A Deep Learning Architecture for Conservative Dynamical Systems: Application to Rainfall-Runoff Modeling

Grey Nearing
Google Research
gsnearing@google.com

Frederik Kratzert
LIT AI Lb & Institute for Machine Learning, Johannes Kepler University
kratzert@ml.jku.at

Daniel Klotz
LIT AI Lab & Institute for Machine Learning, Johannes Kepler University
klotz@ml.jku.at

Pieter-Jan Hoedt
LIT AI Lab & Institute for Machine Learning, Johannes Kepler University
hoedt@ml.jku.at

Günter Klambauer
LIT AI Lab & Institute for Machine Learning, Johannes Kepler University
klambauer@ml.jku.at

Sepp Hochreiter
LIT AI Lab & Institute for Machine Learning, Johannes Kepler University
hochreit@ml.jku.at

Hoshin Gupta
Department of Hydrology and Atmospheric Sciences, University of Arizona
hoshin@arizona.edu

Sella Nevo
Google Research
sellanevo@google.com

Yossi Matias
Google Research
yossi@google.com

Abstract

The most accurate and generalizable rainfall-runoff models produced by the hydrological sciences community to-date are based on deep learning, and in particular, on Long Short Term Memory networks (LSTMs). Although LSTMs have an explicit state space and gates that mimic input-state-output relationships, these models are not based on physical principles. We propose a deep learning architecture that is based on the LSTM and obeys conservation principles. The model is benchmarked on the mass-conservation problem of simulating streamflow.

AI for Earth Sciences Workshop at NeurIPS 2020.

1 Introduction

Due to computational challenges and to the increasing volume and variety of hydrologically-relevant Earth observation data, machine learning is particularly well suited to help provide efficient and reliable flood forecasts over large scales [12]. Hydrologists have known since the 1990's that machine learning generally produces better streamflow estimates than either calibrated conceptual models or process-based models [7]. Yet, both research and operational models in the field are dominated by the latter. One review article [14] framed the issue like this: *"physical process-oriented modellers have no confidence in the capabilities of data-driven models' outputs with their heavy dependence on training sets, while the more system engineering-oriented modellers claim that data-driven models produce better forecasts than complex physically-based models."*

The current state-of-the-art accuracy for streamflow modeling is by [9], who used Long Short Term Memory (LSTM) networks to produce daily streamflow simulations in several hundred catchments across the continental US (CONUS). Not only did LSTMs outperform all of the conceptual models available for benchmarking, LSTMs were more accurate in ungauged basins (basins where training data was withheld completely) than conceptual models calibrated to each individual basin [10] (all basins contributed training data). The problem of *Prediction in Ungauged Basins* was the decadal problem of the International Association of Hydrological Sciences from 2003-2012 [6].

Nevertheless, the argument that prediction of physical systems should benefit from theoretical understanding is compelling. Ideally, we would like a class of models that is able to capture (1) all of the information available from large data sets, and (2) all useful information available from the past decades of domain-specific scientific discovery. This requires *Knowledge-Guided Machine Learning* (KGML), which is an emerging branch of science in its own right [8].

This paper represents a step toward KGML for physical systems like watersheds. We investigate more in-depth a deep learning architecture with an explicit state-space (like the LSTM and most dynamical systems models), and that also obeys conservation laws (unlike the LSTM) [3]. Conservation is enforced through normalization layers in the deep learning architecture. This basic concept for embedding conservation symmetries into neural networks is simple and can be applied to almost any machine learning architecture - here we apply it to a time series model based on the LSTM.

2 Model Structure

The basic principle is to normalize a set of activation nodes and scale by the quantity to conserve:

$$\vec{o} = \hat{\sigma}(W \cdot \vec{f} + \vec{b}) \times c. \tag{1}$$

Here, $\vec{o}$ are a vector of N outputs from a particular activation layer, $\vec{f}$ are a vector of input features, W and $\vec{b}$ are trained weights and biases, and c is a (scalar) quantity that we want to conserve. The activation function can be any positive, bounded activation function (here we notate a sigmoid activation), and $\hat{\sigma}$ indicates that a normalization is applied to the output of the activation function. Normalization can be done in many ways - for example, L1 or softmax:

$$\hat{\sigma}_i(\cdot) = \frac{\sigma_i(\cdot)}{\sum_i^N \sigma_i(\cdot)}, \quad \hat{\sigma}_i(\cdot) = \frac{e^{\sigma_i(\cdot)}}{\sum_i^N e^{\sigma_i(\cdot)}}. \tag{2}$$

