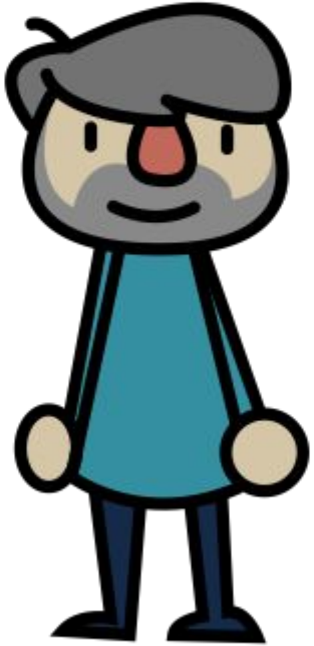
Learning from mistakes: Online updating for deep learning models

a field guide

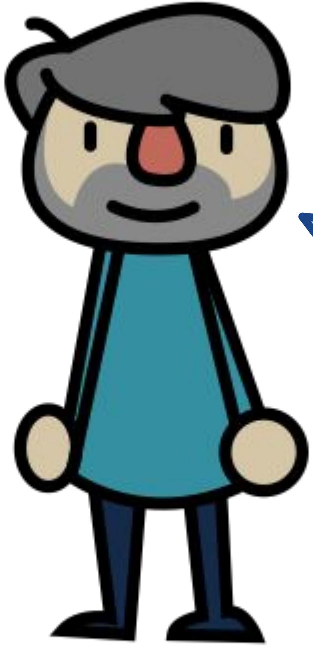
Daniel Klotz, Frederik Kratzert, Alden K. Sampson, Günter Klambauer, Sepp Hochreiter, and Grey Nearing


Hi there!
I am **Daniel**.
Pleasure to meet you.



In the following I will present an an **updating procedure** for differentiable **simulation models**.

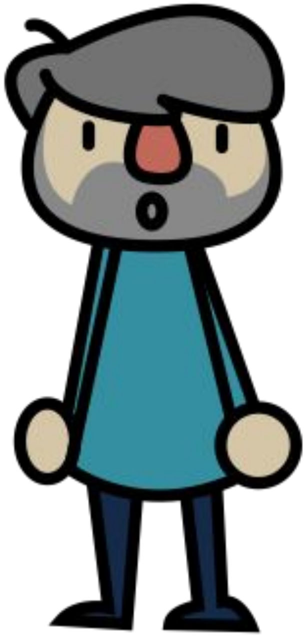
I wanted to present more. But given the circumstances I decided to focus more on a specific idea.



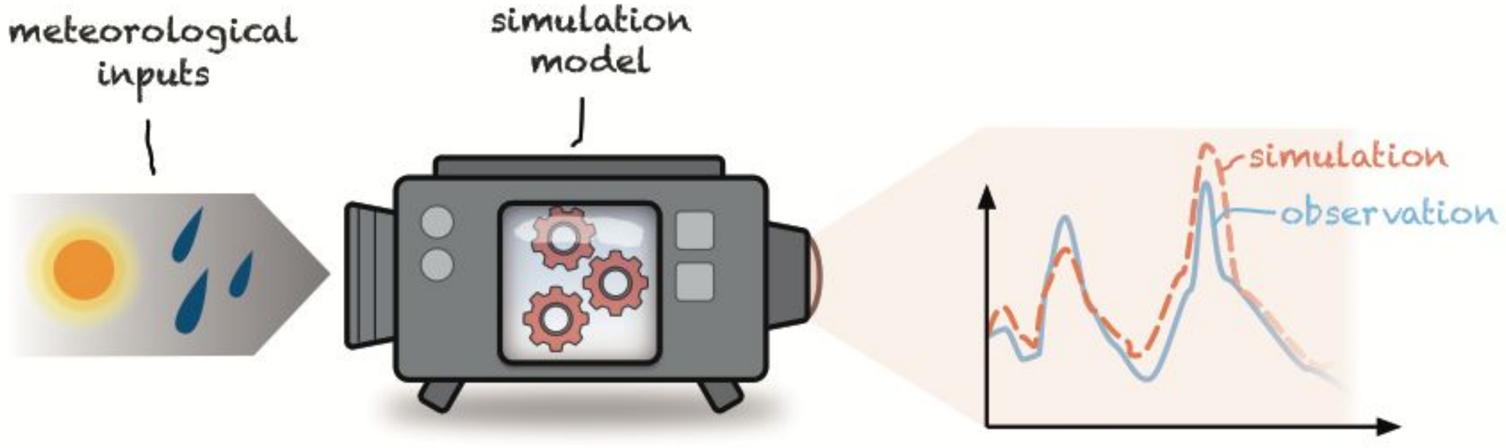
Let me start with a
TLDR for the impatient ones:

We propose to backpropagate the error-signal through the simulation model with respect to the inputs; and use it to minimize said errors to get a new set of updated inputs (and hence model states) which provide better forecasts.

Some **first results** are shown at the end of the **presentation.**

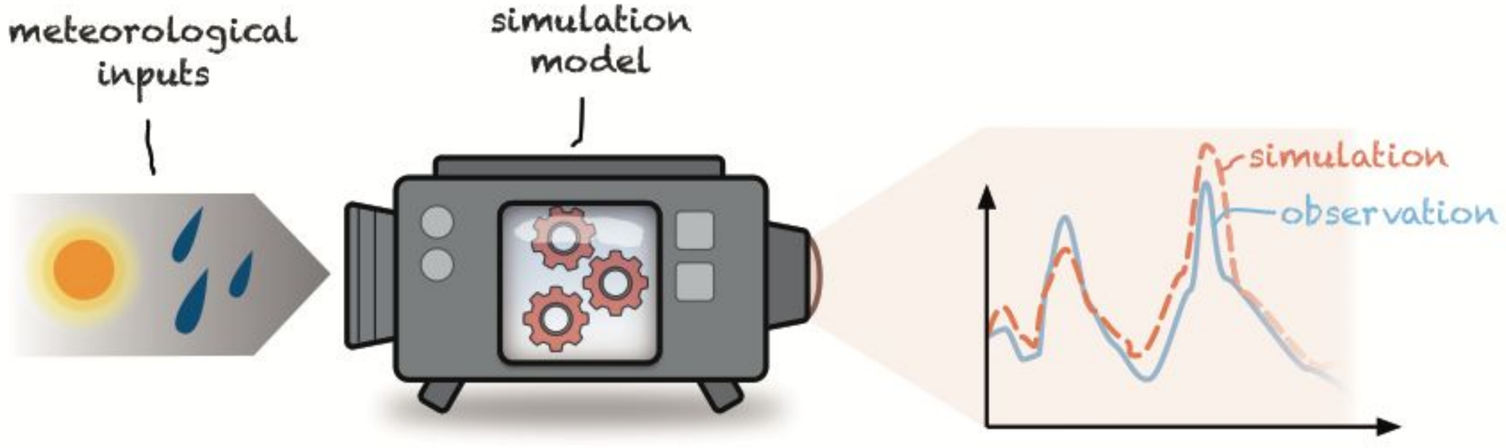


Now, when I talk about **simulation models** I am adapting the definition by Beven and Young (2013).



