Affordable and Fast Geosteering Inversion Using A Physics-Driven Deep Learning Network

YUCHEN JIN, XUQING WU, YUEQIN HUANG AND JIEFU CHEN

University of Houston

Cyentech Consulting LLC.

Outline

Introduction
Background

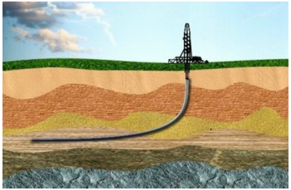


Figure: The schema of directional drilling. (www.amerexco.com)

- Geosteering is a key technique in directional drilling.
  1. The drilling tool could emit a series of electromagnetic waves.
  2. Reflected EM waves are collected by sensors. (**Logging**)
  3. The drilling angle would be adjusted by analyzing collected data. (**Drilling**)
- Logging and drilling need to be synchronous.
- This work is focus on **fast logging**.

Introduction
Background

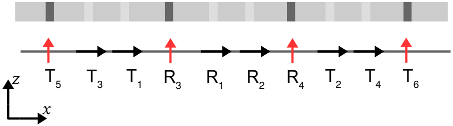


Figure: FWI logging tool with antennas.

- T represents transmitting antennas, and R represents receiving antennas.
- The collected data for each receiver is a combination of the reflected transmitting signals.

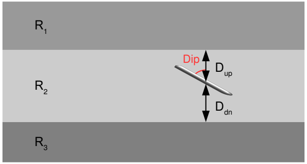


Figure: Directional drilling schema for an example of 3-layer model.

- The **earth model** are formulated by geophysical parameters.
- R represents resistivities, $\mathrm{D_{up}}$ and $\mathrm{D_{dn}}$ are boundaries, and Dip is the dip angle.
- The **observed measurements** are collected by the receiving antennas.

Geosteering Inverse Problem

$$\begin{aligned}\hat{\mathbf{x}} &= \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}) \\ &= \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathcal{F}(\mathbf{x})\|_2^2 + \lambda \mathcal{R}(\mathbf{x}).\end{aligned} \tag{1}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = 2\left(\mathbf{y} - \mathcal{F}(\mathbf{x})\right)\frac{\partial \mathcal{F}}{\partial \mathbf{x}} + \lambda \frac{\partial \mathcal{R}}{\partial \mathbf{x}}. \tag{2}$$

- In (1), the **electromagnetic forward model** could be regarded as a function $\mathcal{F}$ which accepts the **earth model** and produces **synthetic measurements**. $\mathcal{R}$ is a regularization term.
- (2) is usually used in **deterministic optimization** [1, 2]. The gradient $\frac{\partial \mathcal{F}}{\partial \mathbf{x}}$ is a *Jacobian* matrix which could be numerically calculated.

- Two methods for logging.

Table: Different logging methods.

	On ground	Underground
Data Amount	Inadequate	Adequate
Computation	Fast	Slow
Memory	Large	Small

- On ground method.
  - Data is **not enough** but hardware is powerful.
  - Use optimization method.
- Underground method.
  - All data is available but hardware is limited.
  - Use lookup table.

Introduction
Challenge

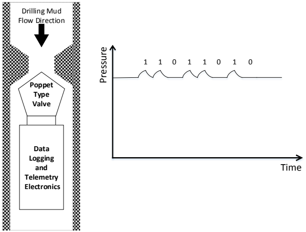


Figure: Positive Pulse Pressure Wave Generator and Corresponding Pressure Waveform with Encoded Digital Data. [3]

- The collected data need to be transmitted back to the ground by **pressure wave**.
- The **communication rate** would be a bottle neck.

Introduction
Challenge

Table X		Table Y
0x0000: A=0, B=0, C=0	⟷	0x0000: F(0, 0, 0)
0x0001: A=0, B=0, C=1	⟷	0x0001: F(0, 0, 1)
0x0002: A=0, B=0, C=2	⟷	0x0002: F(0, 0, 2)
...		...
0x0013: A=0, B=0, C=19	⟷	0x0013: F(0, 0, 19)
0x0014: A=0, B=1, C=0	⟷	0x0014: F(0, 1, 0)
0x0015: A=0, B=1, C=1	⟷	0x0015: F(0, 1, 1)
...		...
0x018F: A=0, B=19, C=19	⟷	0x018F: F(0, 19, 19)
0x0190: A=1, B=0, C=0	⟷	0x0190: F(1, 0, 0)
...		...
0x1F3F: A=19, B=19, C=19	⟷	0x1F3F: F(19, 19, 19)

Figure: Lookup table method for fast estimation of the inversion.