This basic principle can be used in many types of deep networks. As an example, consider the network illustrated in Figure 1, which shows a single timestep of a recurrent neural network. This network functions like a dynamical systems model in that it obeys a basic, overall conservation principle by using a discrete-time input/state/output relationship:

$$\vec{x_t} = \vec{x_{t-1}} + \vec{i_t} - \vec{o_t} \tag{3}$$

where $\vec{x_t}$ are model states at time t and $\vec{i_t}$, $\vec{o_t}$ are input and output fluxes across a control volume boundary. There are two types of model inputs: ones that are conserved, $\vec{c_t}$, and ones that are not conserved, $\vec{a_t}$, which we will call auxiliary inputs. Each normalized activation layer, $\hat{\sigma}$, results in a set of fluxes (mass, energy, momentum, etc.) from either inputs to states, $\vec{i_t}$, states to states, R_t, or states to outputs, $\vec{o_t}$. States, $\vec{x_t}$, and outputs, $\vec{o_t}$, are in the same physical units as the conserved

inputs. Each normalized activation layer, $\hat{\sigma}$ acts on all auxiliary inputs, $\vec{a_t}$, and the current state, $\vec{x_t}$. Because the states are storage states in physical units, they are not necessarily bounded above (although typically bounded below at zero), so can grow to overwhelm the activation functions. As such, normalized values of the states are used in all activation gates. The redistribution matrix, R_t, is a column-normalized matrix, where each column is scaled by the current value of the corresponding state so that each element in this matrix is a flux from one state to another.

Because all data contain error and because of the possibility for unobserved sources and sinks in a physical system, it is necessary sometimes for dynamical models to only loosely enforce conservation constraints. This can be achieved in two ways: by having either a subset of states, x_t or a subset of outputs, o_t. As an example, in the case of streamflow hydrology this might occur when simulating rainfall-runoff relationships without observations of evapotranspiration or percolation to the aquifer. In this case, we would expect some amount of the total precipitation to be 'lost' with respect to observed streamflow. This can be accounted for by not summing certain output fluxes in the target variable, which allows the model to learn to directly estimate the net total of unobserved fluxes from the overall mass balance.

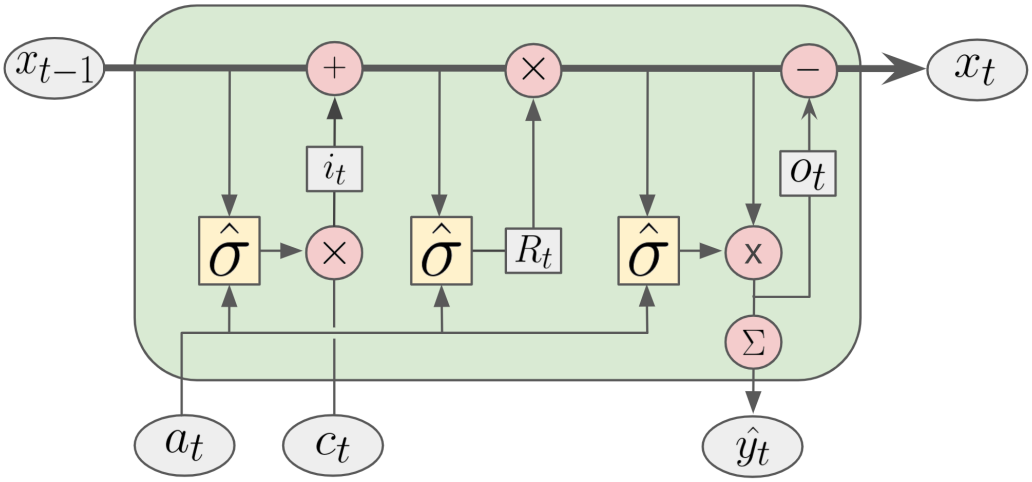

Figure 1: A single timestep of the MC-LSTM. Notation is described in the text.

The model is formally described as follows:

$$\vec{x}_t = (R_t \cdot \vec{x}_{t-1}) + \vec{i}_t - \vec{o}_t \tag{4}$$

$$R_t = \hat{\sigma}\left(W_r \cdot (\vec{a}_t, \vec{x}_{t-1}) + B_r\right) \tag{5}$$

$$\vec{i}_t = \hat{\sigma}\left(W_i \cdot (\vec{a}_t, \vec{x}_{t-1}) + \vec{b}_i\right) \times c_t \tag{6}$$

$$\vec{o}_t = \hat{\sigma}\left(W_o \cdot (\vec{a}_t, \vec{x}_{t-1}) + \vec{b}_o\right) \cdot \vec{x}_t \tag{7}$$

$$\widehat{y_t} = \sum_{i=2}^{N} o_{i,t}, \tag{8}$$

where $R_t, W_* \in \mathbb{R}^{N \times N}$, $a_t, i_t, o_t, x_t \in \mathbb{R}^N$, and W_*, $\vec{b}_*$, and B_* are learned weights and biases. The sum over the output vector (Eqn. 8) allows multiple states to contribute to a single, aggregate boundary flux, which in our case is streamflow. This sum starts at $i = 2$ to account for unobserved sinks.