A **simulation model** uses a set of inputs, that reflect our process understanding, to generate a simulation of a given phenomena.

...

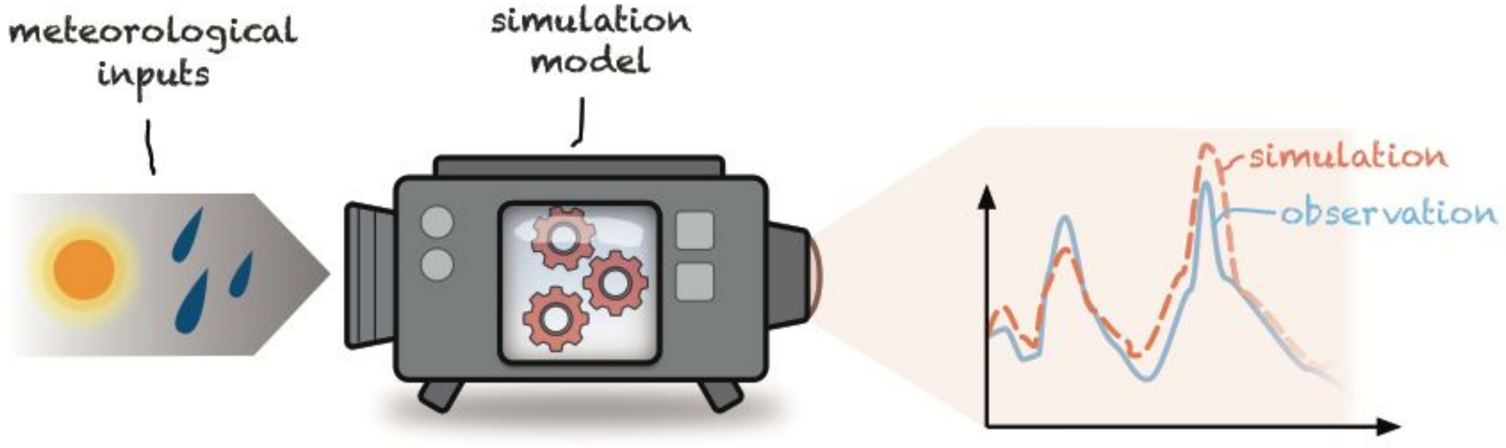


In rainfall-runoff modelling we provide the model with a time series of **meteorological inputs** (such as temperature and precipitation) to simulate the **runoff**.

Most rainfall-runoff models in use today are simulation models.

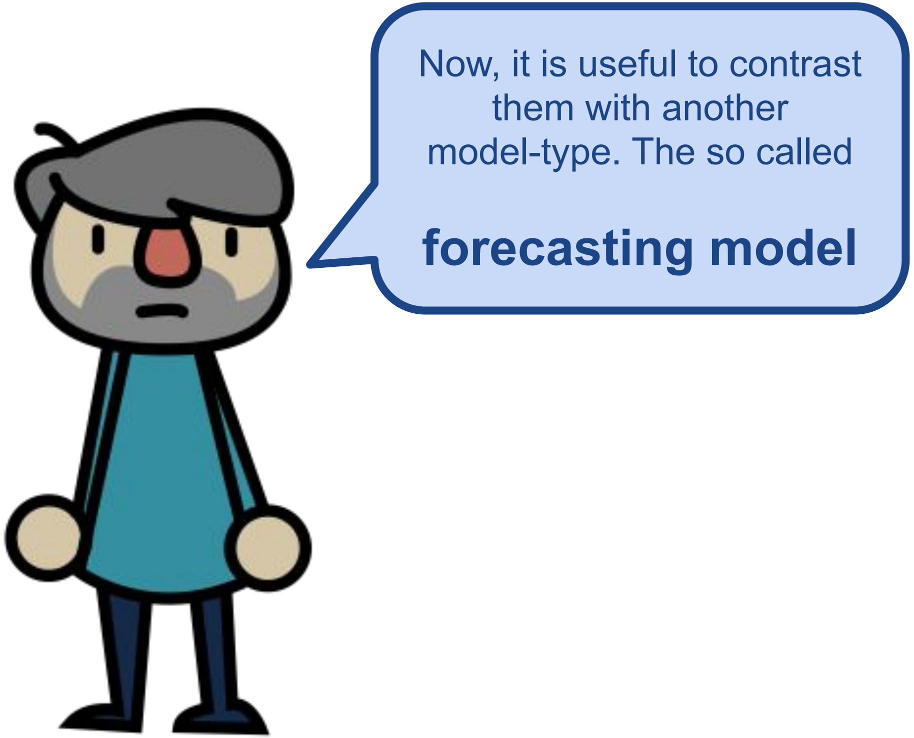
As pointed out by Beven and Young (2013) this assumes that the meteorological inputs are available for any given time steps. When simulation models are used for forecasting purposes this will almost never be the case.

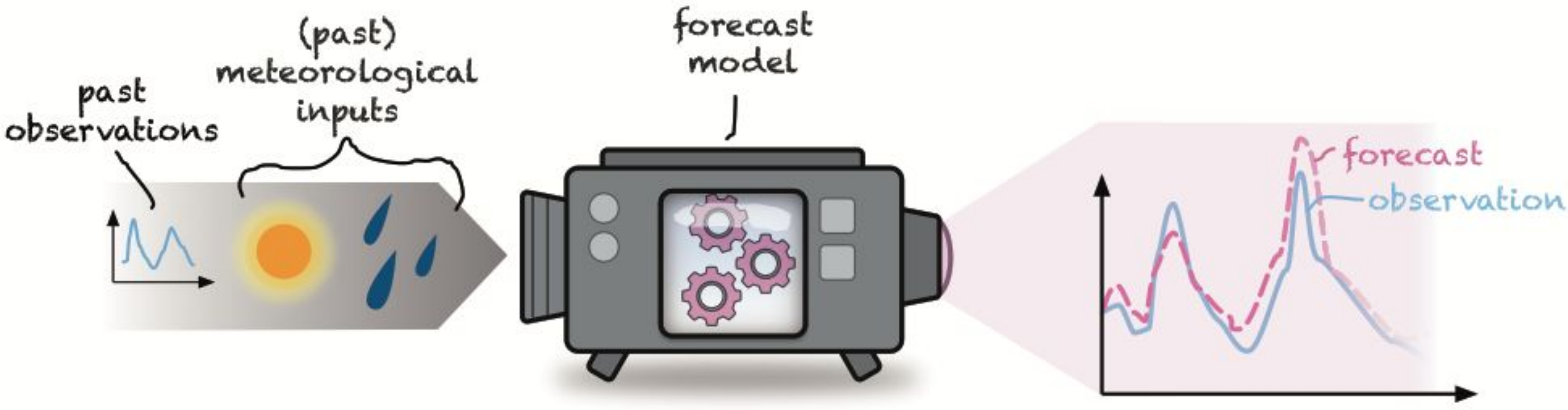
...



