https://upright.pub/index.php/ijrsps/

Original Contribution

A Single Long Short-Term Memory Network can Predict Rainfall-Runoff at Multiple Timescales

Mani Manavalan[1], Naresh Babu Bynagari[2]

Keywords: LSTM, Memory Network, Rainfall-Runoff, Timescales

International Journal of Reciprocal Symmetry and Theoretical Physics

Vol. 2, Issue 1, 2015 [Pages 1-7]

Long Short-Term Memory Networks, otherwise known as LSTMs, have not been left out when it comes to applying them to daily discharge forecasts rather successfully. A good number of experimental cases, be as it may, need forecasts in a manner with a more granular time frame. Case in point, the correct forecast of brief but intense flooding apexes can bring about a difference with the capacity of saving lives in mass. Still, such climaxes have the capability of escaping the rough non-permanent resolve of daily forecasts. Nevertheless, when an LSTM data is naively learned on an hourly data basis, it entails a time-consuming process with lots of stages, which makes the training complex and computationally-cum financially costly. With this research, we suggest a pair of Multi-Time Scale LSTM or MTS-LSTM frameworks that collaboratively forecast a multiplicity of timescales inside a single model. This is done as they proceed with long-past investments in one non-permanent resolve and diversify into every timescale in order to arrive at more current input stages. For this, we carry out a test on these models on a total of 516 basins through the continental United States and standard in comparison with the United States National Water Model. Juxtaposed with naive forecasts that have distinctive LSTM for each time scale, multi-timescale designs will be computationally the more efficient party, suffering no loss of correctness. Outside the quality of predictions, the multiple-facing timescale has the capacity to process a variety of input variables at various timescales. That, in question, proves quite relevant when it comes to operational applications in which meteorological forcings' lead time is contingent upon their non-permanent resolutions.

INTRODUCTION

The modelling methods for rainfall-runoff that are leaned on deep learning—especially the LSTMS networks—have been proven to be efficient in numerous studies. Long Short-Term Memory networks have the capacity to forecast a multiplicity of catchments with the aid of one model and still bring out more correct predictions when compared to cutting-edge procedure-reliant methods in an array of standards. The various methods need hydrology-related data at different timescales. Case in point, operators at the hydropower level may be attentive to day-by-day or weekly—and possibly even longer—inputs for their reserve bases. Meanwhile, flood prediction is something that needs sub-daily forecasts.

Nevertheless, the bigger bulk of the work associated with the application of the deep learning model in order to flow the prediction stream has mostly be conducted at the timescale level. Day-

[1]Technical Project Manager, Larsen & Toubro Infotech (LTI), Mumbai, **INDIA**
[2]Andriod Developer, Keypixel Software Solutions, 777 Washington rd Parlin NJ 08859, Middlesex, **USA**

by-day predictions are ideal for mid-range to longer-range determinations. On the contrary, day-day-day input resolution can mute diurnal variations with the ability to influence the final signatures with some variations. That includes evapo-transpiration as well as snow melting. Therefore, day-to-day forecasts are more often than not rather too coarse for it to supply actionable data for short-term predictions. Case in point, when there is heavy flooding, the disparity between balancing out the outcomes spread throughout the day and a similar volume of water packed into a-few-hour flashflood can bring about a difference that is life-threatening.

Due to this situation, the hydraulic approaches that can be applied usually operate at a multiplicity of timescales with the help of many non-dependent settings of a conventional, procedure-reliant rainfall-runoff method. In an example, the National Water Model (NWM) of the United States National Oceanic and Atmospheric Administration (NOAA) generates short-range predictions on an hourly basis. It also produces medium-to-long-range predictions on a three-to-six-hour basis every six hours. With this approach, the demand of computational resources will be multiplied, and, as such, can culminate in non-consistent forecasts at any point where a pair of setups converge in their prognostic timescales. The challenge that is the multiplicity of input as well as output time ranges is a popularly known one in the machine learning universe.

