Neural estimation of local slopes

Breno Bahia and Mauricio Sacchi
SAIG Annual Meeting
January 2022

SIGNAL
ANALYSIS &
IMAGING GROUP

Neural estimation of local slopes

Breno Bahia and Mauricio Sacchi
University of Alberta
January 2022

SIGNAL
ANALYSIS &
IMAGING GROUP

- Plane-wave destruction (PWD)
  - Slope estimation
    - Classic literature review
- Deep learning
  - Residual Minimization
  - Deep Image Prior
- Examples
  - Neural estimation of local slopes

Plane-wave destruction (PWD)

Local plane wave differential equation

$$\partial_x u(t,x) + p(t,x)\partial_t u(t,x) = 0$$

$u(t,x) \rightarrow \text{wavefield}$

$p(t,x) \rightarrow \text{local slope}$

For a constant slope

$$u(t,x) = f(t - px)$$

Plane-wave destruction (PWD)

Local plane wave differential equation

$$\partial_x u(t,x) + p(t,x)\partial_t u(t,x) = 0$$

$$u(t,x) \rightarrow \text{wavefield}$$

$$p(t,x) \rightarrow \text{local slope}$$

For a constant slope

$$u(t,x) = f(t - px)$$

Applications of local slope fields:

- Denoising and wavefield separation
- (Antialiased) Interpolation
- Velocity-independent imaging
- Attribute analysis

Plane-wave destruction (PWD)

Local plane wave differential equation

$$\partial_x u(t,x) + p(t,x)\partial_t u(t,x) = 0$$

$$u(t,x) \rightarrow \text{wavefield}$$

$$p(t,x) \rightarrow \text{local slope}$$

For a constant slope

$$u(t,x) = f(t - px)$$

Applications of local slope fields:

- Denoising and wavefield separation
- (Antialiased) Interpolation
- Velocity-independent imaging
- Attribute analysis

Local slope estimation tools:

- Slant-stack (Harlan et al., 1984)
- Plane-wave destruction (Claerbout, 1992)
- Dip search (Marfurt, 2006)
- *Structure tensors (Hale, 2007)*

Local slope estimation

Claerbout (1992) - Earth soundings analysis: *PVI*.

Local plane wave differential equation

$$\partial_x u(t,x) + p(t,x)\partial_t u(t,x) = 0$$

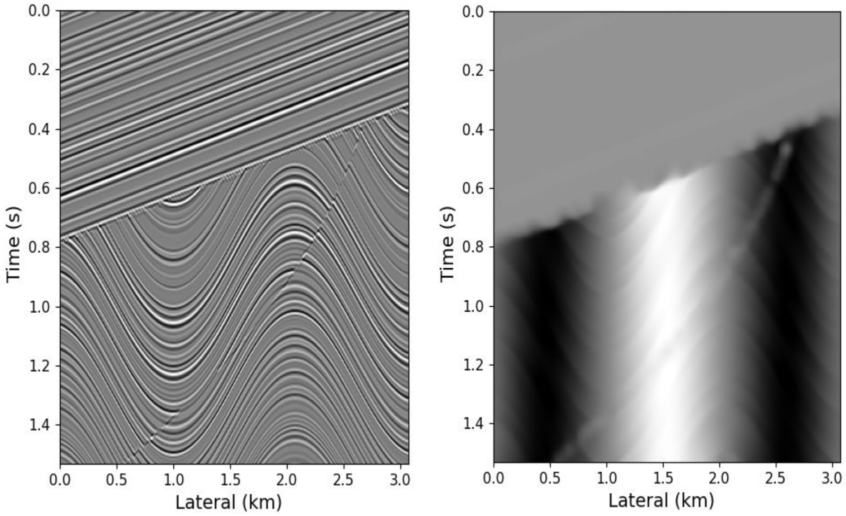
$u(t,x) \rightarrow \text{wavefield}$

$p(t,x) \rightarrow \text{local slope}$

$$J(\mathbf{P}) = \sum_{ix}\sum_{it}\left(\mathbf{U}_x(it,ix) + \mathbf{P}(it,ix)\circ\mathbf{U}_t(it,ix)\right)^2$$

$$= \|\mathbf{U}_x + \mathbf{P}\circ\mathbf{U}_t\|_2^2$$

$$\hat{\mathbf{P}}(it,ix) = -\frac{\sum_{ix}\sum_{it}\mathbf{U}_x(it,ix)\circ\mathbf{U}_t(it,ix)}{\sum_{ix}\sum_{it}\mathbf{U}_t(it,ix)^2}$$