Note that we are not assuming a specific representation.

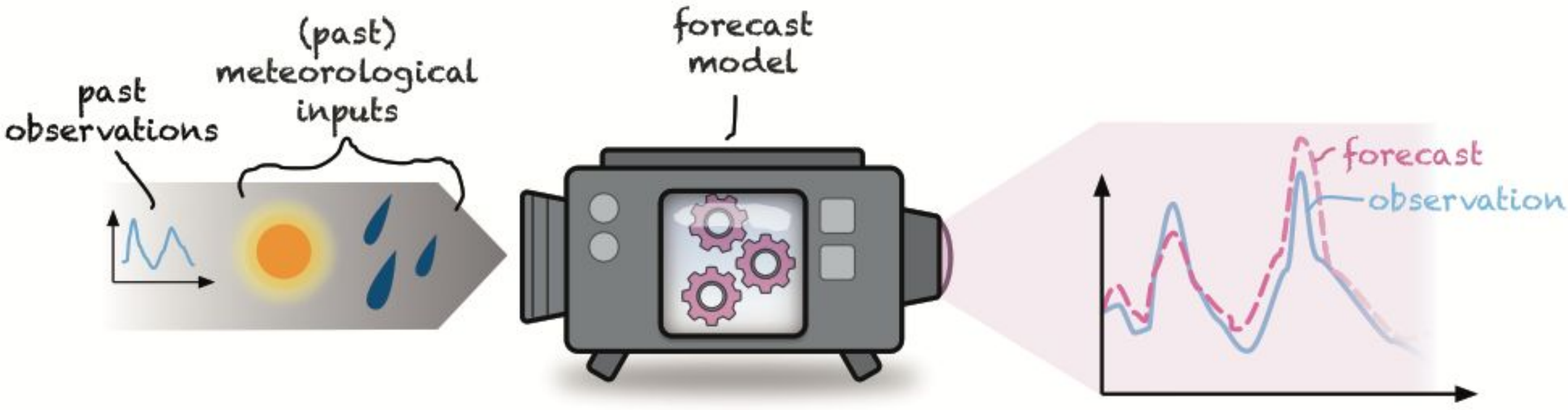
As long as the model is *differentiable almost everywhere* our proposed approach works. We use the updating method with machine learning (ML) models. But, it would also work with physical models, process-based models, or whatever other model type you have.


Now, it is useful to contrast them with another model-type. The so called
forecasting model



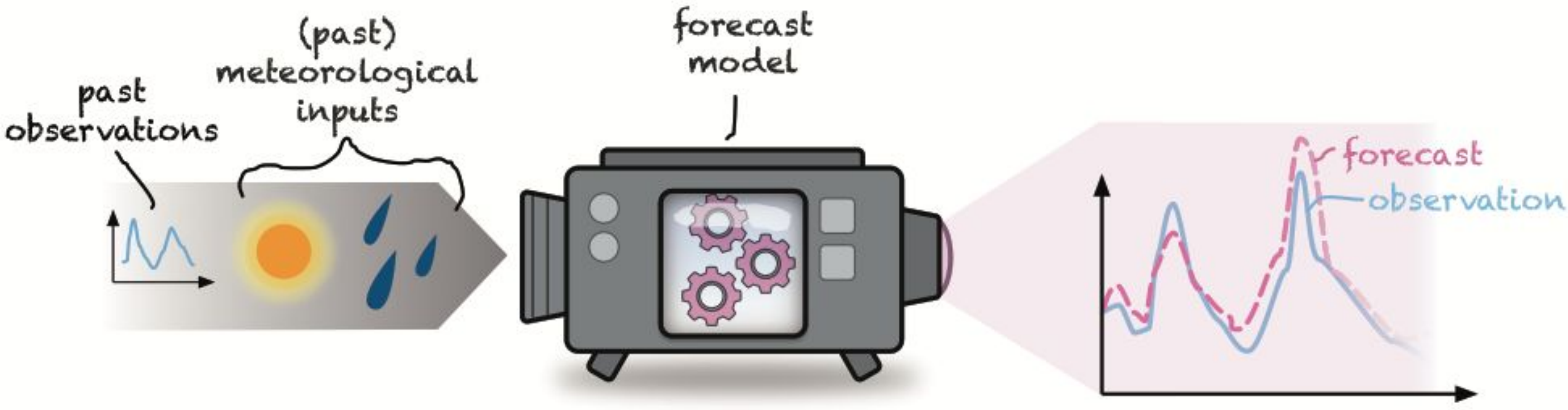
A **forecasting model** uses all available inputs up to a given time-step to generate a forecast for a given time-horizon.

...



For the case of rainfall-runoff forecasting this usually means that the past observation of the model states and outputs - most crucially **runoff** - are also used to make the forecast.

...



Adapting conventional simulation models to a forecasting setting can be challenging.

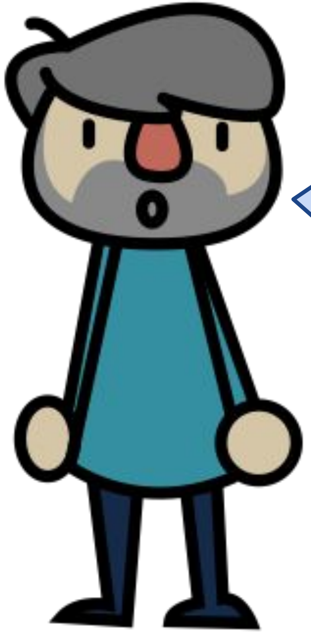
For an ML model, however, one can simply define the past observation as additional inputs and carry on.


me

...


hypothetical
audience

So, which approach is better?



There is no better or worse here.

If the goal is to foster understanding simulation models are often preferred.

If you want to have the best model for a specific task metric you will most likely prefer forecast models.

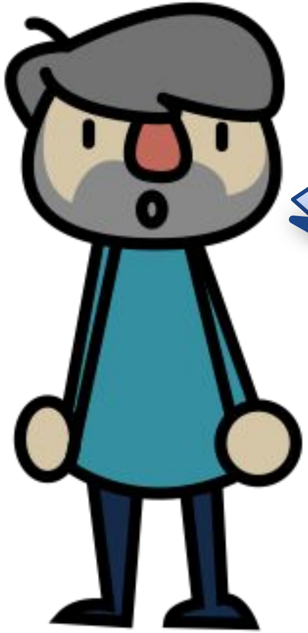
...

So, which approach is better?

There is no better or worse here.

If the goal is to foster understanding simulation models are often preferred.