REVIEW OF RELATED LITERATURE

The flexibility of re-occurring neural designs structure-wise, creates an avenue for the approaches that together carry out the processing of various timescales, hierarchically. The ways for one to divide and conquer the lengthy processes via stratified processing goes back to tens of years ago (Schmidhuber (1991); Mozer (1991)). In recent years, a clockwork structure was proposed (Koutnik et al., 2014) to partition a repeating cycle of neural systems according to layers with separate time speeds. In the scenario, every layer is updated paying attention to individual frequency. By doing so, the layers that have lower frequency will enable the system to train dependencies of the longer-term family. Even when a hierarchical method like this is applied, the neurons with high frequency will mandatorily process the series for the entire time (Manavalan & Ganapathy, 2014). That, ultimately, makes the training process somewhat slow. An LSTM process was extended to process inputs that were irregularly sampled through a time entrance that attends only to the investment at stages involved training constancy.

This way, the discriminant overlaid input of messages can be assisted. However, the process is more likely to not be suited to the prediction of rainfall-runoff. The reason is that it does not have any means to aggregate the inputs while the time gate is not open. Through a demonstration, a study showed how hierarchy-based processing acts as a catalyst for when LSTMS are translating written expressions and identifying handwritings. Yet, the approach is dependent on a binary move that can be differentiated only by means of a workaround. A good number of these models were tailored to perform tasks such as natural language processing as well as other applications that are not physical (Ahmed & Dey, 2009). In dissimilarity to these assignments, time series for the modelling of rainfall-runoff have frequencies that are regular with fixed interperational policies. Whereas, words in strokes or natural lingua francas in handwriting are not the same length-wise.

An area of application came closer than ever in this regard. The study forecast the speed of the wind considering input information at a variety of timescales. Nonetheless, in semblance to the said language as well as handwriting application scenarios, forecasting a single times series was the goal—whether they are sentences, wind speeds or strokes. The objective we had encompassed a multiplicity of outcomes, one for every timescale in focus. As such, the prediction of multi-timescale rainfall-runoff has resemblance with the optimization with multiple objectives (Donepudi, 2014). For ours, the various goals have close relations with one another since the aggregation of discharge throughout the time stamps ought to be conservatice in nature.

Case in point, every forecast covering a 24-hour period ought to average to one daily step in forecasting. Instead of viewing the challenge from a multi-aim angle, researchers modelled time-flowing controls using the ODE-LSTMs. The ODE-LSTMs are a method known for combining LSTMs with a blend of typical differential equations as well as continuous neural systems. The models that result from this can produce nonstop forecasts with a habitual granularity (Ahmed & Dey, 2010). At the onset, this appears to be an approach with high promises. But, it comes with many setbacks for our application. In the first place, since a single model can produce forecasts for all the focused-on timescales, one will not be able to easily use a variety of forcings for the many target timescales. In the second place, originally, ODE-LSTMs are designed to cater to scenarios in which the input informationes comes at inconsistent intervals.

From our perspective, the opposite is factual. The meteorological forcings possess stationary signals, which makes them regular to a large extent (Maleque et al., 2010). For practical reasons, too, we do not require forecasts with habitual granularity because a fixed group of target timescales proved enough for the cause. Finally, in our exemplary demonstrations, with the ODE-LSTMS to determine time ranges that were unconnected to the learning which turned out worse and significantly slower than the (dis) aggregation of the fixed time scale forecasts for our multiplicity of timescale LSTM. As a result of these, we omitted the ODE-LSTMs from the major evaluation. In this paper, we demonstrate how LSTM-hinged structures can collaboratively forecast outcomes at many timescales with a single model.

With this research, we contribute, firstly, by outlining a pair of LSTM structures that forecast results at a variety of timescales. We attempted to build on the reality that watersheds are damped while the pedigree of entire mass and energy results are critical. As such, the impact of variation at the high frequency level assumes less importance when it comes to lead times of longer durations. Our approach to make availability regarding many output time scales procedures shortens the sequences of the inputs since the inputs with the higher resolutions are only vital for the time steps in the last few stages. We put a benchmark on their daily as well as hourly forecasts against a naive resolution that learns each LSTM for each timescale and a conventional hydrology-related model, which is the United States National Water Model.

