Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-35-AC1, 2020

Geoscientific Model Development Discussions

Interactive comment on "PDE-NetGen 1.0: from symbolic PDE representations of physical processes to trainable neural network representations" *by* Olivier Pannekoucke and Ronan Fablet

Olivier Pannekoucke and Ronan Fablet

olivier.pannekoucke@meteo.fr

Received and published: 23 April 2020

We would like to thank the referee for his/her comments. To answer to some of his/her points:

P: One of the advantages of the convolutional layer translation is implementation in modern deep learning frameworks.

A: We agree with the referee comment, and we exploit convolutional layers to fill the gap between partial differential and neural network. Introduced in a time-integration, this produces an efficient implementation of the dynamics for the known part. Not that the connexion between neural network and differential equation has lead to some better understanding of what can be done by a neural network (ODE-Net, ResNet, bilinear layers,..) and is an active area of research. The package we propose, helps to fill the gap between the statistics and the physics in facilitating the development of useful architectures for evolution equations as encountered in geophysics.

P: Although the authors suggest that the function closure may be modeled with deep learning architectures, no experiments in this direction are shown.

A: The aim of the manuscript is not to introduce a deep learning architecture for the closure, but to facilitate the construction of a deep learning architecture taking into account the known physics: the focus is on the hybridation between the physics and the machine learning. Though the closure itself may not result in a deep architecture, the overall generated model leads to a deep architecture. More precisely, in the reported experiment, the number of layers introduced to train the closure is 6+5+13+25=49 layers for the known part of the dynamics and 2+3+2+4+4=15 layers for the neural network used in the closure. The ResNet implementation of the RK4 uses 11 layers. Hence, there are 75 layers used, with several convolutional layers among them. This is not a simple neural network, but at the end, it is able to learn from the data. The aim of this example was to focus on the neural network generation of the known part of the dynamics in order to facilitate the discovery of unknown terms, and we chose a simple problem to illustrate this. As discussed in Section 3.3 (p10-11), the implementation of an unknown term depends on the amount of knowledge we have. Here we chose to close the term from partial derivatives. For other problems, there would be no other choice than considering a deep neural network, for instance using multiple ResNet blocks, normalization, and so on, or architectures inspired from recent studies on closure modeling (eg, Bolton et al., 2019). And this can be plugged in our package as an exogenous neural network. Note that in the revision of the manuscript we will include an additional way to facilitate the design of the dynamics without plugging an exogenous model – which we think to be easier for the physicist not used to handle neural network layers.

We start to prepare a revision of the manuscript considering his/her comments.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-35, 2020.