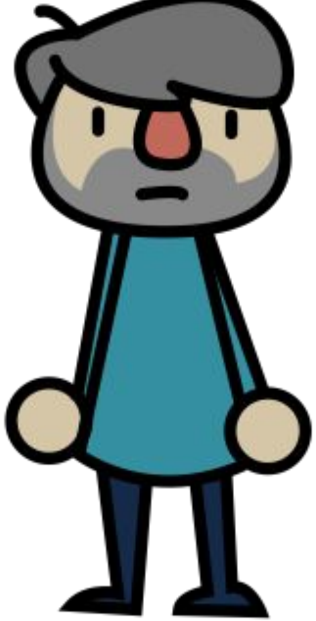
If you want



I once found a nice quote about the these different modelling goals in <u>Hoffman, Minkin & Carpenter (1996)</u>:

“If understanding is sought, simpler models, not necessarily the best and predicting all observables in detail, will have values. Such models may highlight important causes and channels. [...] If predictability is sought at all cost - and realities of marketplace and judgments of the future of humanity may demand this - then simplicity may be irrelevant.”

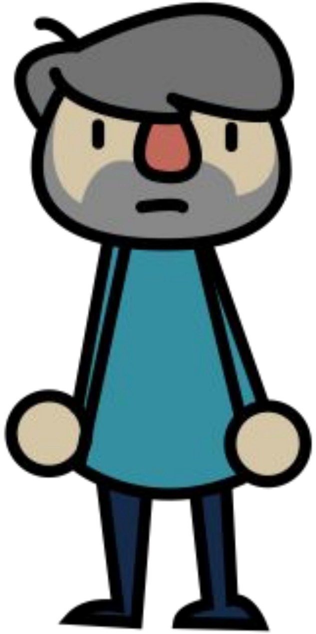
Why then use updating methods for ML approaches at all?



Simulation models represent the underlying hydrological processes. They can be tested more ease (e.g. by relating the model states to hydrological reference-measurements).

...

Why then use updating methods for ML approaches at all?

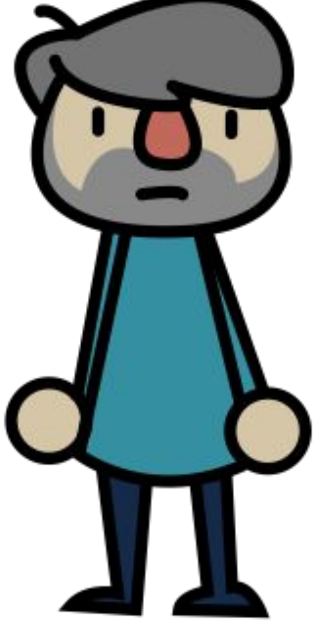


Simulation models represent the underlying hydrological processes. They can be tested more ease (e.g. by relating the model states to hydrological reference-measurements).

Forecasting models can have difficulties when continuous discharge measurement are not available.

...

Why then use updating methods for ML approaches at all?



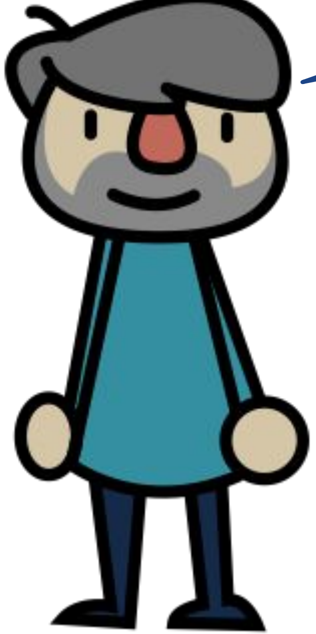
Simulation models represent the underlying hydrological processes. They can be tested more ease (e.g. by relating the model states to hydrological reference-measurements).

Forecasting models can have difficulties when continuous discharge measurement are not available.

And, last but not least, it is also not possible to use them in ungaged ungauged settings.

Oh.. wow. Ok, so how does it work

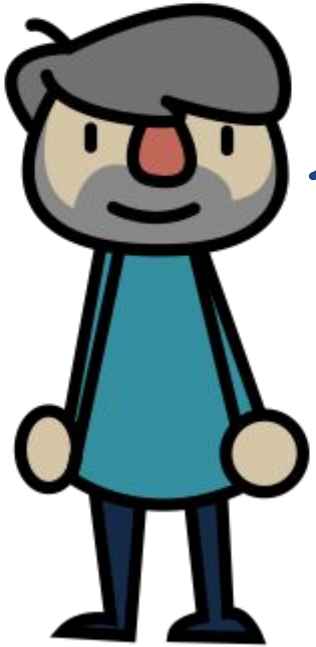
As I said earlier. The core-idea is relatively simple.



...

Oh.. wow. Ok, so how does it work

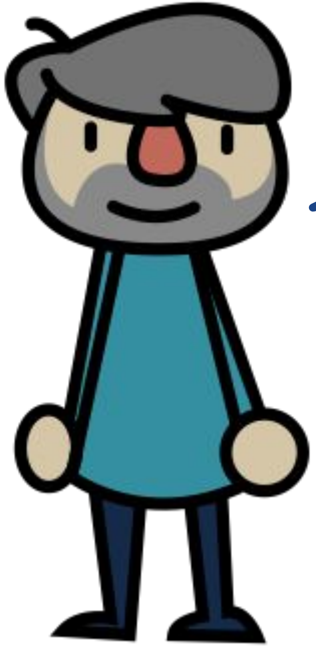
First we choose an updating window. A time-window before a given time-step where we have (runoff) observations. We call this the **updating period**.

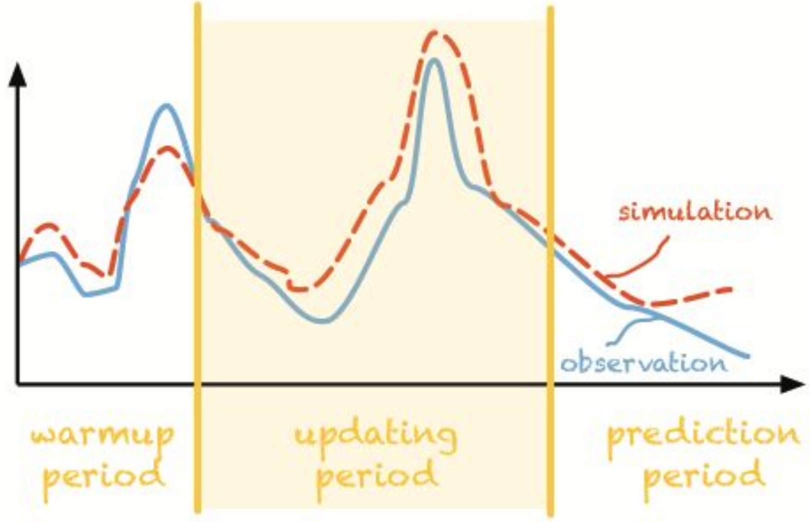


...

