Solid Earth Discuss.,
https://doi.org/10.5194/se-2019-157-RC1, 2019

Interactive comment on "Deep learning for fast simulation of seismic waves in complex media" *by* Ben Moseley et al.

Andrew Valentine (Referee)

andrew.valentine@anu.edu.au

Received and published: 22 December 2019

This paper by Moseley et al. describes how modern deep learning approaches can be used to construct a fast approximation to a seismological forward modelling code. This is an interesting and timely contribution, and the manuscript itself is clear and well-written. I have a few comments, set out below, but I have no hesitation in recommending the manuscript for acceptance once the authors have had an opportunity to respond to these.

- In any ML-based approach, the training data is central to the applicability of the method. The author's trained network appears effective for simulating waveforms in models that are generated using the same criteria as were used to make the

training set. However, I suspect performance will be significantly worse for models that have significantly different character. This is something that deserves more discussion than it receives, perhaps with some examples. A particular concern in practical settings may be how an end-user can assess whether their input model is 'sufficiently close' to the training set.

- The main 'selling point' of the author's approach is that it enables seismograms to be generated significantly faster than would be possible using 'traditional' forward models. However, this comes with a number of caveats that I think need to be discussed more carefully.

    1. (As above) the author's approach is (I suspect) only effective for models that are sufficiently similar to those in the training set. The numerical forward code does not suffer from this restriction, and can handle complexities that aren't present in the authors' setup (e.g. anisotropy, variable density). How much of a speedup could be achieved by using a numerical code that had been designed with prior knowledge of the characteristics of the authors' training set? Put another way: the speedup could be made to seem even more impressive by using a code designed for a vastly more complex setting (e.g. SPECFEM3D) to build the training set. How fair is the comparison that is being presented?
    2. The headline speed comparisons ignore the costs of building a training set and then training the neural networks, which are significant. How many simulations does a user need to envisage performing before the author's approach becomes cost-effective overall? I think this is going to be a rather large number. Again, some discussion of the pros and cons of the author's approach would seem desirable.

- What do the authors foresee as the primary application(s) of their approach? The discussion seems to mainly envisage inversion-related use cases. Some comments on this:

1. The ‘fast seismic inversion’ approach discussed in Section 3 is essentially a variant of the ‘prior sampling’ approach discussed in detail by Käufl et al (2016). The key strength and weakness of this as an inversion strategy is that all samples (i.e. the training set) are generated without reference to any observed data. This enables very rapid inference once data becomes available, but it means that most training samples lie far from the observed data and are largely wasted from the perspective of any one inference. The end result is that inferences are considerably less well-constrained than would be possible with posterior sampling (see Fig. 9 of Käufl et al 2016). The bottom line is that prior sampling only seems a worthwhile strategy for (a) problems where time is of the essence, e.g. earthquake early warning, or (b) problems where the ‘same’ inference task needs to be solved many thousands of times with different data vectors.
2. Using the learned model in Monte Carlo simulations seems superficially attractive, but comes with significant caveats. Fundamentally the inference remains entirely based on the information contained within the training dataset, and so all the limitations of prior sampling remain. The random walk would need to be constrained to only generate models compatible with the training data, if results are to be meaningful. Perhaps it would be possible to progressively retrain the learned simulation as the McMC proceeds, to ensure accuracy in relevant parts of the model space: this starts to move towards the Bayesian optimisation approaches discussed in (e.g.) Wang et al (2013). To play devil’s advocate: if a problem is too complex to tackle using an McMC approach using physical simulations, can we really be confident that a learned model is sufficiently accurate to yield meaningful results? How big a training set is required to capture the full complexity of the physical problem?

Some discussion and commentary on these issues, and other potential applications, would be appreciated.

- In general, we can make numerical simulations faster by introducing physical approximations. In such cases we typically have some intuition for how those approximations will impact upon results. Learned models offer a speedup without explicit physical approximations, but come with uncertainties that are difficult to quantify rigorously, and which may vary considerably depending on the particular set of inputs chosen. Would the authors like to comment on the pros and cons of the two different strategies for reducing computational costs?

- Referencing, especially in the introduction, seems rather haphazard. If citations are to be given for broad, well-established topics such as the utility of seismic simulations in reservoir characterisation, I would expect these to be to major review papers or to ‘classics’: these are going to be most useful for a reader who is unfamiliar with the field. Without intending any criticism of the cited works, this does not really seem to be the case at present. Moreover, the authors’ survey of the history of machine learning in geophysics is very short-sighted, ignoring anything more than a couple of years old. There are neural network papers in the geophysical literature from the late 1970s onwards, and it would be nice to see some acknowledgement of this body of work. Valentine & Trampert (2012) is probably the first instance of ‘deep learning’ in seismology, though the term had not been invented at that point (and we did not have the benefit of modern computational frameworks).

References:

Käufl et al, 2016. Solving probabilistic inverse problems rapidly with prior samples. https://academic.oup.com/gji/article/205/3/1710/656483

Valentine & Trampert, 2012. Data space reduction, quality assessment and searching of seismograms: autoencoder networks for waveform data. https://academic.oup.com/gji/article/189/2/1183/622660

Wang et al, 2013. Bayesian Optimization in a Billion Dimensions via Random Embeddings. https://arxiv.org/abs/1301.1942

Interactive comment on Solid Earth Discuss., https://doi.org/10.5194/se-2019-157, 2019.