- Use the **best-matched sample** in the table to estimate a coarse solution.
- Drawbacks:
  - Large **memory consumption**.
  - Samples are extremely **coarse**.

Outline

Proposed Method
Network Structure

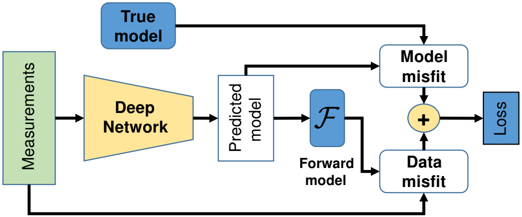


Figure: The deep physics-driven CNN structure.

- The deep network is a 1D network which is adapted from VGG16 model. The model is trained by Adam optimizer [4].
- Each convolutional layer composes of a convolution, an instance normalization [5] and a PReLu activation [6].
- The loss function of the network includes a model misfit and a data misfit.

Train a deep neural network

$$\arg\min_{\boldsymbol{\Theta}} \ \beta_1 \mathcal{L}_{\mathrm{ml}}(\mathbf{y},\ \boldsymbol{\Theta}) + \beta_2 \mathcal{L}_{\mathrm{dl}}(\mathbf{y},\ \mathcal{F},\ \boldsymbol{\Theta}), \quad (3\text{-}1)$$

$$\mathcal{L}_{\mathrm{ml}}(\mathbf{y},\ \boldsymbol{\Theta}) = \|\mathbf{x} - N(\mathbf{y},\ \boldsymbol{\Theta})\|_2^2, \quad (3\text{-}2)$$

$$\mathcal{L}_{\mathrm{dl}}(\mathbf{y},\ \mathcal{F},\ \boldsymbol{\Theta}) = \|\mathbf{y} - \mathcal{F}(N(\mathbf{y},\ \boldsymbol{\Theta}))\|_2^2, \quad (3\text{-}3)$$

- In training phase, we adjust the network parameters $\boldsymbol{\Theta}$.
- The **model misfit** $\mathcal{L}_{\mathrm{ml}}$ is calculated by fitting the ground truth of **earth models** in train set.
- The **data misfit** $\mathcal{L}_{\mathrm{dl}}$ is calculated by letting the **synthetic measurements** fit the **observed ones**.

Train a deep neural network

$$\arg\min_{\Theta} \ \beta_1 \mathcal{L}_{\mathrm{ml}}(\mathbf{y},\ \Theta) + \beta_2 \mathcal{L}_{\mathrm{dl}}(\mathbf{y},\ \mathcal{F},\ \Theta), \quad (3\text{-}1)$$

$$\mathcal{L}_{\mathrm{ml}}(\mathbf{y},\ \Theta) = \|\mathbf{x} - N(\mathbf{y},\ \Theta)\|_2^2, \quad (3\text{-}2)$$

$$\mathcal{L}_{\mathrm{dl}}(\mathbf{y},\ \mathcal{F},\ \Theta) = \|\mathbf{y} - \mathcal{F}(N(\mathbf{y},\ \Theta))\|_2^2, \quad (3\text{-}3)$$

- In training phase, we adjust the network parameters Θ.
- The **model misfit** $\mathcal{L}_{\mathrm{ml}}$ is calculated by fitting the ground truth of **earth models** in train set.
- The **data misfit** $\mathcal{L}_{\mathrm{dl}}$ is calculated by letting the **synthetic measurements** fit the **observed ones**.

Get test results

$$\mathcal{F}^{-1}(\mathbf{y}) \approx N(\mathbf{y},\ \Theta). \quad (4)$$

- In testing phase, the network parameters Θ are fixed.
- The **feed-forward** network could produce the predictions **quickly**.

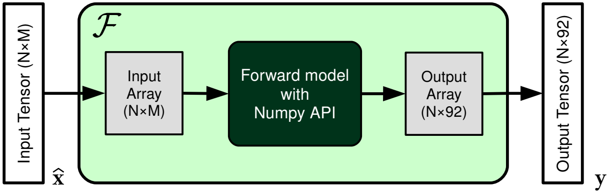


Figure: The implement of the forward model.