Our results demonstrate that every LSTM solution is capable of forecasting at substantially more Nash-Sutcliffe efficiency compared to NWM on all involved time ranges. There is only a small amount of correctness difference between the LSTMs but the naive method proves to have significantly more advantage in comparison to multi-timescale LSTMs (Azad et al., 2011). In the second place, we introduce a scheme for regularization, one that reduces the inconsistency existing throughout the considered timescales as they graduate from the naive stages. In accordance to our findings, the regularization does not only decrease the irregularity but as well culminates in considerably better forecasts generally. Thirdly, we show that LSTMs with many time ranges have the capacity to ingest individual and various groups of forces for each of the considered timescales. Closely, that is similar to operational predicting scenarios where forces with more non-permanent resolutions usually have briefer lead times compared with forces possessing low resolutions (Manavalan, 2014).

RESEARCH METHODOLOGY

To maintain a certain degree of comparability as well as continuity, we carried out our research in a way that is as comparable as possible with preceeding benchmarking papers on the CAMELs dataset. Of the total 531 CAMELS basins that were used in previous studies, 516 of them possessed hourly stream measurement data sources from the USGS Water Information System via the Instantaneous Values REST API. With this service, there are historical quantifications provided at different sub-daily resolutions, which often based on every 15 to 60 minutes. We average it to an hourly and daily time step for every basin involved. Because our forcing information and standards model data are reliant of UTC-derived timestamps, we transformed USGS streamflow timestamps into UTC.

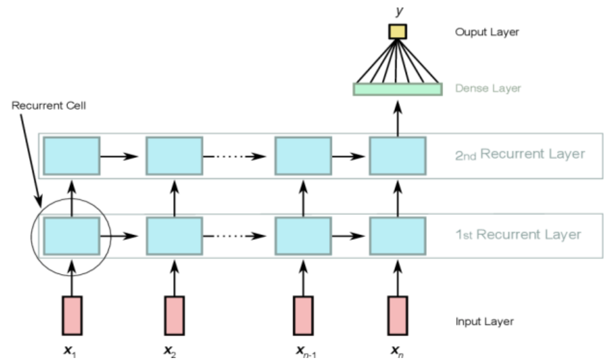
CAMELs are known to provide nothing but day-to-day meteorological forcing information. But, we were in need of hourly forcing for this paper. In order to maintain congruence with the preceding CAMELS case studies, we adopted the hour-by-hour NLDAS-2 product. This comprises meteorological data dating as far back as 1979 (Xia et al., 2012). Then, spatially, we carried out an averaging for the forcing variables for every basin. In addition, we carried out an averaging for basin-specific hour-by-hour weather-related variations for everyday values. There was also a training of our models starting from October 1st 1990 to September 30th, 2003. This timeframe was employed as a period of validation, wherein we evaluated many structures and chose the most ideal model hyperparameters. The entire LSTM models employed in this research take eleven forcing variables. They are concatenated at every step in time with similar 27 rigid catchment characteristics derived from the CAMELS set of data used in another study.

For our experiments, we made use of two sets of designs, serving as baselines with which comparison can be made to the suggested frameworks. A proposed LSTM proved able to adapt naively. To hour-by-hour streamflow modelling and the United States National Water Model as well as the National Oceanic and Atmospheric Administration. The NOAA comes up with streamflow forecasts on an hourly basis concerned with the NWWM. That, in question, is a process-derived model that is based on the WRF-Hydro.

NAIVE LONG SHORT-TERM MEMORY NETWORKS

LSTMs are spinoffs (Hochreiter and Schmidhuber, 1997) from the recurrent neural systems fashioned to carry out modelling for long-term dependency framework that exists between the input data and the output counterparts. Long Short-Term Memory Networks are able to sustain an internal storage state that is refreshed at every interval step by means of a group of activated controls referred to as gates. What are the roles of these gates? They are in control of the relationship between the input and the state—via an input gate. The gates also control the relationship between the state and the output via the output gate. Gates also govern the storage timescales by means of a forget gate.

