Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-327-RC2, 2020

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on "RadNet 1.0: Exploring deep learning architectures for longwave radiative transfer" *by* Ying Liu et al.

Anonymous Referee #2

Received and published: 20 February 2020

Interactive comment on"RadNet 1.0: Exploring deep learning architectures for longwave radiative transfer" byYing Liu et al.

Anonymous Referee

Received and published: 20 February 2020

The paper explores the use of neural networks architectures to model radioactive transfer in global circulation models. The paper is interesting and well written. Authors carried out thorough experiments and the conclusions about the use of NN in the field are interesting. Additionally, the source code for the implementation is publicly available, which is really useful. I have a few concerns about the experiments and how the performance is calculate.

- Speedups of table 2 are obtained by comparing the computational time of RRTMG and NN. I suppose that the baseline time for RRTMG is obtained in the Xeon CPU, and all the values are computed using this baseline. The objective of the paper is to evaluate the advantage of replace a RRTMG for a NN. To do that, a comparison in term of execution time is done. But, to be fair, the comparison of the NN should have been done with a RRTMG implementation for GPU. In the conclusions, authors say that "a complete rewrite of RRTMG for GPUs gives very similar performance gains as what we have achieved in our setup using NNs (Price et al., 2014; Mielikainen et al., 2016)". You should include evidences to support this affirmation. For example, in Mielikainen et al. author state that their implementation provides a speedup of 202x on a Tesla K40 GPU, compared a single-threaded Fortran counterpart running on Intel Xeon E5-2603 CPU. A higher speedup (370x) is indicated in Table 2, for NN model A and a batch size of 4096 running in GTX 1080. Both GPUs architectures are very different, so result are not directly comparable. In fact, single precision floating point (float32) performance is higher in the GTX 1080 that in the Tesla K40 (it's the opposite for double precision).

- Related with the previous question, Intel Xeon CPU E3-1230 v5 is a quad-core processor (8 virtual cores). Is the execution of RRTMG single or multi-threaded? This should be indicated.

- More details should be included about how the time is measured. cProfile is used for calculate the execution time, but with times of only 0.31 ms, as indicated in table 2, the accuracy of the measure is not guaranteed, as a typical time granularity on Unix is 1 ms. Alternatives, as the time Python module, could provide more accurate measurements. Additionally, information about how many measurements have been done and how the final values have been computed should be included.

Additional comments

- Introduction: To include a specific section with the related work, instead of including it in the introduction, could be beneficial

- A recent work describing other GPU-based accelerating methods for the RRTMG_LW could be analyzed (https://www.mdpi.com/2076-3417/10/2/649)

- Section 2.2. Is it realistic the perturbed dataset as has been created? Can be these values representative of a real situation?

- Section 4.4. In the single column model simulation, why the NN model F is used? Why not use model A or model C?

- The developed code used TensorFlow, this should be indicated in the paper along with the version employed. It could be also interesting to describe the benefits of using this library in comparison with others, eg PyTorch.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-327, 2020.

Printer-friendly version

Discussion paper