Local slope estimation

Local plane wave differential equation

$$\partial_x u(t,x) + p(t,x)\partial_t u(t,x) = 0$$

$u(t,x) \rightarrow$ wavefield

$p(t,x) \rightarrow$ local slope

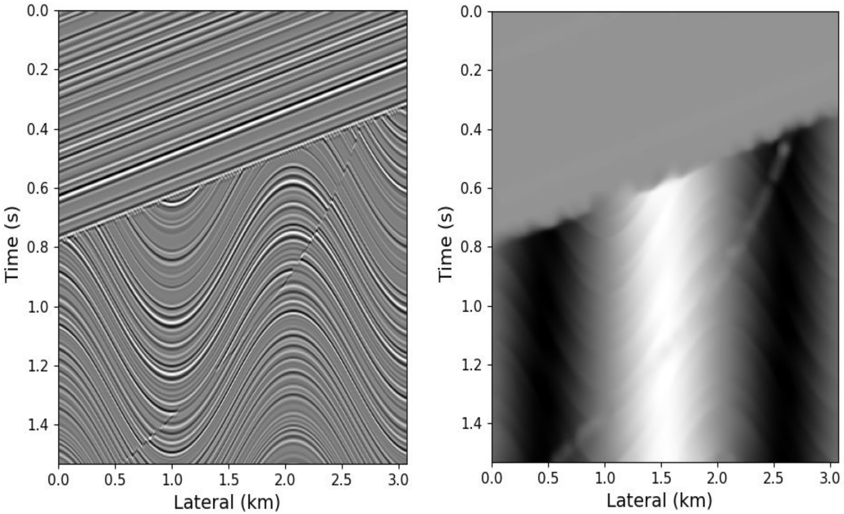
$$J(\mathbf{P}) = \sum_{ix}\sum_{it}\left(\mathbf{U}_x(it,ix) + \mathbf{P}(it,ix)\circ\mathbf{U}_t(it,ix)\right)^2$$

$$= \|\mathbf{U}_x + \mathbf{P}\circ\mathbf{U}_t\|_2^2$$

$$\hat{\mathbf{P}}(it,ix) = -\frac{\sum_{ix}\sum_{it}\mathbf{U}_x(it,ix)\circ\mathbf{U}_t(it,ix)}{\sum_{ix}\sum_{it}\mathbf{U}_t(it,ix)^2}$$

Claerbout (1992) - Earth soundings analysis: *PVI*.

Schleicher et al. (2007) - On the estimation of local slopes. *Geophysics*.



Local slope estimation

Fomel (2002) - Applications of plane-wave destruction filters. *Geophysics.*

$$u(t, x) = f(t - px)$$

$$U(\omega, x + \Delta x) - e^{i\omega p \Delta x} U(\omega, x) = 0$$

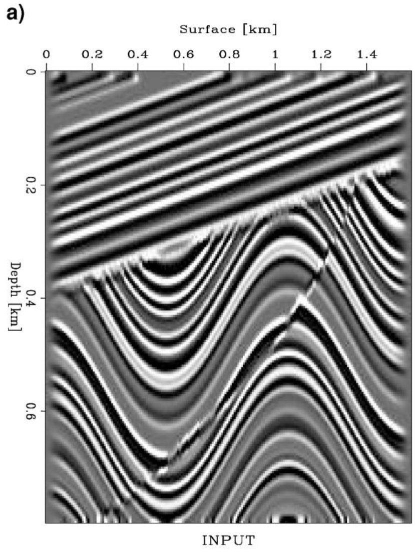
Approximate the phase shift operator by fractional delay filters.