Long Short-Term Memory networks have the ability to survive with more comprehensive time series compared to classic recurrent systems of neurals. That is because they are not prone to disappearing gradients while the training process is ongoing (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997; Rouf et al., 2014). Considering the reality that LSTMs can process input phases in a sequential order, longer duration series will culminate in longer learning as well as inference durations. This does not present much of a problem for day-to-day forecasts because windows to look back to the last 365 days seems to be the right amount for a good number of basins, mostly in the datasets called CAMELS. This was learned everyday using an input chronology duration of 356 days. In terms of hourly information, surprisingly half a year is equivalent to over 4300 time phases, thus amounting to reasonably long learning and inference time periods.



Additionally, at least to the calculative overhead, the Long Short-Term Memory network forgets the gate, making it difficult to train long-term dependents due to the fact that it reintroduces disappearing gradients into the LSTM effectively. Be as it may, it is not ideal to simply omit the gate for deletion as evident in empirical LSTM studies. Plus, the explorative tests we conducted manifested that this is a result-deteriorating element. In order to address this drawback, a proposal was made to initialize the bias of the gate for forgetting to a minimal value of positivity. This way, the training kicks off with an open gate, naturally after which it will enable the flow of gradient across a higher number of period steps.

Using the bias initialization contraption for all the LSTM models we considered gave us the avenue to add the LSTM with an hourly input rate as the remote hourly baseline for the models we proposed (Ahmed, 2012). The framework of this naive criteria bears resemblance with the day-to-day LSTM, only that we absorbed input chronologies of 4,320 hours, which is equivalent to 180 days. Inclusively, we ranged the training rate and the size of the batches for the naive hour-to-hour LSTM considering the reality that it is in the reception of 24 times more samples compared to the day-to-day LSTM. Due to this critically slow learning, it becomes harder for one to search for hyperparameters.

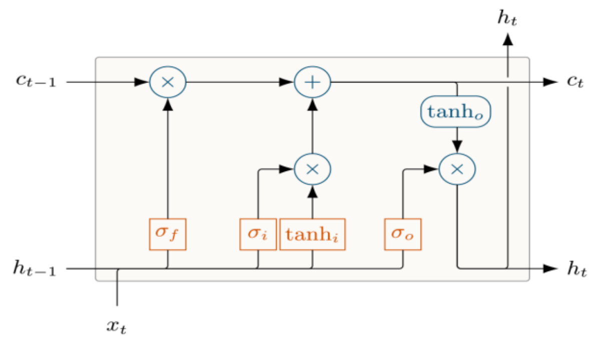
FORECASTING MULTIPLE TIMESCALESS WITH LSTMS

We assessed a pair of LSTM frameworks with the capacity to predict simultaneously at a multiplicity of timescales. For simplicity sake, the succeeding descriptions adopt the instance of a two-timescale design which produces day-to-day and hour-by-hour forecasts. However, the frameworks that we give descriptions to in this study are generalistic to other timescales as well as to over two timescales.

The very first model, called the shared multi-timescale LSTM or the sMTS-LSTM, is a fundamental extension of the naive method. As usual, we produce day-to-day forecasts, wherein the Long Short-Term Memory network absorbs an input chronology of TD time stages at day-to-day resolution and the outcomes is a forecast at the final time stape.

After that, we factory-set the unexposed cell states to their values starting from the time phase. Then, the hourly input process is absorbed in length so it can come up with a sum of 24 hour-by-hour forecasts that are in alignment to the final daily forecast. To express differently, for every forecasting phase, we carry out a pair of forward passes via the same Long Short-Term Memory network. This produces a daily forecast, one which formulates 24 predictions in correspondence to the hour. Then, we include a one-hot timescale kind of encoding to initialize the sequence in a manner through which the LSTM will be able to discern daily inputs from the hourly ones. When it comes to this approach, the key insight is: the hour-by-hour forward communication begins with the LSTM states that start from the day-by-day forward relations. Effectively, the LSTM can access a huge window for looking back. However, in dissimilarity to the hourly LSTM with naive attributes, there is no suffering from the delivery effects of the significantly time-consuming input chronology (Bynagari, 2014).

