Feasibility of Machine Learning for Seismic Modeling/Inversion

Hani Alzahrani and Jeffrey Shragge

Center for Wave Phenomenon, Colorado School of Mines
Why neural networks?
Neural networks in seismic modeling

input layer

hidden layers

output layer

ΔV

velocity model perturbation

ΔG

Green’s function perturbation
Network training

1. Design model
2. Generate training data
3. Train network
4. Test network
Network design
Training data

velocity model (v)

1500 m/s

1700 m/s

Green’s function (G)

Design

train data

Test

Train
Random velocity models

- random starting depth
Random velocity models

- 1-5 random control points
Random velocity models

- interpolate between points
- assign velocities to layers
Training data – computing perturbations

\[ \Delta v = (1500 \text{ m/s} - 1700 \text{ m/s}) - (1500 \text{ m/s} - 1700 \text{ m/s}) = \Delta G \]
Training data – 10,000 training input/output pairs

\[ |\Delta v| \leq 200 \text{ m/s} \]
Training the network – forward problem
Predicted wavefield L2 misfit

![Graph showing predicted wavefield L2 misfit over iterations]

- **Design** → **training data** → **Test** → **Train**
Training the network – inverse problem

input $\Delta G$  true $\Delta V$  predicted $\Delta V$
Predicted velocity L2 misfit
Testing the network – forward problem

input $\Delta V$  true $\Delta G$  predicted $\Delta G$
Testing the network – inverse problem

input $\Delta G$  true $\Delta V$  predicted $\Delta V$
Conclusion

• Neural network “learned” forward and inverse problem
  • Forward: predict $\Delta G$ from $\Delta v$
  • Inverse: predict $\Delta v$ from $\Delta G$
Conclusion

• Neural network “learned” forward and inverse problems
  • Forward: predict $\Delta G$ from $\Delta v$
  • Inverse: predict $\Delta v$ from $\Delta G$

• Future work: time-lapse seismic monitoring