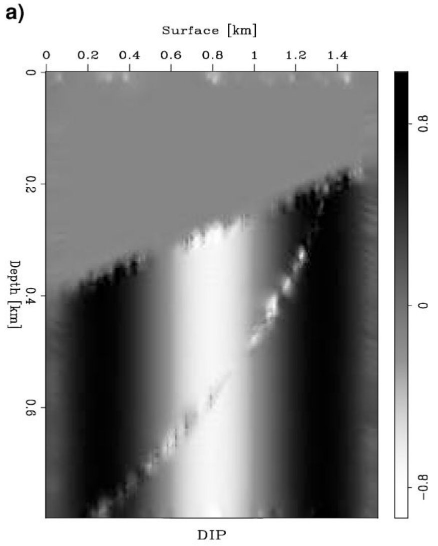
$$\mathbf{C}(\mathbf{P})\mathbf{U} = \mathbf{0}$$

$$(\mathbf{C}'(\mathbf{P}_0)\Delta\mathbf{P} + \mathbf{C}(\mathbf{P}_0))\mathbf{U} = \mathbf{0}$$

$$\lambda_x \mathbf{D}_x \Delta\mathbf{P} = \mathbf{0}$$

$$\lambda_t \mathbf{D}_t \Delta\mathbf{P} = \mathbf{0}$$





Local slope estimation

$$u(t, x) = f(t - px)$$

$$U(\omega, x + \Delta x) - e^{i\omega p \Delta x} U(\omega, x) = 0$$

Approximate the phase shift operator by fractional delay filters.

$$\mathbf{C}(\mathbf{P})\mathbf{U} = \mathbf{0}$$

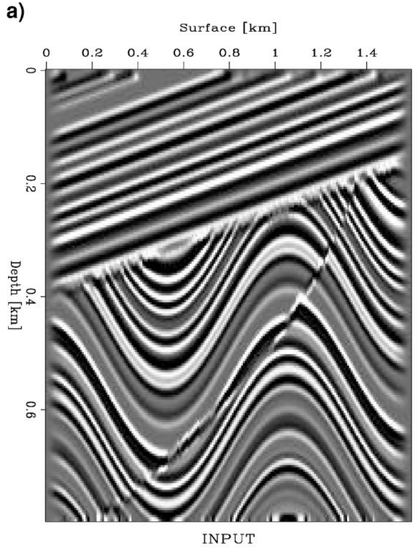
$$(\mathbf{C}'(\mathbf{P}_0)\Delta\mathbf{P} + \mathbf{C}(\mathbf{P}_0))\mathbf{U} = \mathbf{0}$$

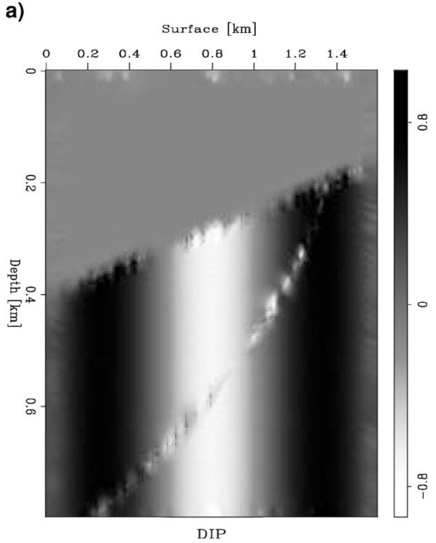
$$\lambda_x \mathbf{D}_x \Delta\mathbf{P} = \mathbf{0}$$

$$\lambda_t \mathbf{D}_t \Delta\mathbf{P} = \mathbf{0}$$

Fomel (2002) - Applications of plane-wave destruction filters. *Geophysics.*

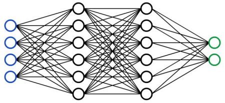
Chen et al. (2012) - Accelerated plane-wave destruction. *Geophysics.*

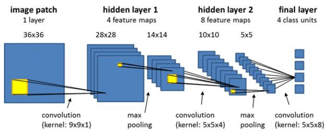


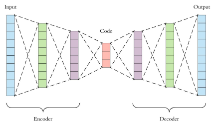


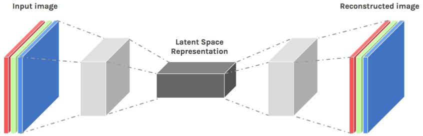
Deep learning

Neural networks









$$\mathbf{z}^{[l]} = \mathbf{W}^{[l]}\mathbf{a}^{[l-1]} + \mathbf{b}^{[l]}$$

$$\mathbf{a}^{[l]} = g^{[l]}.(\mathbf{z}^{[l]})$$

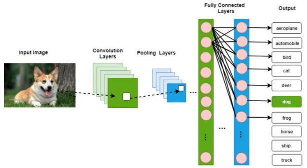
$$l = 1 \ldots L$$

$$\mathbf{a}^{[0]} = \mathbf{x}$$

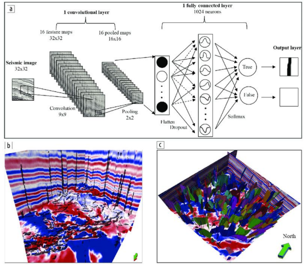
- Two-stage regression model
  - Linear transformation
  - Activation pass
- Learnable parameters
  - Weights (W)
  - Biases (b)
- Go deep
  - Better models
  - Overfitting
- Training
  - Gradient descent
    - Automatic differentiation