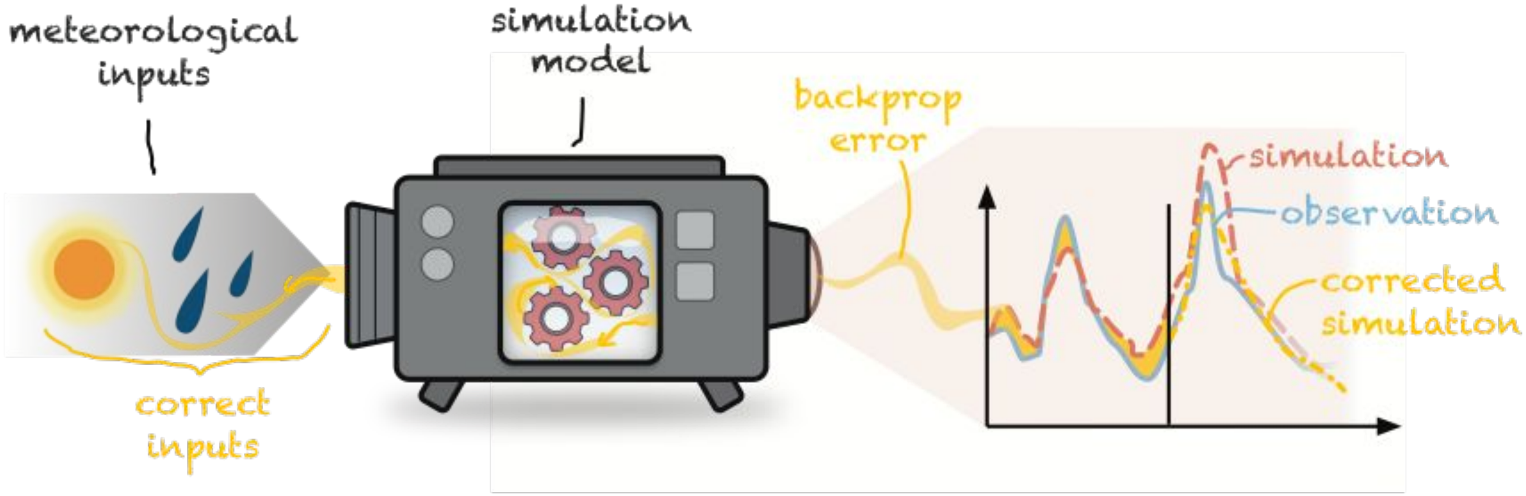
Oh.. wow. Ok, so how does it work

The updating period refers to the timesteps between the **warm-up period** (where the model runs without using the observations) and the **prediction period** (the future timesteps for which predictions are made). I made a sketch to illustrate this:



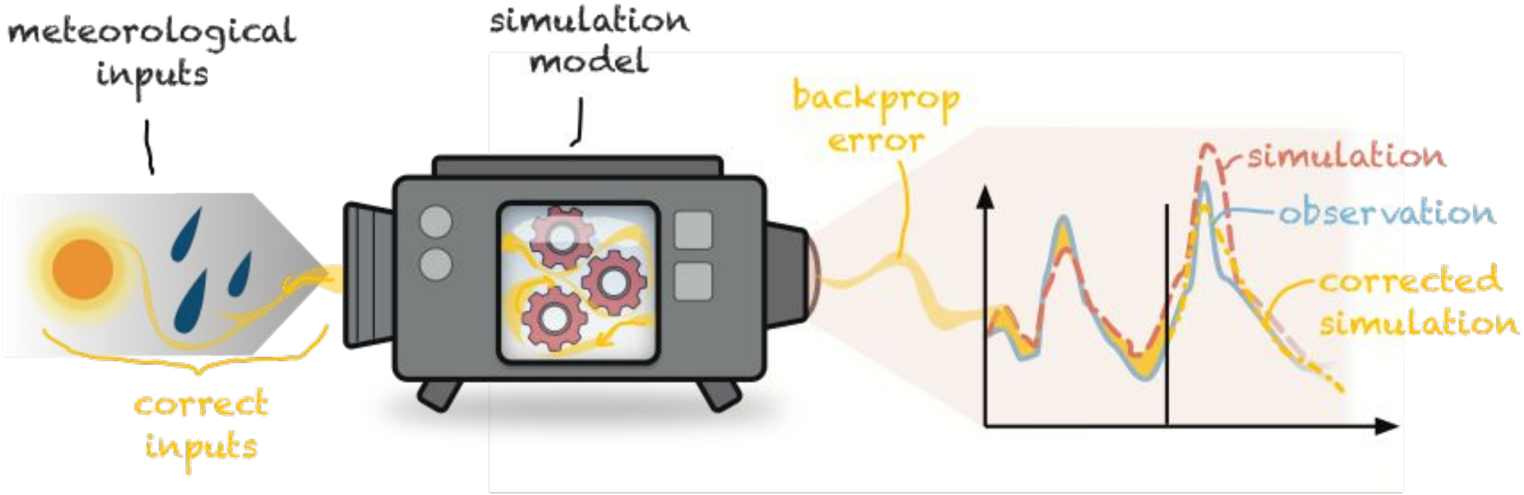


...



Then ...
We **differentiate** (for neural networks this is commonly done via “backpropagation”, see Schmidhuber_2015)
the **error-signal** (for updating period) with respect to the **inputs**,
and use the resulting **gradient** to find **new inputs**.

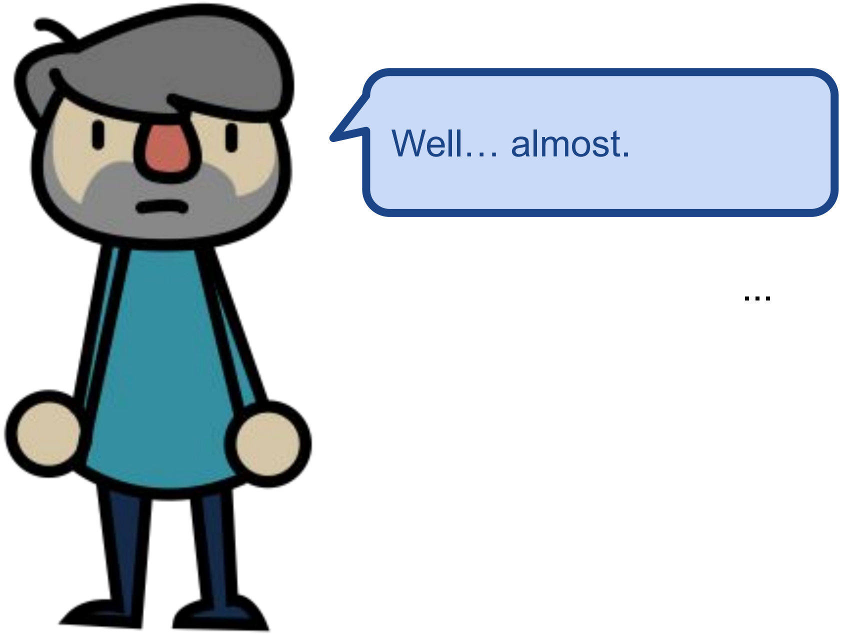
...