- The forward model function is **highly nonlinear**.
- It accepts the **earth model parameters** ($1 \times M$ vector) and produces the **synthetic measurements** (1×92 vector).
- We use N to represent N samples.

Proposed Method
Back Propagation

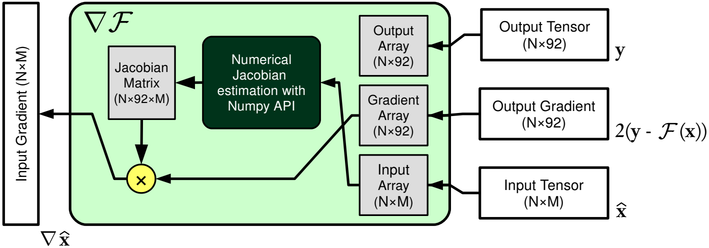


Figure: The implement of the forward model.

- The **back-propagation** only uses the current synthetic input of the forward model ($\hat{\mathbf{x}}$) and the gradient from the next layer ($2(\mathbf{y} - \mathcal{F}(\mathbf{x}))$).
- The gradient would be back-propagated to the previous layer in the deep network.

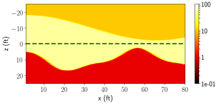
Proposed Method
Advantages

- The network could be deployed for underground method.
  - The network is totally **feed-forward** and only requires **light computation** (about 0.3s for 80 points). The lookup table is slower (about 60s) while the optimization method is much slower (about 400s).
  - The network has **a small data size** (lower than 30MB) compared to a lookup table (about 1.6GB), which requires **lower memory consumption**.
- The network could make use of **all data** by taking advantage of underground method, while the optimization method could not.
- The network could get a far more **accurate prediction** compared to lookup table.
- The **computational cost** of the network **would not increase** with the data amount.

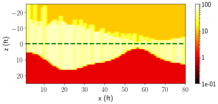
Outline

Results

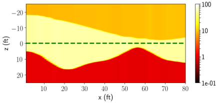
Example: 3-layer model inversion



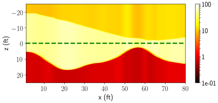
(a) Ground truth.



(b) Lookup table.



(c) DNN with physics.



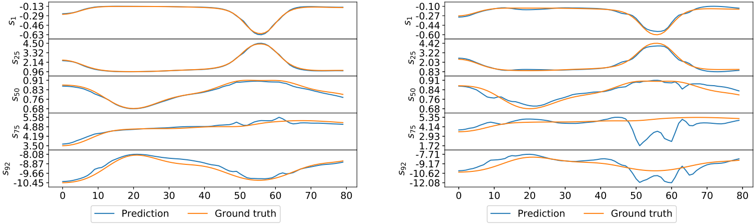
(d) DNN without physics.

Figure: The result of an example.

- The results show the comparison of the predicted earth models.
- The proposed network achieves better resistivity prediction compared to that of the conventional data-driven network.

Results

Example: 3-layer model inversion



(a) With physics engine. (b) Without physics engine.

Figure: The result of an example.

- We select some curves which show that the **physics-driven network(PhDNN)** could achieve a better curve fitness.

Results
Numerical Tests

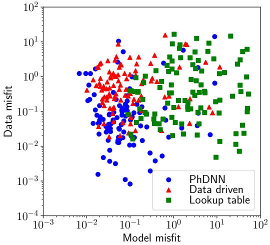


Figure: The numerical tests over compared methods.

- We generated 100 examples earth models like the shown one.
- The test over the 100 examples show that compared to the **data-driven network**, the proposed one could achieve the **same** model misfit but a **better** data misfit.

Outline

Reference I

- K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- N. G. Franconi, A. P. Bunger, E. Sejdi, and M. H. Mickle, “Wireless communication in oil and gas wells,” *Energy Technology*, vol. 2, no. 12, pp. 996–1005.
- D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

Reference II

- D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization. corr (2016)," *arXiv preprint arXiv:1607.08022*, 2016.
- K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

Outline

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science, and Office of Advanced Science Computing Research, under Award Numbers DE-SC0017033.

Thank you for listening!

IT'S TIME FOR Q&A.