Inversion with the Born approximation in a deep learning framework

Zhan Niu*, Jian Sun, Daniel O. Trad
niuz@ucalgary.ca

Abstract

Least squares reverse time migration (LSRTM) is an important technique that is starting to be used in the industry. LSRTM is closely related to Full-waveform Inversion (FWI) but instead of seeking for an optimal velocity model, it searches for an optimal reflectivity. Machine learning, on the other hand, has gained attention in the geophysics community and has become one of the most booming subjects in computer science. Various tools and methodology have been developed in the last few years and geophysicists have been finding applications by using these tools to solve more efficiently or with better quality long standing processing, imaging and interpretation problems. In this report, we first introduce an implementation of the Born modelling using the recurrent neural network (RNN) and second, we perform an inversion of the model by training the RNN with generated data. The inversion process can be proven to be same as LSRTM. The performance of different optimizers is compared and discussed. We conclude that the ADAM optimizer is the most stable and time efficient for this method.

Theory

$$\left(\frac{1}{v_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right)\boldsymbol{p}_0 = \boldsymbol{f}$$
$$\left(\frac{1}{v_0^2}\frac{\partial^2}{\partial t^2} - \nabla^2\right)\delta\boldsymbol{p} = \frac{1}{v_0^2}\cdot\boldsymbol{m}\cdot\frac{\partial^2}{\partial t^2}\boldsymbol{p}_0 \tag{1}$$

Based on the Born modelling theory, the perturbation wavefield can be considered as the wavefield of a source that is the zero-lag cross-correlation of the model and the second time derivative of the background wavefield. With the above equations, we can calculate the perturbation wavefield given a velocity perturbation.

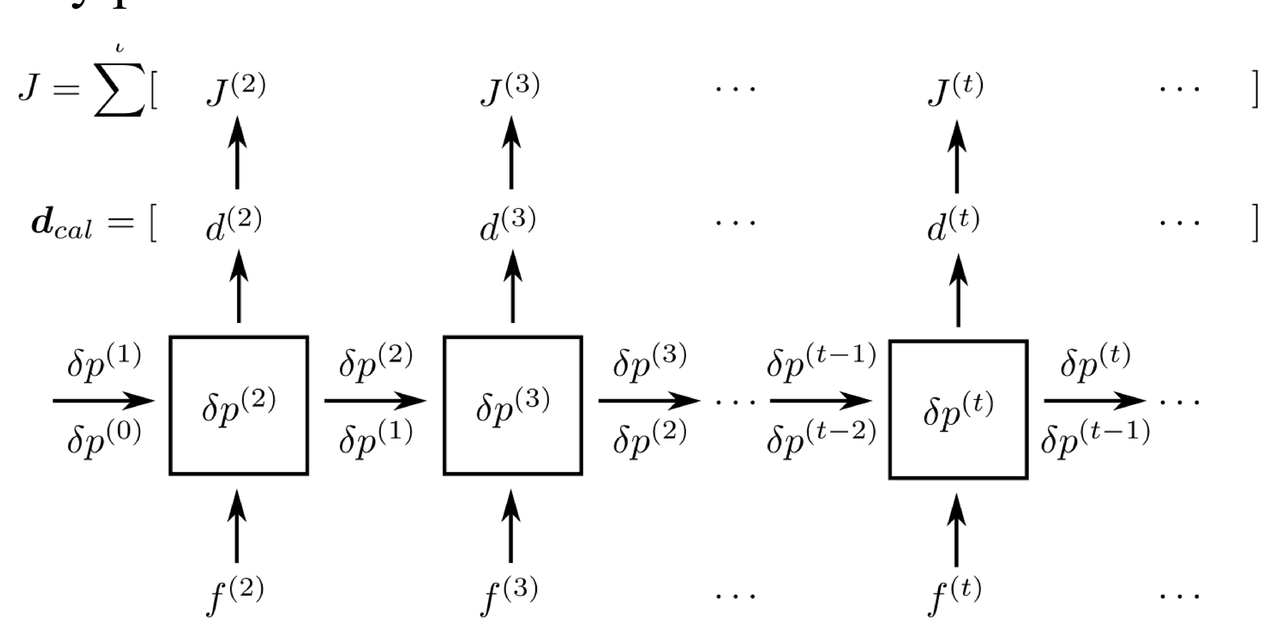

FIG. 1. The diagram of the RNN structure. The black boxes are neural cells which take the source and two previous perturbation wavefields to compute the next perturbation wavefields. The output of each cell is the shot record at a given time and the most recent two wavefields will be passed to the next cell.

