Author Response to Reviews of

How can we benefit from regime information to make more effective use of LSTM runoff models?

Reyhaneh Hashemi et al.

HESS, doi:10.5194/hess-2021-511

RC: Reviewer Comment AR: Author Response RV: Revision

Dear Editor, Dear Referees,

Thank you for the time you took to review our revised manuscript and for your reports.

We would like to invite you to find our point-by-point response to the reports provided by the referees John Quilty and Anonymous Referee #3 in sections 1 and 2 of the present document, respectively.

Kind regards,
Authors

1. REFEREE: John Quilty

1.1. General comments

RC: **The authors have satisfactorily addressed my comments on their initial manuscript. I recommend the paper be published after correcting the minor issues noted below.**

Thank you for the opportunity to review this interesting paper!

AR: We would like to thank you for reading our revised manuscript and your interest. Please find in the following paragraphs our point-by-point response to your comments or/and the modifications made in the second revision.

1.2. Minor comments

RC: **All comments below refer to the track-changes version of the article.**

1.2.1 Reference for the equations of Section 3.1 — A primer in Long Short Term Memory (LSTM)

RC: **A reference should be included in the sentence just before Equation 4. The source of all equations that were not developed by the authors in this paper should be clearly indicated.**

RV: Following the reviewer's comment, we have have included the following sentence in §3.1 of the revised manuscript: "Equations (4) to (9) given below are all from Goodfellow et al. (2016), with a slightly different notation.".

1.2.2 Choice of standardization (Lines 250-251)

RC: **"Also, since the temperature feature can take negative values, we can not in principle benefit from a [0, 1] scaling." What principle is being referred to? If a variable that takes negative values is scaled to [0, 1], it still (qualitatively) conveys the same information (as it did on the original scale), where the origin is shifted from 0 to some corresponding number in [0, 1].**

AR: Thank you for this relevant comment, you are right. The min–max normalization is still theoretically applicable, although, for the reasons mentioned in the first AR document, it is recommended to use standardization (the Z–score normalization).

RV: We have removed the referenced sentence from the manuscript.

1.2.3 Table 3

RC: **Should the terms 'runoff index' and 'total precipitation index' in the first column be swapped with one another?**

AR: Yes; thank you for spotting this mistake.

RV: We have done the swap in the second revision of the manuscript.

1.2.4 epsilon in Equation 15

RC: What was epsilon set to?

AR: Following Kratzert et al. (2019), we set epsilon (ϵ) in Equation 15 to 0.1.

RV: We have updated the paragraph just after Equation 15 as follows: “Following Kratzert et al. (2019), ϵ (= 0.1) is added to the denominator in equation NSE* [...]”.

1.2.5 Grammatical issues

RC: I noticed a few grammatical issues related to the authors’ replies to my various comments from the initial review. I suspect there may be others. Again, I suggest the authors do another run through the manuscript to catch similar issues.

a. L210: ‘brief’ instead of ‘quick’.

b. L225: ‘disregard’ instead of ‘throw away’.

c. L233: remove ‘of’.

AR: Thank you for this suggestion. The new revised manuscript has been read and edited thoroughly and all issues related to style and register, or grammar, have been addressed.

2. REFEREE: Anonymous Referee #3

2.1. Review summary

RC: **In this paper entitled "How can we benefit from regime information to make use of LSTM runoff models more effectively?", Hashemi et al. developed long short term memory (LSTM) models to assess their capability for runoff modeling according to how long memory (lookback hyperparameter) depends on hydrological regimes (i.e. on the information existing up to annual time scale), how the models are trained (local, regional or "national"-scale training), and in the end, answer the question "what is the most effective way of using LSTM for making runoff predictions?" (quite a broad question).**

This paper, which has undergone a number of modifications by the authors already, is overall very well written and organized, with clear objectives. This type of paper certainly deserves being brought to the hydrological community. I have a few concerns, though, that I think should be addressed before the paper be considered for final publication. They meet, to some extent, those already expressed previously by one reviewer. The authors will decide whether they can just use these comments below to modify the text or if additional trials are needed.

AR: We would like to thank you for your review as well as your comments and suggestions. Please find below our point-by-point response to your individual comments.

2.2. Main comment

RC: **It would have been probably better to explore a little deeper the parameter space in my opinion (as emphasized by reviewer 2 previously). At least, should the paper be published, it is mandatory to explain why some important parameters were kept constant and what is the rationale behind this decision: otherwise, my feeling is it will not be of sufficient help to the readership and potential users to use this work as a support to develop their own models, for instance.**

I am not saying the values of the parameters are not suited, but without any *strong rationale* (physical or anything else) supporting this choice, it is difficult, in the framework of ML/DL approaches, to justify the selection of just a few values of a limited number of hyperparameters.