The second framework, on the other hand, is a variant that is more generalistic to the sMTS-LSTM particularly designed to handle multi-timescale forecasts. As a result, we refer to it as the multi-timescale LSTM or the MTS-LSTM—which functions as the communal version however dividing LSTMs into two separate branches, one going to each of the timescales. First, we produce a forecast with an LSTM serving as the rough timescale with the aid of a complete input chronology of length, known as TD. Secondly, we subject the day-to-day unexposed and cell states to reuse, from the step (TD-TH/24) as the preceding states for a finer timescale LSTM. This, in turn, produces the aligning 24-hour forecasts. Because both LSTM diversifications possibly come in different sizes, the states are fed via a linear state transmission layer—after which they are reused as initial hour-by-hour rates.



For this setting, every LSTM diversification only gets inputs of their corresponding timescales. Due to this, there was no need for us to subject the timescale to one-hot encoding. This framework makes the reason we refer to the other variant as "shared" MTS-LSTM clearer in nature. In effect, the sMTS-LSTM ablates the MTS-LSTM. These two variants share one thing in common, which is the architecture. Nevertheless, their weights are distributed across the entire timescale diversifications and the transfer layer of its state are identity-related operations.

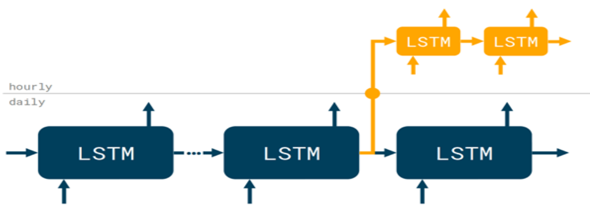
INPUT VARIABLES PER TIMESCALE

One salient upside of the MTS-LSTM in juxtaposition with the sMTS-LSTM framework is derived from the fact that its input dimensionality is flexible in nature. As every time scale goes through processing in an individual LSTM diversification, the various input variables can be absorbed to realize forecasts for many timescales.

In a way, this is a key disparity in operational case studies wherein, case in point, dwells day-to-day weather predictions with substantially longer lead periods compared to the obtainable hour-to-hour forecasts. Or, in a case when one is using local sensing information obtainable only at specific overpass frequencies. For these scenarios, MTS-LSTMs can process daily forcings in its hour-sensitive touchpoint. Hence, the per-timescale forcings approach gives room for many timescales to have use cases for different inputs.

In bid to assess this capability, we employed two groups of forcings as day-to-day inputs. First is the Dayment and Maurer forcing categories partitive to the dataset known as CAMELS (Newman et al., 2014). Due to the insufficiency of forcing samples for other hours, we experimented in two ways: continually conducting just the hourly NLDAS forcings and additionally absorbing the aligning

Daymet and Maurer forcings at each hour of the considered day. The Maurer forcings range just until 2008, so we had this study during the validation timeline from October of 2003 to September of 2008.



Based on the fact that the MTS-LSTM as well as the sMTS-LSTM frameworks have the capacity to simultaneously produce forecasts at a multiplicity of timescale, it is possible for us to incentivize forecasts that align throughout the timescales. Dissimilar to other environments such as computer vision, consistency is something extensively defined in our approach. Moreover, the forecasts are regular should the mean of each day's hourly forecasts be similar to the specific day's daily forecasts. With this, we can state explicitly that the hurdle in terms of a regularization control for loss loss function. The loss function is also able to stabilize them with the mean square disparity among day-to-day and day-averaged hour-to-hour forecasts. Bear in mind that despite the fact that the regularization is described with just two concurrently forecast timescales, the model is generalistic to more timescales. That is because we can plus the mean squared disparity existing between every time scale pair.

OBSERVATIONS & CONCLUSION

The motive of this paper is to create a generalistics application of LSTM-reliant rainfall-runoff modelling to a cornucopia of timescales. The exercise is not as negligible as basically running a variety of deep learning approaches at various timescales as a result of extensive periods available for looking back. It also has to do with associated storage and calculative costs. Using the MTS-LSTM and the sMTS-LSTM, we introduced two LSTM-derived downpour-runoff models using the individual physical attributes of the simulation challenge. From the results, it is evident that the LSTMs have more advantage compared to process-derived approaches—which we refer to as the sMTS-LSTM—has the capacity to process long-term dependents.

