Author Response to Review of

How can regime characteristics of catchments help in training of local and regional LSTM-based runoff models?

Reyhaneh Hashemi, Pierre Brigode, Pierre-André Garambois, Pierre Javelle
HESS, doi:10.5194/hess-2021-511

RC: Reviewer Comment AR: Author Response

Reviewer: John Quilty

1. GENERAL COMMENTS

RC: **This paper carefully studies long short-term memory networks (LSTM) for rainfall-runoff prediction, using a large-sample of catchments in France. The key focus is on exploring local and regional models as well as the impact of the 'lookback' period, an important hyper-parameter of LSTM, with respect to predictive performance and physical understanding of the model results. The authors include well-thought out experiments to identify the impact of the lookback period and cases where local and regional LSTM models are best suited. The authors also benchmark LSTM with GR4J, due to its useful ability to capture ground water exchanges with aquifers and/or between catchments.**

The authors spend a considerable amount of effort on tying the performance of LSTM, locally and regionally, to a physical understanding of the results. Some examples include the comparison between local and regional LSTM models with GR4J in terms of a water balance exercise in Section 5.3 as well as the ability of LSTM to predict runoff in controlled catchments at a higher degree of accuracy than GR4J (in Section 5.4). This paper also presents findings (e.g., LSTM does not necessarily outperform simple conceptual rainfall-runoff models) that are counter to other recent studies on LSTM (Gauch et al., 2021; Kratzert et al., 2019; Lees et al., 2021); in all such cases, the authors take the time to carefully describe potential reasons for these differences. Overall, this paper is very strong and I could not find much to criticize. The methodology seems correct. The figures are very nice and easy to interpret and I did not find any of the content, tables, or figures to be superfluous.

I suspect this paper will be very useful to other researchers interested in exploiting the generality of machine learning for hydrological modelling and rainfall-runoff prediction, in particular. I think the paper only needs some very minor corrections and some additional brief explanations (as noted below). Afterwards, the paper could be published.

AR: We would like to thank you for reading our manuscript carefully and with interest and for your encouraging comments. We are very pleased to read that you find our paper useful. We would like to invite you to find our point by point response to your comments in the following sections.

2. SPECIFIC COMMENTS

2.1. Line 32

RC: Does LSTM also help mitigate against exploding gradients? If so, this would be good to mention as well.

AR: Thank you for this relevant question. The answer is, yes. This is because vanishing and exploding gradients both result from the same mathematical challenge when optimizing neural networks (NNs) with very high non linearities, although the latter case (exploding gradients) is less frequent Goodfellow et al. (2016)). A full description on how LSTM overcomes both vanishing and exploding gradients is given in Hochreiter and Schmidhuber (1997). To put it briefly and in very simple words, in deeply nested NNs, such as (vanilla) RNNs when lookback (T) becomes large, it happens that a factor — which in the problematic case is not close to an absolute value of 1 — gets multiplied by itself over and over — T times — due to the chain rule of the calculus. Therefore, the result will either exponentially shrink (if the factor is initially < 1) or exponentially grow (if the factor is initially > 1) and this is where the vanishing or exploding gradient issue arises.

LSTM establishes derivatives that neither vanish nor explode. Contrary to RNNs, LSTM weights are not constant in all time steps and depend on the input of each time step (X_t).

Following the reviewer's suggestion, we propose to include this explanation in the new version. We also intend to rewrite this section to make it more clear and tractable. For instance, we realized that the term "sharing important information between time steps of a time sequence" that we used in Line 151 could be misleading as we did not specify what kind of parameter sharing we were referring to and the reader could confuse it with the "constant weights" in RNNs.

2.2. Figure 1

RC: It would be good to include a description of the acronyms HP and FR in the figure caption (since it is unclear what these acronyms represent).

AR: Thank you for noticing this. Following your suggestion and the point made by the Anonymous Referee #2 on our naming strategy, we plan to revise all instances of acronym/letter based names. Nevertheless, "HP" and "FR" were intended to be acronyms for Hyperparameter and France, respectively.

2.3. Grammatical corrections

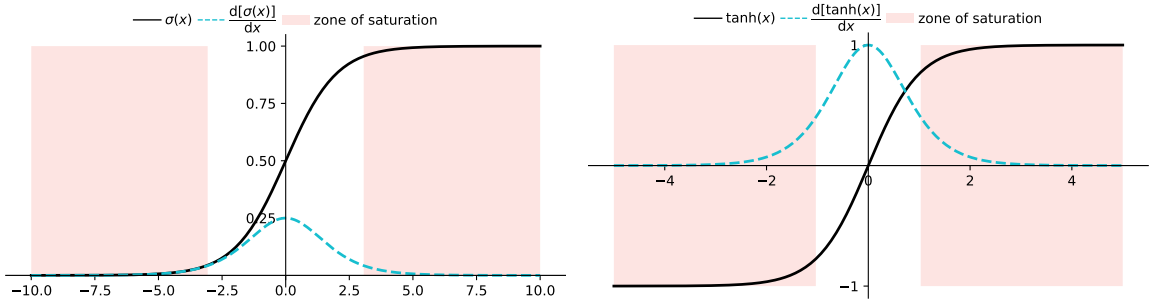
RC: For the most part, the paper is well-written but there are a number of grammatical errors. I stopped correcting such errors around line 154. I recommend that a very carefully read through the paper be completed before re-submission.