Neural networks

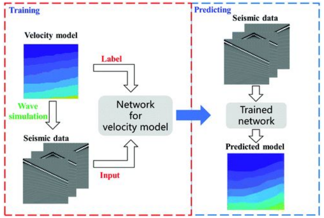
- Classification



- Fault detection

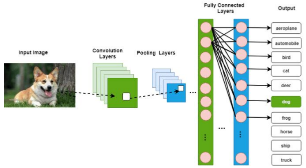


- Inversion

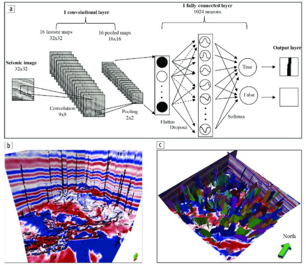


Neural networks

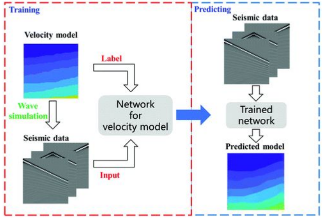
- Classification



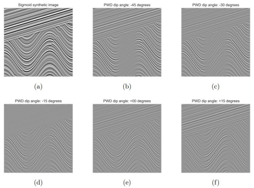
- Fault detection



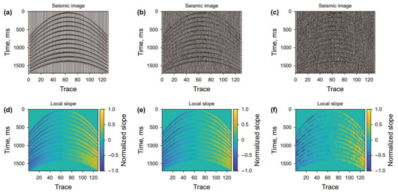
- Inversion



- Slope estimation
  - Shi, Y. (2020) - Plane-wave neural networks. *Chapter VI in Ph.D. Dissertation. (U. of Texas at Austin)*



  - Huang et al. (2021) - A deep learning framework for estimation of seismic local slopes. *Petroleum Science.*



  - Zu et al. (2021) - Slope estimation by convolutional neural networks. *EAGE.*

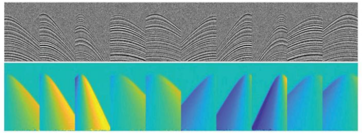


Figure 2 *The training datasets. The first row show the (input data) noisy data and the second row shows the target (accurately local dip).*

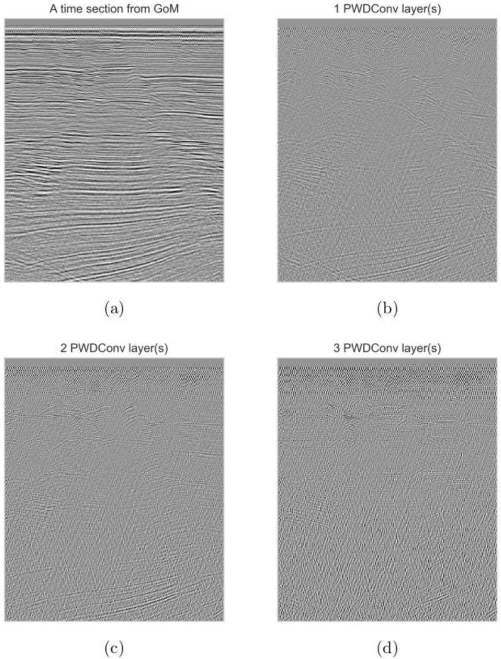
Applications: Slope estimation

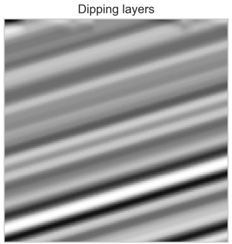
- Shi, Y. (2020) - Plane-wave neural networks. *Chapter VI in Ph.D. Dissertation. (U. of Texas at Austin)*