Theory

Since the error was summed through out the time, the gradient is the sum of all gradients at each time step as well:

$$\boldsymbol{g}_i = \sum_{t=0}^{T}\boldsymbol{g}_i^{(t)} = \sum_{t=0}^{T}\frac{\partial J^{(t)}}{\partial \boldsymbol{m}}. \tag{2}$$

The gradients at each time step can be proved to be the dot product of the time-reverse-propagated wavefield of the data residual and the 2nd time derivative of the time-forward-propagation of the background wavefield, i.e.

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{m}} = \frac{1}{\boldsymbol{v_0}^2}\boldsymbol{B}(\boldsymbol{r},\boldsymbol{x},T-t)\cdot\frac{\partial^2}{\partial t^2}\boldsymbol{p}_0(f,\boldsymbol{x},t) \tag{3}$$

The model can be then updated by using the ADAM, Fletcher-Reeves and L-BFGS method.

Synthetic Data Examples

We tested the algorithm on the scattering model and the Marmousi model. Both methods use the zero initial model and a smoothed background velocity.

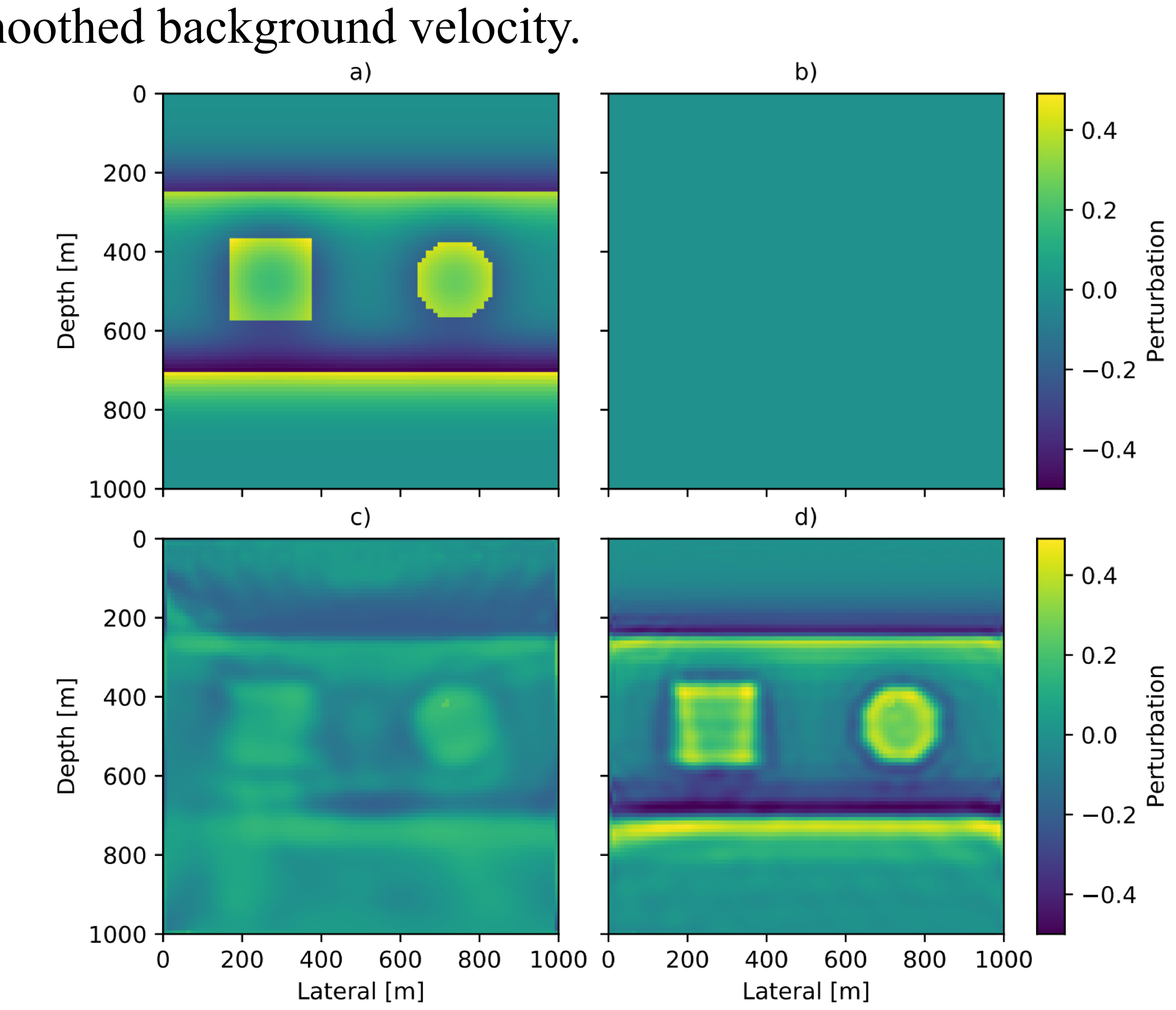

FIG. 2. The updated model at specific iterations. a) The true model; b) The initial model; c) The estimated model at the 10th iteration with ADAM optimizer using learning rate 0.3; d) The model at the 50th iteration.

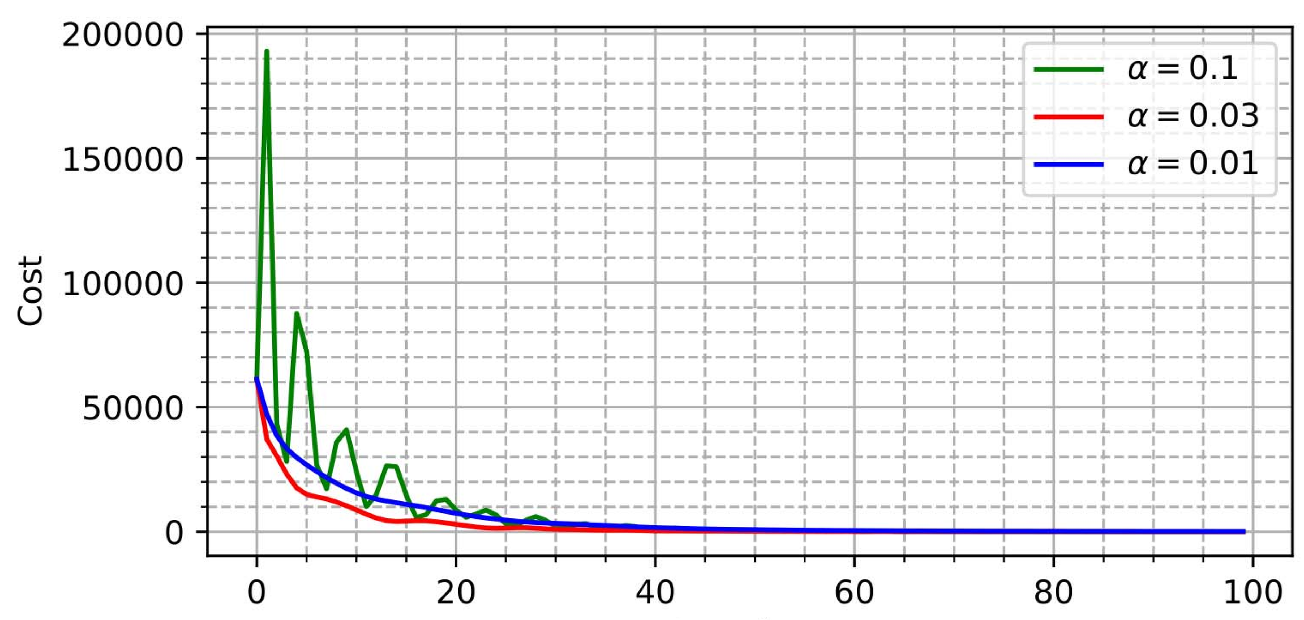

FIG. 3. The cost functions for different value of α with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ in all cases.