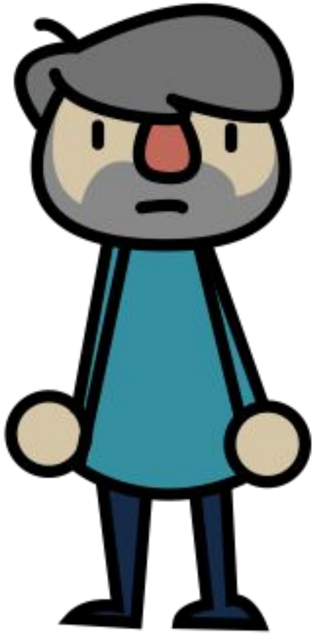


The new or **updated inputs** are fed back to the model to obtain **updated states** and a **corrected simulation**.


That’s it!

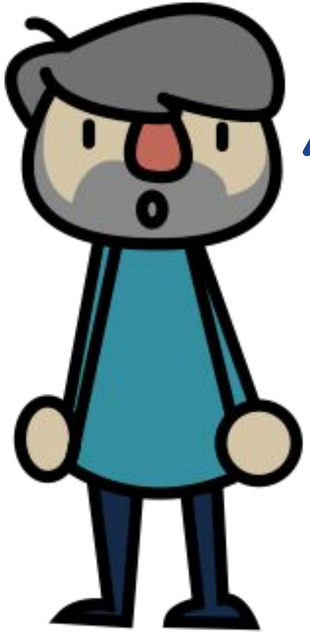
...


Well… almost.
...



To be honest I simplified somethings for the sake of clarity.

...



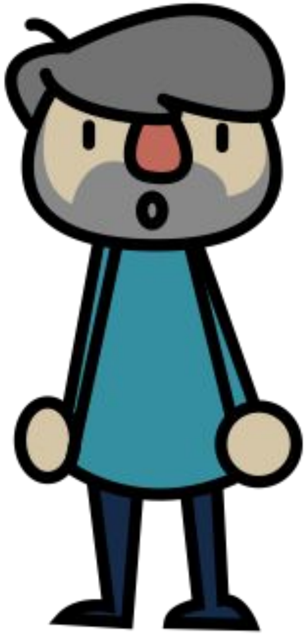
In order to apply our proposed approach we still had to figure out **important parameters**.

Namely:

- Choosing a good updating window,
- weighting between the original observations and the updated ones, and
- making sure that the updated inputs are in sensible ranges

How did we do that you ask?

No! I don't care. Just show me some results!



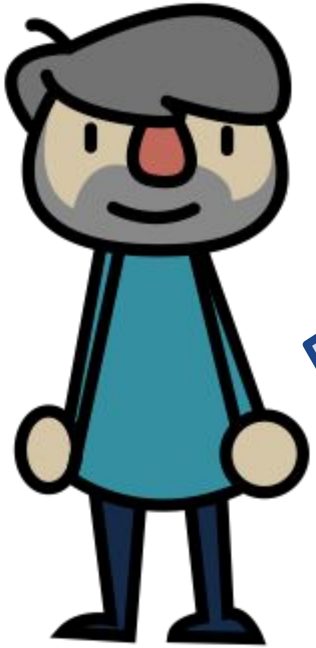
In our case we literally defined these as

new parameters (e.g. the updating window is a weight-vector),

and used **second-order optimization** (i.e.: we differentiated through the forward-back-forward steps of the updating procedure and used SGD)

on **new data** (i.e.: different data than the one the model was trained with).

Can I see some results now?



Yes, yes…

In the following u can see some **results from a prototype** implementation.

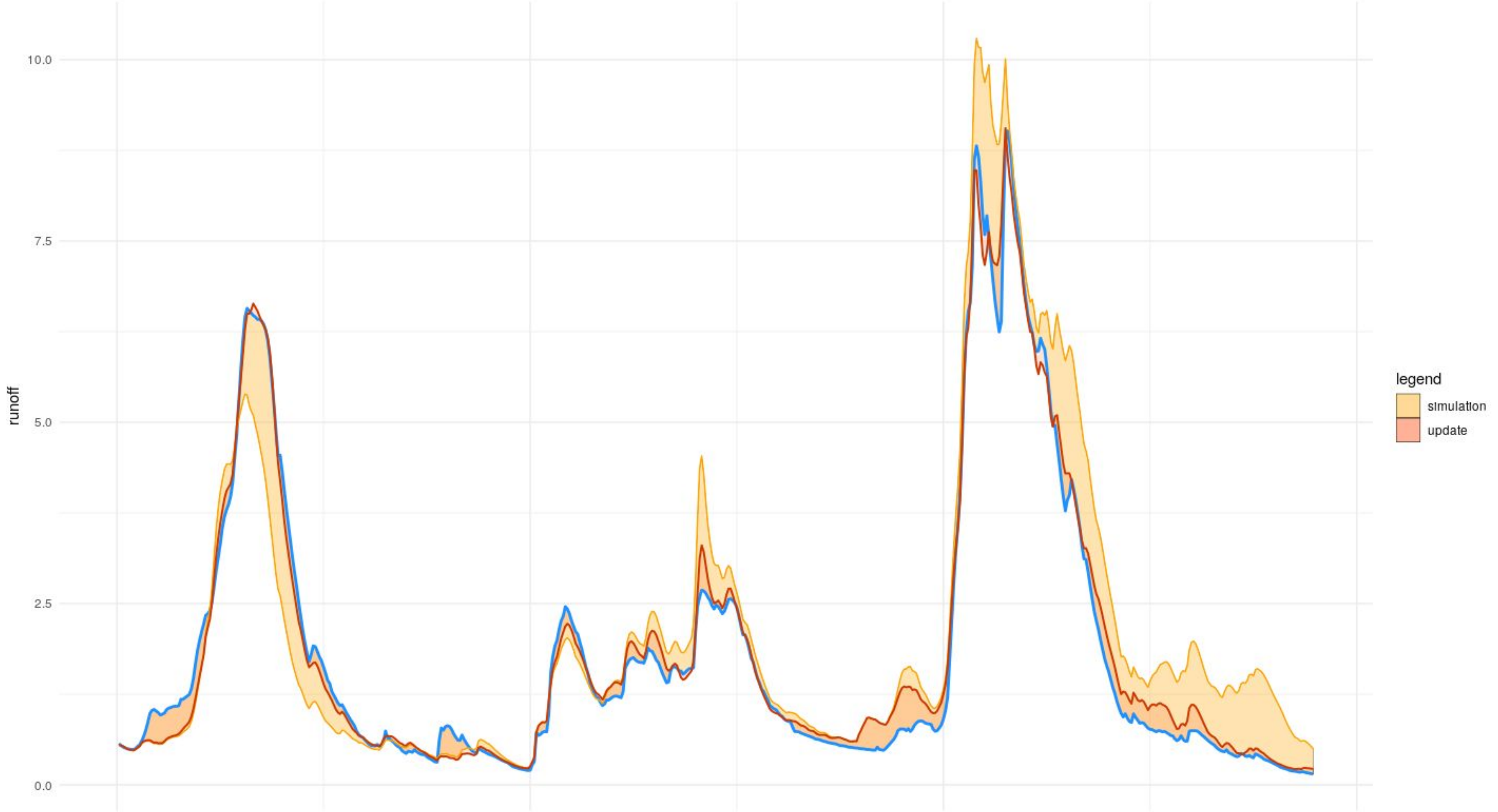
The figures show **two years for different forecasting timesteps** (i.e.: days after the update was made)**.**