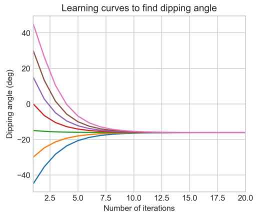
- Customized convolution kernel (PWDConv2D)
    - Add heuristics to the NN
        - Interpretable
        - Tailored for representing seismic data
    - Single meaningful parameter: slope

$$\mathbf{C}(\mathbf{P})\mathbf{U} = \mathbf{0}$$

$$\mathbf{C}(p) = \begin{bmatrix} -\frac{(1+p)(2+p)}{12} & \frac{(1-p)(2-p)}{12} \\ -\frac{(2+p)(2-p)}{6} & \frac{(2+p)(2-p)}{6} \\ -\frac{(1-p)(2-p)}{12} & \frac{(1+p)(2+p)}{12} \end{bmatrix}$$





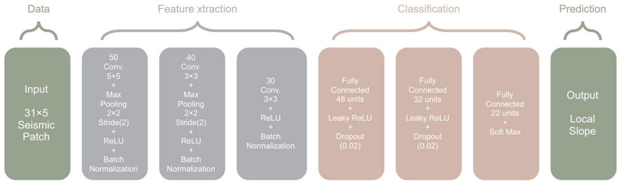


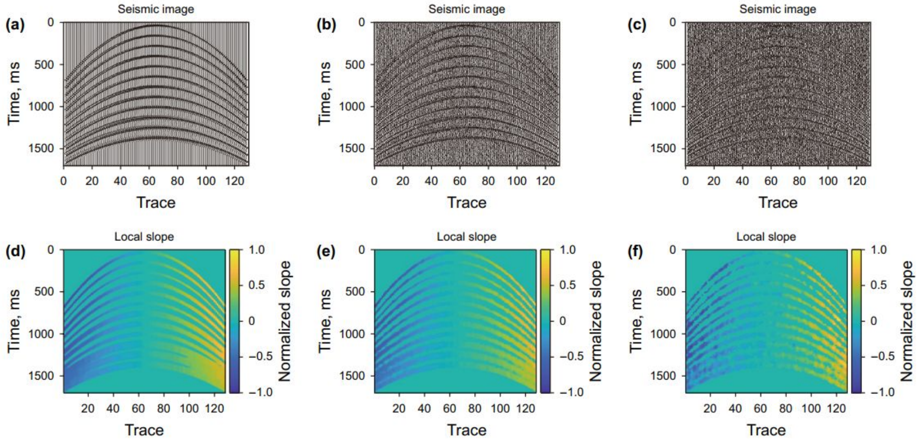
Applications: Slope estimation

- Huang et al. (2021) - A deep learning framework for estimation of seismic local slopes. *Petroleum Science.*

- Conventional CNN with fully-connected layers
  - Slope estimation as a classification problem
  - 5 million patches of synthetic data
  - Labels:
    - 22 groups of slopes

- Resistance to noise





Applications: Slope estimation

- Zu et al. (2021) - Slope estimation by convolutional neural networks. *EAGE Abstract*.

- Local slope estimation from noisy data
    - Supervised
        - Input: Noisy data
        - Labels: Dip field (PWD)
    - Architecture:
        - Convolutional section
        - Deconvolutional section
            - Transposed conv.

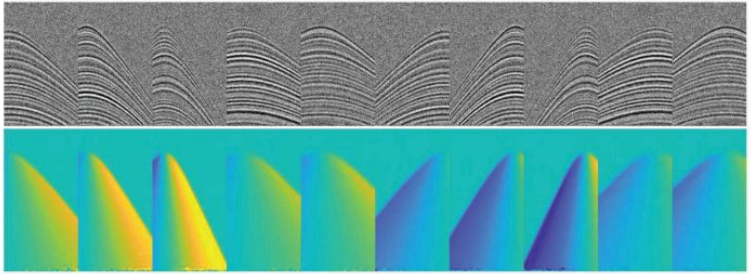


Figure 2 The training datasets. The first row show the (input data) noisy data and the second row shows the target (accurately local dip).

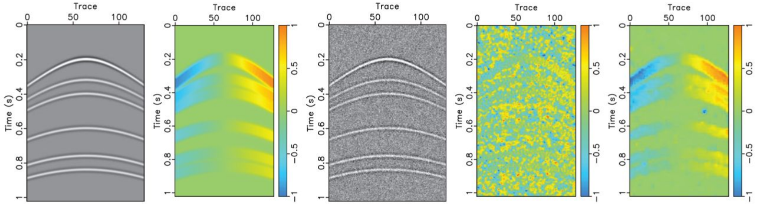


Figure 3 (a) The clean data, (b) the dip estimated from (a) using PWD, (c) the noisy data, (d) the dip estimated from (c) by PWD, (e) the dip estimated from (c) using the trained network.