AR: Thank you for this relevant comment. In the following paragraphs, we explain for each hyperparameter individually why it has been tuned (or not) and why some certain exploring values (and not others) have been taken into account.

2.2.1 Learning rate

In the present paper, Adam is used as the optimization algorithm and its base learning rate is fixed to 10^{-4}.

Why did not we tune its (base) learning rate? Adam (Kingma and Ba, 2014) is from the family of algorithms with *adaptive* learning rates and is considered to be a robust algorithm with respect to the choice of its hyperparameters, including its base learning rate (Goodfellow et al., 2016).

Why did we fix it to 10^{-4}? A suitable learning rate value would give an asymptotic converging (optimization) learning curve and would not overshoot effective local minima (Bengio, 2012). Given these factors, we fixed Adam's basic learning rate to 10^{-4} and performed a post hoc examination of the (optimization) learning curves for the different models in the experiments we conducted that did not reveal any divergence of the training criteria due to a too high learning rate. The rate of 10^{-4}, which is 10 times lower than Adam's default

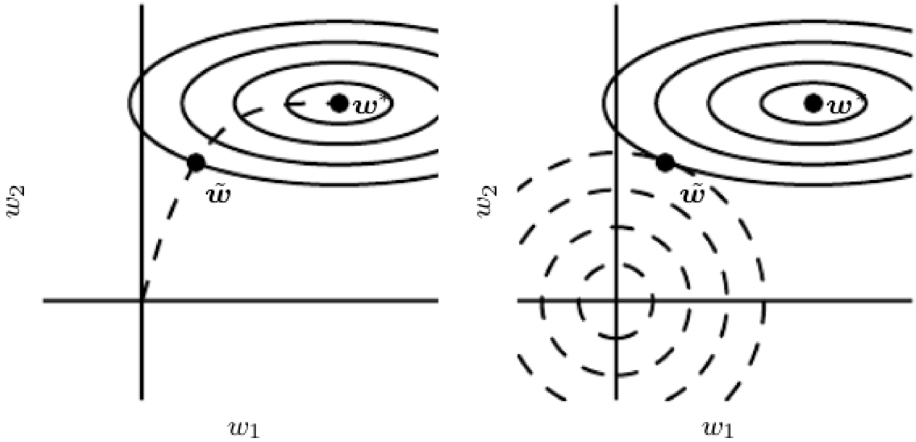

Figure 1: [*Both figure and caption are from the textbook of Goodfellow et al. (2016).*] An illustration of the effect of early stopping. (Left) The solid contour lines indicate the contours of the negative log-likelihood. The dashed line indicates the trajectory taken by SGD beginning from the origin. Rather than stopping at the point ω^* that minimizes the cost, early stopping results in the trajectory stopping at an earlier point $\tilde{\omega}$.(Right) An illustration of the effect of L^2 regularization for comparison. The dashed circles indicate the contours of the L^2 penalty, which causes the minimum of the total cost to lie nearer the origin than the minimum of the unregularized cost.

base learning rate (in Keras), was selected to provide better steps with respect to local minima. Given this lower chosen learning rate, in order to ensure that full regime training had been provided and that the training criterion had sufficient time to decay, we did not impose a predetermined number of epochs, instead allowing the LSTM to continue to learn for as long as its performances improved in the validation data. Further, 10^{-4} is the chosen value in similar previous studies (Kratzert et al., 2018; Lees et al., 2021).

RV: We have added a discussion including the elements provided in this answer to §3.4 of the revised manuscript.

2.3. Dropout rate

AR: *Why did not we consider a larger variation for dropout?* The early stopping algorithm implemented in the paper already acts as a regularizer. Goodfellow et al. (2016) show how, in the case of a simple linear model with a quadratic error function and simple gradient descent early stopping is equivalent to L^2 regularization (see Figure 1 taken from Goodfellow et al. (2016)). Our results showed that the use of a second regularization strategy (dropout rates > 0) in conjunction with early stopping would not further improve performance (compared to the use of early stopping alone, i.e., dropout rate $= 0$). This is consistent with the results provided in previous studies by Kratzert et al. (2019) and Lees et al. (2021), where no other regularization (e.g., early stopping) is implemented and dropout rates > 0 give better results than dropout rate $= 0$. Given this, there was no point testing more dropout variations in our paper.

RV: We have added a discussion including the elements provided in this answer to §3.4 of the revised manuscript.

2.3.1 Batch size

RC: For instance, it is not clear why batch size was kept to 128.

AR: This is a very interesting question; thank you.

Why did not we tune batch size? Bengio (2012) notes that the impact of the size of training batches (B) is mostly computational and that, theoretically, B should mainly impact training times and convergence speeds, with no significant impact on test performance. That is, larger Bs would speed up computation but need to encounter more samples in order to arrive at the same error since there are fewer updates per epoch, and vice versa for smaller Bs.

