long-term hydrological forecasting. However, a limitation of LSTMs is their inability to effectively capture two-dimensional structures, an area where Convolutional Neural Networks (CNNs) excel. Recognizing this, we introduce the ConvLSTM, which integrates the strengths of both LSTM and CNN, to extract spatiotemporal features from precipitation fields for predicting river runoff in the Baltic Sea catchment, summarized by 97 inidivual rivers.

Modeling the Baltic Sea is to a large part the result of the quality of the freshwater input, that is used for the simulation. Meier and Kauker(2003) showed that decadal salinity variations of about 1 $g\ kg^{-1}$ are caused, inter alia, by annual runoff variations. Further, Meier and Kauker (2003) showed that about 50 % of the decadal salinity variability can be explained by variations in freshwater input into the Baltic Sea.

This paper delves into the application of deep learning, particularly ConvLSTM, to the challenging task of precipitation nowcasting, a domain yet to fully harness the potential of advanced machine learning techniques. We present ConvLSTM as a novel solution to this spatiotemporal sequence forecasting challenge, highlighting its advantages and potential future applications.

Methods

LSTM network

The Long Short-Term Memory (LSTM), a specialized form of Recurrent Neural Networks (RNNs), is specifically tailored for modeling temporal sequences. Its unique design allows it to adeptly handle long-range dependencies, setting it apart from traditional RNNs in terms of accuracy (see Figure 1).

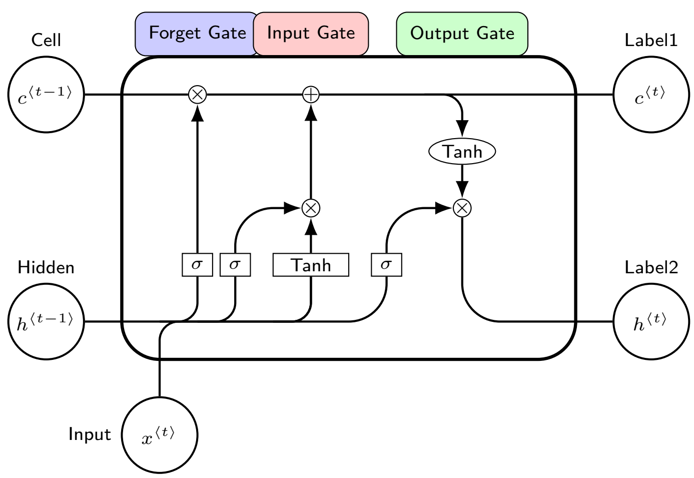

Figure 1: Inner structure of a Long Short-Term Memory Cell

This performance in modeling long-range dependencies has been validated in various studies. The key component of LSTM's innovation is its memory cell, c_tt, which stores state information, also refered to as long-term memory. This cell is accessed, modified, and reset through several self-parameterized gates. For the input of the sequence x_t input, the forget gate f_t defines the percentage of the previous long-term memory status c_{t-1} that should be retained stored. Next the input gate i_t decides how much of the input is added to the the long-term memory, forming the updated cell state c_t. The decision to propagate the latest cell output, c_t, to the final state, h_t, is governed by the output gate, o_t, representing the updated short-term memory of the hidden state h_t. A significant advantage of this architecture is the memory cell's ability to retain gradients. This mechanism addresses the vanishing gradient problem, where, as input sequences elongate, the influence of initial stages becomes harder to capture, causing gradients of early input points to approach zero. The LSTM's activation function, inherently recurrent, mirrors the identity function with a consistent derivative of 1.0, ensuring the gradient remains stable throughout backpropagation.

One LSTM cell hence maybe expressed as:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

with

- x_t: Input vector at time step t.
- h_{t-1}: Hidden state from the previous time step.
- C_{t-1}: Cell state from the previous time step.
- W and b: Weight matrices and bias vectors, respectively, associated with the gates of the LSTM. The subscripts denote the specific gate or operation they are associated with (e.g., W_f and b_f are the weight matrix and bias for the forget gate, respectively).
- σ: Sigmoid activation function, which squashes values between 0 and 1.
- tanh: Hyperbolic tangent activation function, which squashes values between -1 and 1

ConvLSTM network

The FC-LSTM fails to handle information when handling spatiotemporal data due to its reliance on full connections in both input-to-state and state-to-state transitions. To adress this limitation we use a convLSTM architecture. convLSTM replaces the fully connected operations in the LSTM with convolutional operations. Hence, all inputs $X_1, \dots, X_t$, cell outputs $C_1, \dots, C_t$, hidden states $H_1, \dots, H_t$, and gates i_t, f_t, o_t of the ConvLSTM are 3D tensors. The last two dimensions of these tensors represent spatial dimensions, specifically rows and columns. Conceptually, these inputs and states can be visualized as vectors positioned on a spatial grid.

In the ConvLSTM, the future state of a specific cell on this grid is determined by the inputs and past states of its neighboring cells. This spatial consideration is integrated by employing a convolution operator in both state-to-state and input-to-state transitions, as illustrated in Fig. 2. The foundational equations for ConvLSTM are:

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
H_t &= o_t \circ \tanh(C_t)
\end{aligned}
$$

In summary, the ConvLSTM excels at processing tasks that demand a combined understanding of spatial patterns and temporal sequences in data. It merges the image-processing capabilities of Convolutional Neural Networks (CNNs) with the time-series modeling of Long Short-Term Memory (LSTM) networks.

Implemented model architecture

We first implemented the ConvLSTM using and encoder/decoder structure as discussed in ... . To predict all 97 rivers entering the Baltic Sea at once, we flatten the output and use fully connected layers resulting in 97 outputs.

For the computation we use the following set of hyper parameters:

Table 1: Hyperparameters

Parameter name	Parameter size
Num. timesteps	32
Conv. Kernelsize	(7,7)
Num. ConvLSTM layers	4
Batch size	32
Learning Rate	1e-4 with CosineAnnealing

Acknowledgments

Phasellus interdum tincidunt ex, a euismod massa pulvinar at. Ut fringilla ut nisi nec volutpat. Morbi imperdiet congue tincidunt. Vivamus eget rutrum purus. Etiam et pretium justo. Donec et egestas sem. Donec molestie ex sit amet viverra egestas. Nullam justo nulla, fringilla at iaculis in, posuere non mauris. Ut eget imperdiet elit.

Open research

Phasellus interdum tincidunt ex, a euismod massa pulvinar at. Ut fringilla ut nisi nec volutpat. Morbi imperdiet congue tincidunt. Vivamus eget rutrum purus. Etiam et pretium justo. Donec et egestas sem. Donec molestie ex sit amet viverra egestas. Nullam justo nulla, fringilla at iaculis in, posuere non mauris. Ut eget imperdiet elit.

References