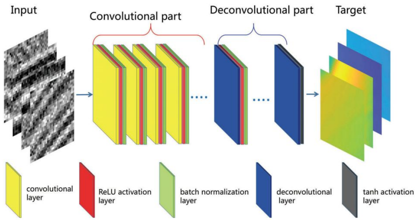


Figure 1 The designed architecture of deep neural network.

Classic machine learning focuses on data-driven models which require minimal domain knowledge and prior assumptions.

- Data-driven (not biased by model)
- Requires big (often labeled) data
- Lots of parameters (very deep models)

Use the underlying physics of our models to train neural networks to estimate parameters of interest.

Scientific Machine Learning (SciML) tries to go beyond the early attempts to bring machine learning into scientific computing such that

1) Less data is required
2) Overfitting is avoided
3) Exploits existing knowledge and tools

Scientific Machine Learning (SciML)

Huang et al. (2020) - Learning constitutive relations from indirect observations using deep neural networks. *Journal of Computational Physics.*

$$\min_\theta \sum_i (\mathbf{P}(\mathbf{u}_i, \mathcal{N}\mathcal{N}_\theta(\mathbf{u}_i)) - \mathbf{F}_i)^2$$

$\mathbf{P} \rightarrow$ Discrete differential operator

$\mathbf{F} \rightarrow$ Discrete external force

$\mathcal{N}\mathcal{N}_\theta(\cdot) \rightarrow$ Neural network with parameters θ

The NN is used to parametrize unknown constitutive relations from physical models.

Example: Slope estimation through PDE residual minimization

Local plane wave differential equation

$$\partial_x u(t,x) + p(t,x)\partial_t u(t,x) = 0$$

$$J(\mathbf{P}) = \sum_{ix}\sum_{it}\left(\mathbf{U}_x(it,ix) + \mathbf{P}(it,ix)\circ\mathbf{U}_t(it,ix)\right)^2$$

Parametrize the slope field by a NN

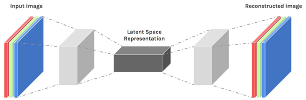
$$\mathbf{P}_{\theta}^{(i)} = \mathcal{N}_{\theta}(\mathbf{U}^{(i)}) = \mathcal{D}_{\gamma}(\mathcal{E}_{\phi}(\mathbf{U}^{(i)}))$$

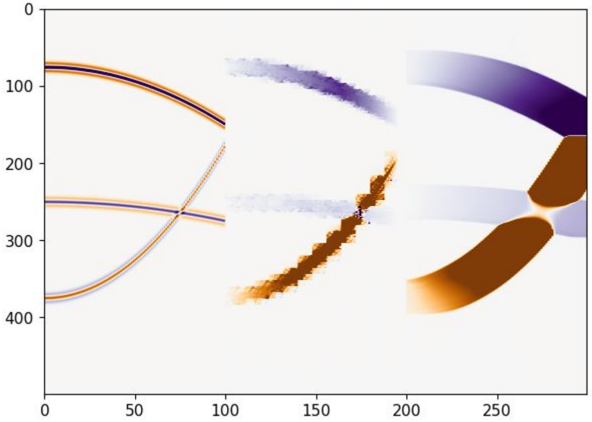
$$J(\theta) = \sum_i \mathbf{D}\left(\mathbf{U}^{(i)}, \mathcal{N}_{\theta}\left(\mathbf{U}^{(i)}\right)\right)^2 = \sum_i\left(\mathbf{U}_x^{(i)} + \mathbf{P}_{\theta}^{(i)}\circ\mathbf{U}_t^{(i)}\right)^2$$