Why did we fix it to 128*?* Typical recommended batch sizes are powers of 2 (since they offer a better GPU runtime), ranging from 32 to 256 (Goodfellow et al., 2016). Very small batch sizes might require a lower learning rate to maintain stability due to the high variance in gradient estimates. Thus, the total runtime can increase significantly when more steps are required to 1) visit the entire sample, and 2) converge (because a lower learning rate is used). Our chosen learning rate—batch size (10^{-4}, 128) – gave a reasonable run time and adequate convergence and test performance.

RV: We have added a discussion including the elements provided in this answer to §3.4 of the revised manuscript.

2.3.2 Hidden unit (HU) size

RC: Or why only 64, 128, 256 hidden units were eventually selected: not less, nothing in between? By the way, is there any specific reason for choosing log2 values? I don't think any numerical constraints would require this in the present context and gaps between successive values are large...

AR: *Why* $\log 2$ *values? Why nothing in between?* Bengio (2012) offers an interesting discussion on the recommended exploration values for a hyperparameter the "Scale of values considered" paragraph of §3.3 of his paper. He explains that, instead of making a linear selection of intermediate value intervals (the values between the lower and upper bands, here 64 to 256), it is often much more useful to consider a linear or uniform sampling in the log domain — in the space of the logarithm of the hyperparameter. This is because the "ratio" between different values is often more important than their absolute difference when it comes to "the expected impact of the change". For this reason, Bengio (2012) states that exploring uniformly or regularly spaced values in the space of the logarithm of the numerical hyperparameter is typically to be preferred for positive valued numerical hyperparameters.

Why not less? Should the optimal HU be lower than 64, using a HU of 64 would not negatively impact generalization performance, it would simply require proportionally greater computation (Bengio, 2012).

RV: We have added a discussion including the elements provided in this answer to §3.4 of the revised manuscript.

2.3.3 Sequence lengths greater than 2 years

RC: Also, I am wondering if it would have been interesting to use sequence lengths (lookback) up to, say, 4 years: I have not seen what the streamflow time series look like but for some of them with strong baseflow and high multi-annual variability (as visible in some regimes of fig.4), it might be possible that some useful information be still present further back in time (even more than 2 years), and that

the annual scale ("regime") does not necessarily contains all the useful information by itself (there have been quite an amount of works published in the past decade on the topic). Without that, it will be probably difficult to provide a meaningful answer to question Q4 about "[...] the most effective way of using LSTM for making runoff predictions", in my opinion...

AR: Thank you for this relevant and interesting comment. We fully agree with you on this point and also believe that the sweet spot [1] for lookback could go beyond 2 [years], in particular, in Uniform catchments. Our paper's focus has just been to show that the (minimum) required lookback would vary depending on the catchment hydrologic characteristics. Question "Up to how many years?" is not addressed in the present paper, albeit very interesting.

Our goal to address Q4 has been to emphasis on the approach involved — group training + *local hyperparameter selection*. In that, we would like to note that local parameter search could be performed in a space of whatever size.

2.4. Minor suggestions

2.4.1 Hysteresis versus memory length (Introduction Section — Line 52)

RC: **I think that confusing the hysteretic behavior and the memory length of a catchment is not strictly speaking true: the first relates mainly to the lagged response to the input, the second to the time taken by the system to dissipate the information of the input.**

AR: Thank you for this comment.

RV: To prevent confusion, "hysteresis" and all of its derivatives have been removed throughout the manuscript. Instead, the "long term dynamics" or "temporal dynamics" terms have been used, depending on the context.

2.4.2 Ground — unsuitable collocation of "aquifers" (Line 54)

RC: **Remove "ground" (!?) and just keep "aquifers".**

RV: Yes; thank you for the suggestion. We have removed the word "ground" and used the term "aquifers" alone.

2.4.3 Choice of standardization (Section 3.2)

RC: **I understand the arguments supporting the choice of classical standardization instead of the usual minmax scaling. Yet, it would be interesting to indicate whether the two types of scaling were tested or not (from the text it seems that no trial was made using minmax scaling but it should be indicated).**

AR: No; we did not test the min max normalization.

RV: We have added the following sentence to the end of §3.2 of the (re-)revised manuscript: " We should, however, note that we have not tested other forms of normalization, for example, the min–max normalization ([0, 1] scaling), and have not investigated their influence on LSTM performance."

[1] term borrowed from Bengio (2012)

2.4.4 Caption of Figure 9

RC: It seems to contradict the legend at the top of the figure (which says, for instance, "solid=mean" while the caption says "solid=training").

AR: Thank you for noticing this mistake. The legend is explaining the correct correspondence.

RV: We have modified the caption of Figure 9 as follows: “[...] In each panel, the dashed and dotted lines correspond respectively to the training and validation data. The solid line is the mean of the training and validation lines. [...]”

REFERENCES

Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

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

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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.

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.

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.