For ease of reference we will refer to this model as a *mass conserving long short term memory network* (MC-LSTM). However, it is important to note that any type of physical or conceptual quantity can be conserved: e.g., mass, energy, momentum, counts, etc.

3 Benchmarking Experiment

3.1 Methods

The model described in Section 2 was applied to the same benchmarking experiments used by [9]. Specifically, we used the model to simulate daily streamflow values at 447 basins from the Catchment Attributes and Meteorological Large Sample (CAMELS) data set [1].

CAMELS includes several daily forcing data sets; we used Maurer forcings to be consistent with [9]. CAMELS also includes several static catchment attributes related to soils, climate, vegetation, topography, and geology [2], and we used the same static attributes as [9] and [10]. All meteorology and catchment features (including precipitation) were used as auxilary inputs, a_t, and were standardized by removing the global (i.e., across all catchments) mean and variance. Precipitation was also used as the conserved input, c_t, and in this capacity was not standardized.

Daily streamflow records from the US Geological Survey were used as training targets with a normalized squared-error loss function that does not depend on basin-specific mean discharge:

$$\text{NSE*} = \frac{1}{B}\sum_{b=1}^{B}\sum_{n=1}^{T}\frac{(\widehat{y}_{b,t} - y_{b,t})^2}{(s_b + \epsilon)^2}, \tag{9}$$

where B is the number of basins, T is the number of time steps per basin, and s_b is the variance of the hydrograph in basin b.

The training period was October 1, 1999 through September 30, 2008 and the test period was October 1, 1988 through September 30, 1999. The model was trained in sequence-to-one mode (to increase minibatch diversity) with 30 epochs, a minibatch size of 256, and look-back of 365 days.

Our implementation of the MC-LSTM described in Section 2 included a standard LSTM, so that only a portion of the states in the model participated in enforcing conservation, and therefore in directly simulating mass outflow (streamflow) from the watershed. Some of the states in our network were acted on by standard LSTM gates (forget, input, and output), and the standard LSTM hidden states were fed into each activation node in the MC-LSTM. This allowed the model to learn aspects of catchment memory that are not directly related to water storage, while simultaneously not corrupting the overall mass balance. The only states that directly contributed to the simulated target variable in the loss function were the conservation states.

We benchmarked against the standard LSTM and also the same conceptual hydrology models used by [9]. The statistics are: (i) Kling-Gupta Efficiency (KGE), (ii) total bias, (iii) the ratio of modeled vs. observed variances (σ_{rat}), (iv) the coefficient of determination (r^2), (v) high-flow bias (FHV), and (vi) low-flow bias (FLV). KGE is often preferred in hydrology over standard squared-error metrics like the RMSE [5], which itself is a function of bias, r^2, and (σ_{rat}). FHV is the bias considering only the top 2% of flows in each basin and FLV is the bias considering only the lowest 30% of flows in each basin. FHV is especially important for flood forecasting and is indicative of extrapolatabiltiy to more extreme events.

3.2 Results

Benchmarking results are in Table 1. The MC-LSTM significantly ($\alpha = 0.05$) out-performed the standard LSTM in terms of KGE, bias, variance, high flows, and low flows, but the LSTM had a higher overall r^2. The inductive bias of the MC-LSTM caused it to out-perform the LSTM in rare (2%) high flow events, which again, are critical for applications like flood forecasting. Additionally, the conservation-constrained model out-performed the standard LSTM in terms of overall bias, and the only models in the benchmark set that had lower overall bias were conceptual models (VIC and HBV), and both had significantly worse overall performance. It is important to note that both the LSTM and MC-LSTM are not as accurate at predicting low-flows as some of the conceptual models. Nonetheless, these 'first-light' results indicate potential for conservation constraints to add value over current state-of-the-art in ML-based hydrological modeling, and especially to provide value in extreme conditions.

As

Table 1: Benchmarking Results. All values represent the median over the 447 basins.