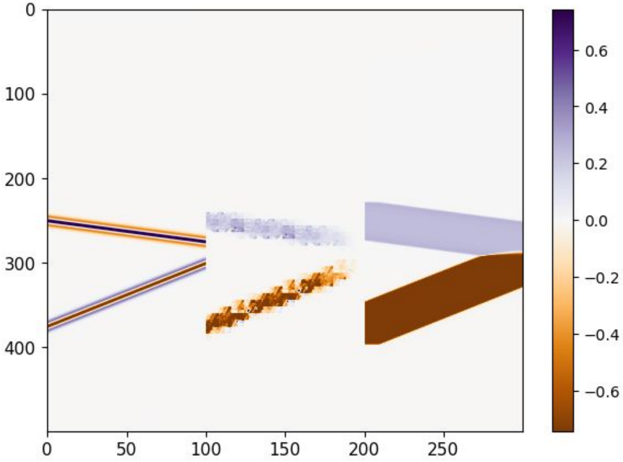
Example: Slope estimation through PDE residual minimization

$$J(\theta) = \sum_i \left(\sum_{ix} \sum_{it} \left(\mathbf{U}_x^{(i)} + \hat{\mathbf{P}}^{(i)} \mathbf{U}_t^{(i)} \right)^2 \right)$$

$$\mathbf{P}_\theta^{(i)} = \mathcal{N}_\theta(\mathbf{U}^{(i)}) = \mathcal{D}_\gamma(\mathcal{E}_\phi(\mathbf{U}^{(i)}))$$







Better generalization.

No active use of labels.

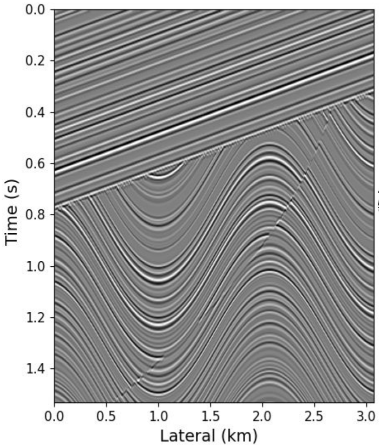
Training is efficient.

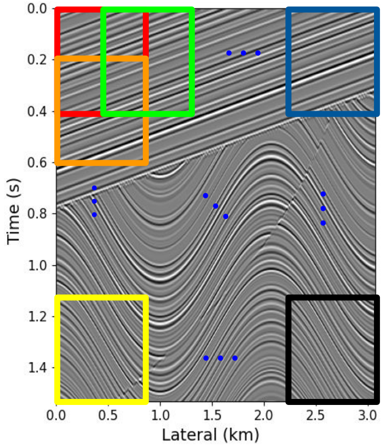
Example: Slope estimation through PDE residual minimization

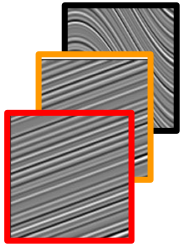
$$J(\theta) = \sum_i \left(\sum_{ix} \sum_{it} \left(\mathbf{U}_x^{(i)} + \hat{\mathbf{P}}^{(i)} \mathbf{U}_t^{(i)} \right)^2 + \lambda_x \|\mathbf{D}_x \hat{\mathbf{P}}^{(i)}\|_2^2 + \lambda_t \|\mathbf{D}_t \hat{\mathbf{P}}^{(i)}\|_2^2 \right)$$

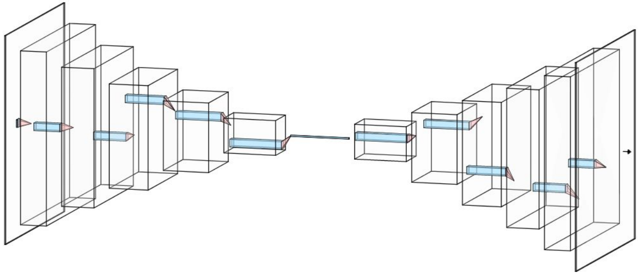
$$\mathbf{P}_\theta^{(i)} = \mathscr{N}_\theta(\mathbf{U}^{(i)}) = \mathscr{D}_\gamma(\mathscr{E}_\phi(\mathbf{U}^{(i)}))$$