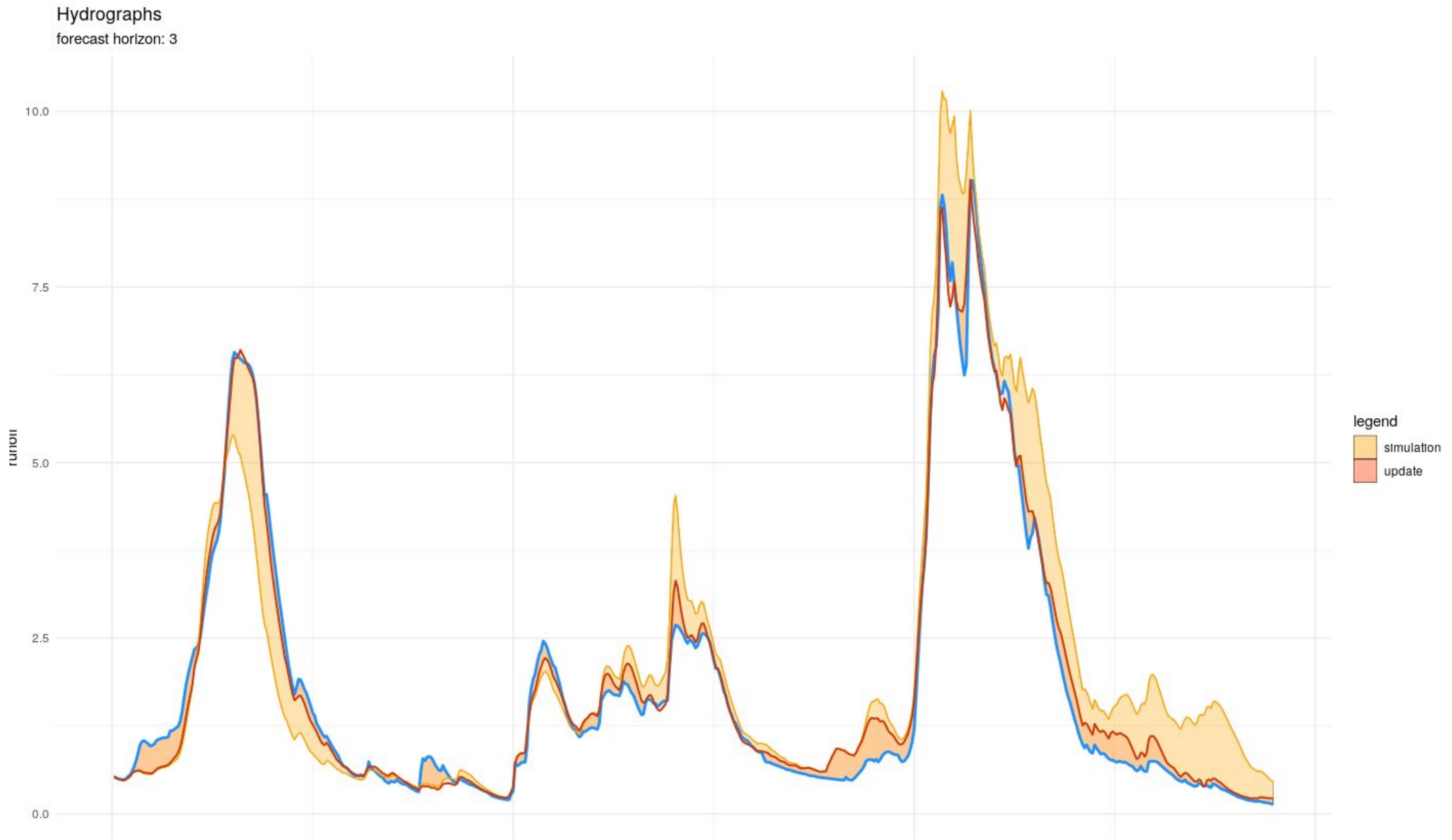
The results are fairly stable.