Model	MC?[a]	KGE[b]	Bias[c]	σ_{rat}[d]	r^2	FHV[e]	FLV[f]
Deep Learning Models							
MC-LSTM Ens.	yes	0.764*	-0.020*	0.842	0.873*	-14.689*	-24.651*
LSTM Ens.	no	0.762	-0.034	0.838	0.886	-15.740	36.267
Conceptual Hydrology Models							
SAC-SMA	yes	0.632	-0.066	0.779	0.792	-20.356	37.415
VIC (basin)	yes	0.588	-0.018	0.725	0.760	-28.139	-74.769
VIC (regional)	yes	0.257	-0.074	0.457	0.651	-56.483	18.867
mHM (basin)	yes	0.691	-0.040	0.807	0.832	-18.640	11.433
mHM (regional)	yes	0.468	-0.039	0.589	0.793	-40.178	36.795
HBV (lower)	yes	0.391	-0.023	0.584	0.713	-41.859	23.883
HBV (upper)	yes	0.681	-0.012	0.788	0.833	-18.491	18.341
FUSE (900)	yes	0.668	-0.031	0.796	0.815	-18.935	-10.538
FUSE (902)	yes	0.690	-0.047	0.802	0.821	-19.360	-68.224
FUSE (904)	yes	0.644	-0.067	0.783	0.808	-21.407	-67.602

[a] *Mass conservation (MC).*
[b] *Kling-Gupta Efficiency:* $(-\infty, 1]$*, values closer to one are desirable.*
[c] *Bias:* $(-\infty, \infty)$*, values closer to zero are desirable.*
[d] *Variance Ratio:* $(-\infty, \infty)$*, values closer to one are desirable.*
[e] *Top 2% high flow bias:* $(-\infty, \infty)$*, values closer to zero are desirable.*
[f] *Bottom 30% low flow bias:* $(-\infty, \infty)$*, values closer to zero are desirable.*
[*] *MC-LSTM is significantly different than the LSTM by Wilcoxon rank test at* $\alpha = 0.05$*.*

4 Conclusion

Knowledge-guided machine learning is critical for the Earth sciences due to the empirical successes of deep learning coupled with the fact that there are strong intuitive reasons why process understanding is likely necessary for predicting out-of-sample, such as under accelerating anthropogenic-driven global change [11]. What we would like to see going forward is for KGML models to help test specific process-relevant hypotheses. Right now, simulating complex dynamical systems like watersheds at a process level generally requires explicit representation of all governing processes, and compensating error structures can arise when certain processes are either not represented or misrepresented [13, 4]. KGML frameworks like the one presented here provide an opportunity to test individual process-level hypotheses without the need for a model that contains a large number of auxiliary hypotheses.

References

[1] Nans Addor, AJ Newman, N Mizukami, and MP Clark. Catchment attributes for large-sample studies, boulder, co: Ucar/ncar, 2017.

[2] Nans Addor, Andrew J Newman, Naoki Mizukami, and Martyn P Clark. The camels data set: catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences (HESS)*, 21(10):5293–5313, 2017.

[3] Anonymous. {MC}-{lstm}: Mass-conserving {lstm}. In *Submitted to International Conference on Learning Representations*, 2021. under review.

[4] Nancy Cartwright and Ernan McMullin. How the laws of physics lie, 1984.

[5] Hoshin V Gupta, Harald Kling, Koray K Yilmaz, and Guillermo F Martinez. Decomposition of the mean squared error and nse performance criteria: Implications for improving hydrological modelling. *Journal of hydrology*, 377(1-2):80–91, 2009.

[6] Markus Hrachowitz, HHG Savenije, G Blöschl, JJ McDonnell, M Sivapalan, JW Pomeroy, Berit Arheimer, Theresa Blume, MP Clark, U Ehret, et al. A decade of predictions in ungauged basins (pub)—a review. *Hydrological sciences journal*, 58(6):1198–1255, 2013.

[7] Kuo-lin Hsu, Hoshin Vijai Gupta, and Soroosh Sorooshian. Artificial neural network modeling of the rainfall-runoff process. *Water resources research*, 31(10):2517–2530, 1995.

[8] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331, 2017.

[9] Frederik Kratzert, D Klotz, G Shalev, G Klambauer, S Hochreiter, and G Nearing. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrol. Earth Syst. Sci.*, 23:5089–5110, 2019.

[10] Frederik Kratzert, Daniel Klotz, Mathew Herrnegger, Alden K Sampson, Sepp Hochreiter, and G Nearing. Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 2019.

[11] Grey Nearing, Frederik Kratzert, Alden Keefe Sampson, Craig Pelissier, Daniel Klotz, Jonathan Frame, and Hoshin Gupta. What role does hydrological science play in the age of machine learning? 2020.

[12] Sella Nevo, Vova Anisimov, Gal Elidan, Ran El-Yaniv, Pete Giencke, Yotam Gigi, Avinatan Hassidim, Zach Moshe, Mor Schlesinger, Guy Shalev, et al. Ml for flood forecasting at scale. *arXiv preprint arXiv:1901.09583*, 2019.

[13] Naomi Oreskes, Kristin Shrader-Frechette, and Kenneth Belitz. Verification, validation, and confirmation of numerical models in the earth sciences. *Science*, 263(5147):641–646, 1994.

[14] E Todini. Hydrological catchment modelling: past, present and future. 2007.