This study is a representation of one step in the direction of the development of operational hydrologic approaches spinning off from the deep learning model. In general, it is our belief that MST-LSTMs are more promising for futuristic purposes. It can not only integrate forcings of various non-permanent outcomes, but can also produce accurate and regular forecasts at a multiplicity of timescales. Furthermore, it's calculative overhead during the learning process and the inference period is substantially smaller compared to the specific models for each timescale.

REFERENCES

Ahmed, A. A. A. (2012). Disclosure of Financial Reporting and Firm Structure as a Determinant: A Study on the Listed Companies of DSE. ASA University Review, 6(1), 43-60. https://doi.org/10.5281/zenodo.4008273

Ahmed, A. A. A., & Dey, M. M. (2009). Corporate Attribute and the Extent of Disclosure: A Study of Banking Companies in Bangladesh. Proceedings of the 5th International Management Accounting Conference (IMAC), OCT 19-21, 2009, UKM, Kuala Lumpur, MALAYSIA, Pages: 531-553. https://publons.com/publon/11427801/

Ahmed, A. A. A., & Dey, M. M. (2010). Accounting Disclosure Scenario: An Empirical Study of the Banking Sector of Bangladesh. Accounting and Management Information Systems, 9(4), 581-602. https://doi.org/10.5281/zenodo.4008276

Azad, M. R., Khan, W., & Ahmed, A. A. A. (2011). HR Practices in Banking Sector on Perceived Employee Performance: A Case of Bangladesh. Eastern University Journal, 3(3), 30–39. https://doi.org/10.5281/zenodo.4043334

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult, IEEE Transactions on NeuralNetworks, 5, 157–166, https://doi.org/10.1109/72.279181

Bynagari, N. B. (2014). Integrated Reasoning Engine for Code Clone Detection. *ABC Journal of Advanced Research*, *3*(2), 143-152. https://doi.org/10.18034/abcjar.v3i2.575

Clausen, B. and Biggs, B. J. F. (2000). Flow variables for ecological studies in temperate streams: groupings based on covariance, Journal of Hydrology, 237, 184–197, https://doi.org/10.1016/S0022-1694(00)00306-1

Court, A. (1962). Measures of streamflow timing, Journal of Geophysical Research (1896-1977), 67, 4335–4339, https://doi.org/10.1029/JZ067i011p04335

Donepudi, P. K. (2014). Voice Search Technology: An Overview. Engineering International, 2(2), 91-102. https://doi.org/10.18034/ei.v2i2.502

Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: continual prediction with LSTM, IET Conference Proceedings, pp.850–855.

Gupta, H. V., Kling, H., Yilmaz, K. K., and Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: impli-cations for improving hydrological modelling, Journal of Hydrology, 377, 80–91, https://doi.org/10.1016/j.jhydrol.2009.08.003

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory, Neural Computation, 9, 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735

Maleque, R., Rahman, F., & Ahmed, A. A. A. (2010). Financial Disclosure in Corporate Annual Reports: A Survey of Selected Literature. Journal of the Institute of Bangladesh Studies, Vol. 33, 113-132. https://doi.org/10.5281/zenodo.4008320

Manavalan, M. (2014). Fast Model-based Protein Homology Discovery without Alignment. *Asia Pacific Journal of Energy and Environment*, *1*(2), 169-184. https://doi.org/10.18034/apjee.v1i2.580

Manavalan, M., & Ganapathy, A. (2014). Reinforcement Learning in Robotics. *Engineering International*, *2*(2), 113-124. https://doi.org/10.18034/ei.v2i2.572

Rouf, M. A., Hasan, M. S., & Ahmed, A. A. A. (2014). Financial Reporting Practices in the Textile Manufacturing Sectors of Bangladesh. *ABC Journal of Advanced Research*, *3*(2), 125-136. https://doi.org/10.18034/abcjar.v3i2.38

--0--