Finally…

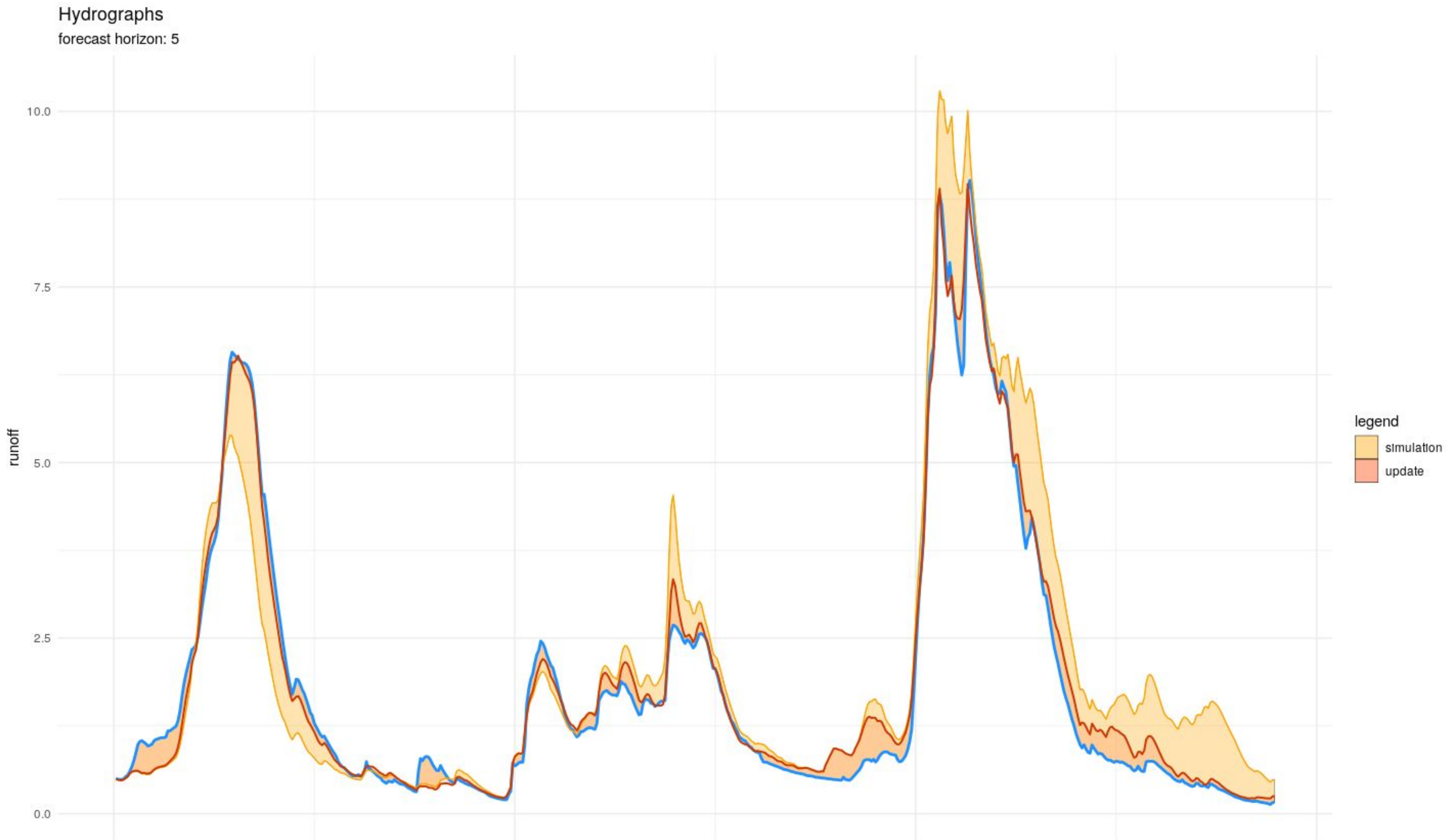
Hydrographs

forecast horizon: 1

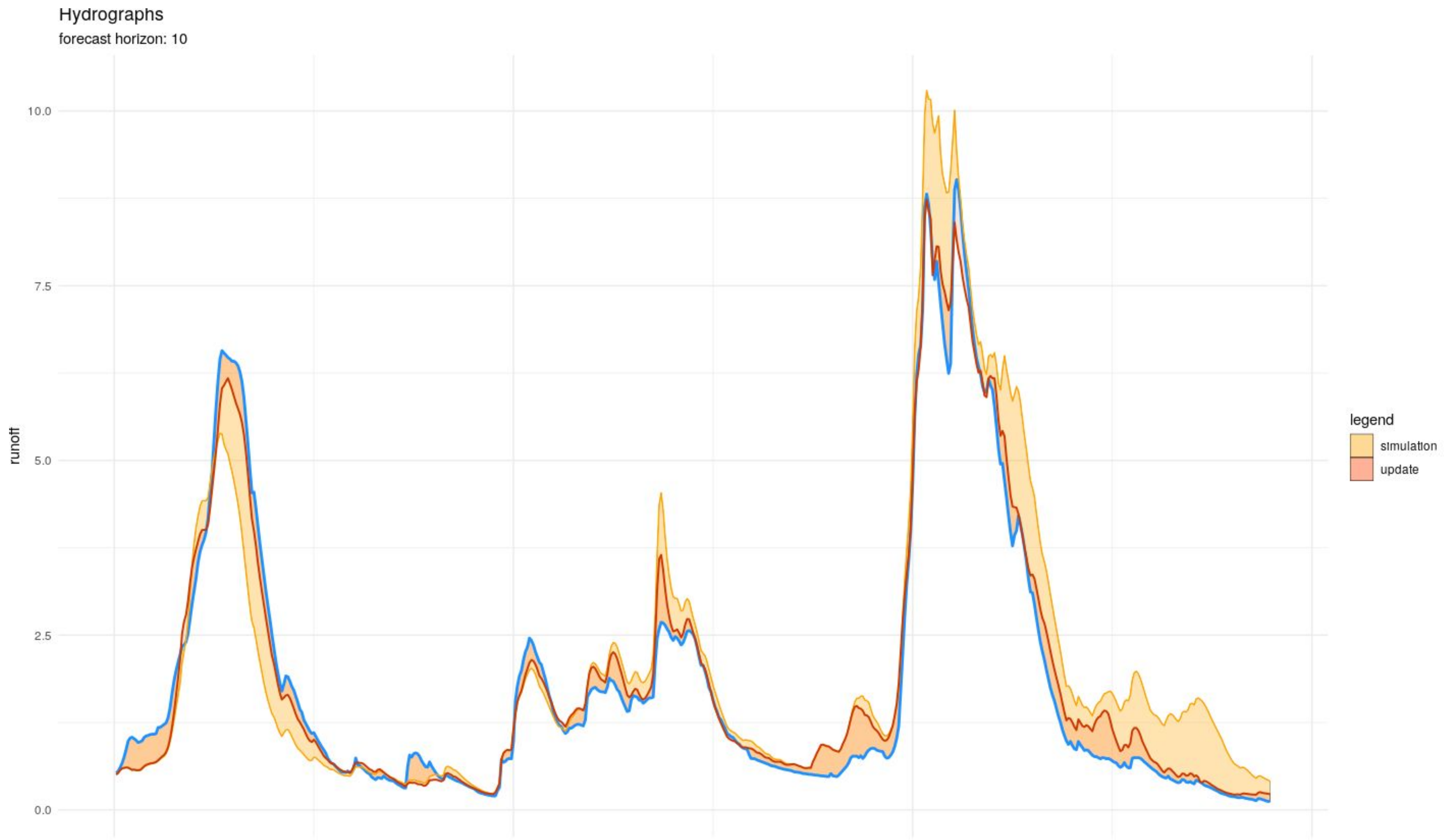




Hydrographs
forecast horizon: 3
runoff
10.0
7.5
5.0
2.5
0.0
legend
simulation
update



Hydrographs
forecast horizon: 5
runoff
10.0
7.5
5.0
2.5
0.0
legend
simulation
update



Hydrographs
forecast horizon: 10
runoff
10.0
7.5
5.0
2.5
0.0
legend
simulation
update


Ok, now that really is all!
Hope to see you around.

Almost…

References:

- Beven, K., & Young, P. (2013). *A guide to good practice in modeling semantics for authors and referees.* Water Resources Research, 49(8), 5092-5098.
- Hoffmann, R., Minkin, V. I., & Carpenter, B. K. (1996). *Ockham's razor and chemistry*. *Bulletin de la Société chimique de France, 2(133), 117-130.*
- Schmidhuber, J. (2015). *Deep learning in neural networks: An overview*. Neural networks, 61, 85-117.