Synthetic Data Examples

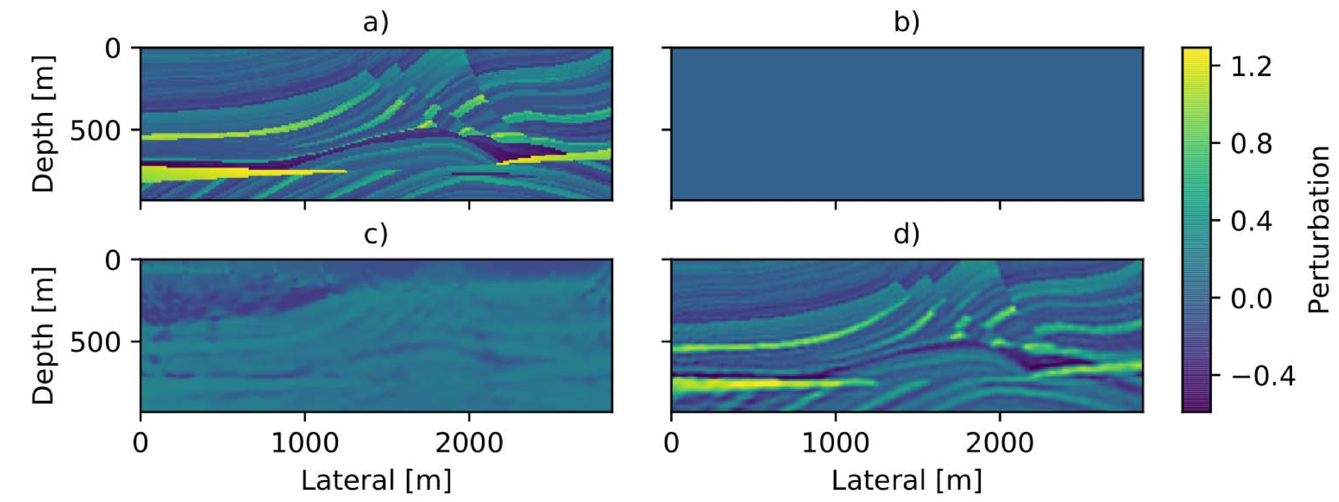

FIG. 4. The inversion results of Marmousi model by RNN. a) The true model; b) The initial zero model; c) The estimated model at the 10th iteration with ADAM optimizer using learning rate 0.3; d) The model at the 50th iteration.

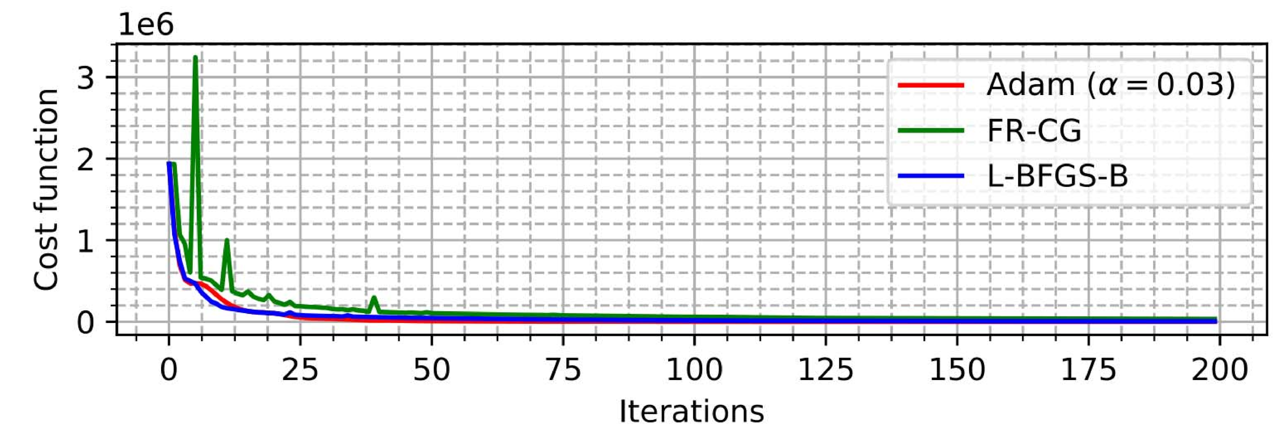

FIG. 5. The first 200 times of cost function calculations using different methods

Conclusion

The Born modelling can be successfully implemented using RNN with TensorFlow. Then, by feeding a theoretical data to the RNN built, the model can be inverted by back-propagation of the RNN. This operation can be proven to be the same as the LSRTM formulation. We found the ADAM method seems to be the most efficient optimizer but it requires to be extra careful when choosing the hyper-parameters. The second efficient optimizer is L-BFGS-B, which does not take extra hyper-parameters. The least efficient optimizer in our tests is the FR-CG, which spends much time in line searching and hence causes too many perturbations to the loss curve. The overall computing performance is good but TensorFlow takes too much time and memory to build the network before the back-propagation. In the future, we are interested in bringing this method to the frequency domain and looking for a more suitable neural network structure for wave propagation.

Acknowledgements

We thank the sponsors of CREWES for continued support. This work was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13.