AR: Thank you for spotting these grammatical errors. We will proofread the entire paper to fix the mentioned mistakes and to check for any other errors.

2.4. Line 165

RC: The sentence on line 165 can be moved to the last sentence of the same sub-section. A short sentence, 'The main equations used in LSTM are as follows (Ref, XXX):' can be used in it's place.

AR: Thank you for this suggestion. We will update this subsection as proposed.



(a) $\sigma(x)$ vs. $\frac{\mathrm{d}[\sigma(x)]}{\mathrm{d}(x)}$ (b) $\tanh(x)$ vs. $\frac{\mathrm{d}[\tanh(x)]}{\mathrm{d}(x)}$

Figure 1: Sensitivity of the derivative of the Sigmoid and $\tanh$ functions to the range of input data.

2.5. Lines 205-206

RC: Is this sort of standardization the most appropriate for LSTM? Since sigmoid and tanh activation functions are used, should not the data be scaled to [0,1] or [-1,1] as these ranges match the output ranges of the (previously mentioned) activation functions? Perhaps others have adopted the form of standardization adopted here, if so, can the authors indicate this?

AR: Thank you for this interesting question. LeCun et al. (2012) explain that why centering the input data around 0 and scaling them by the standard deviation is typically a good idea and usually makes gradient descent converge faster. Besides, we could not in principle benefit from a [0, 1] scaling since the temperature feature could include negative values.

The interesting point of standardization comes when we investigate how the derivative of different activation functions changes with respect to the range of input data. Please note that here we are talking about the activation function for the hidden layers and not the last layer, which is given by the type of the problem — Softmax for multi-class classification, Sigmoid for binary classification, and Identity for regression.

Looking at Figure 1 of the present document, it turns out that the Sigmoid ($\sigma(x)$) and $\tanh$ functions suffer from a problem — their derivative gets saturated very quickly. By the term "saturation", we mean that their derivative approaches very quickly to zero indicating that weights can not get updated effectively at these points thus the NN can not learn effectively. We observe this problem almost everywhere except in the small region in the middle centered around 0 where the derivative is the most dynamic. Therefore, having the inputs centered around 0 with a variance of 1 would also help fall in the useful area of these functions. Please note that even in their dynamic region the derivatives are small and could bring about the vanishing gradient problem in NNs with high non linearities.

Now, you might ask why not simply using an activation function that does not have vanishing gradients, for instance ReLu? The answer is that the ReLu activation function proved to be typically a more appropriate default choice — if we are allowed to use it. In LSTM, there is a specific reason for which we need to stick with the sigmoid function in gates, despite the mentioned disadvantages. Indeed, the it plays a "gate" role — a function granting us a value between 0 and 1 — and it is not possible to replace it by ReLu or any other activation functions not having this output range.

Kratzert et al. (2018) mentioned that they standardized their data using the mean and standard deviation of their training data and we will indicate this information in the new version.

2.6. Adam algorithm

RC: What were the hyper-parameters (alpha, beta_1, beta_2) set to in the Adam algorithm?

AR: We kept all arguments in the Adam optimization module of the Keras library, including α (learning rate), β_1, and β_2 (L^1 and L^2 norms) at their default values (Chollet et al., 2015):

```
tf.keras.optimizers.Adam(
    learning_rate =0.001,
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-07,
    amsgrad=False,
    name="Adam",
    )
```

2.7. Equation 17

RC: what does epsilon represent?

AR: Thank you for noticing this. We forgot to indicate that Kratzert et al. (2019) added this term (ϵ) to the denominator in the equation of NSE^* so that the loss function would not explode when s was very close to 0 (catchments with very small discharge variance).

3. TECHNICAL CORRECTIONS

RC: Lines 16, 35, 57, 61, 73, 135, 148-150, 154, Figure 6

AR: Thank you for these corrections. We will update the mentioned lines and figure.

REFERENCES

F. Chollet et al. Keras, 2015. URL https://github.com/fchollet/keras.

M. Gauch, J. Mai, and J. Lin. The proper care and feeding of camels: How limited training data affects streamflow prediction. *Environmental Modelling Software*, 135:104926, 2021. ISSN 1364-8152. 10.5194/hess-2021-511https://doi.org/10.1016/j.envsoft.2020.104926. URL https://www.sciencedirect.com/science/article/pii/S136481522030983X.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018. 10.5194/hess-2021-51110.5194/hess-22-6005-2018. URL https://hess.copernicus.org/articles/22/6005/2018/.

F. 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. *Hydrology and Earth System Sciences*, 23(12):5089–5110, 2019. 10.5194/hess-2021-51110.5194/hess-23-5089-2019. URL https://hess.copernicus.org/articles/23/5089/2019/.

Y.-A. LeCun, L. Bottou, G.-B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

T. Lees, M. Buechel, B. Anderson, L. Slater, S. Reece, G. Coxon, and S. J. Dadson. Benchmarking data-driven rainfall–runoff models in great britain: a comparison of long short-term memory (lstm)-based models with four lumped conceptual models. *Hydrology and Earth System Sciences*, 25(10):5517–5534, 2021. 10.5194/hess-2021-51110.5194/hess-25-5517-2021. URL https://hess.copernicus.org/articles/25/5517/2021/.