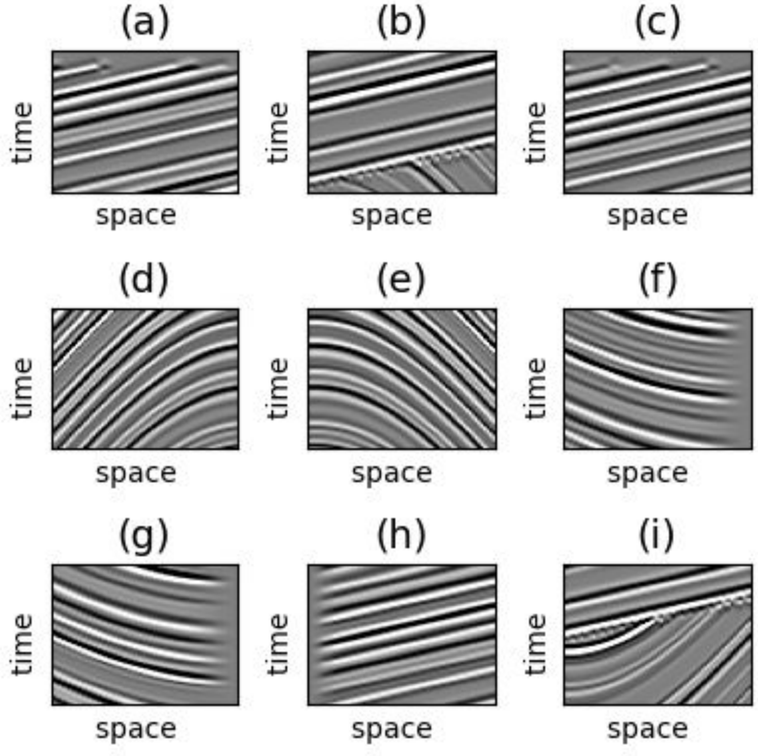
- Training:
  - 64x64 data patches
  - Data augmentation: left-right reversal
  - No active use of labels:
    - The optimization process is guided by the physical model of plane waves.
- Convolutional AutoEncoder:
  - Encoder
    - 6 2x2 convolution layers with 5 tanh activation functions and ReLU in the last layer
  - Decoder
    - 6 2x2 transpose convolution layers with 5 tanh activation functions

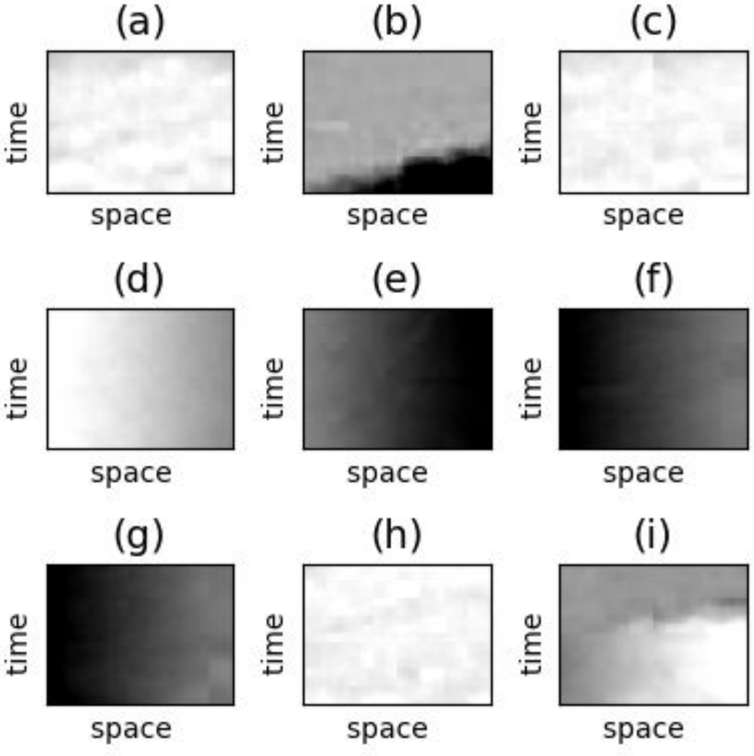






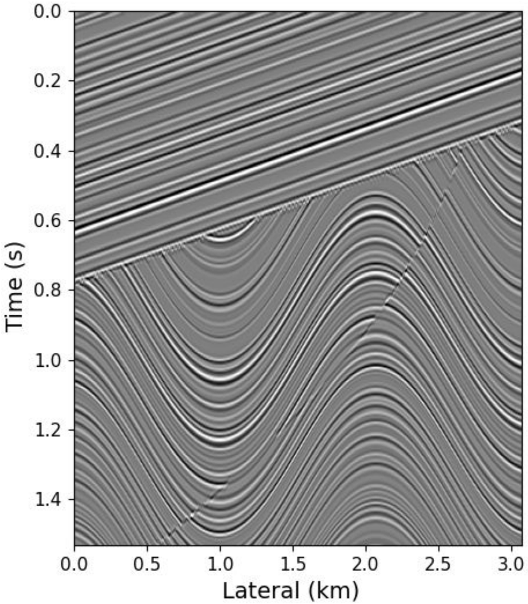




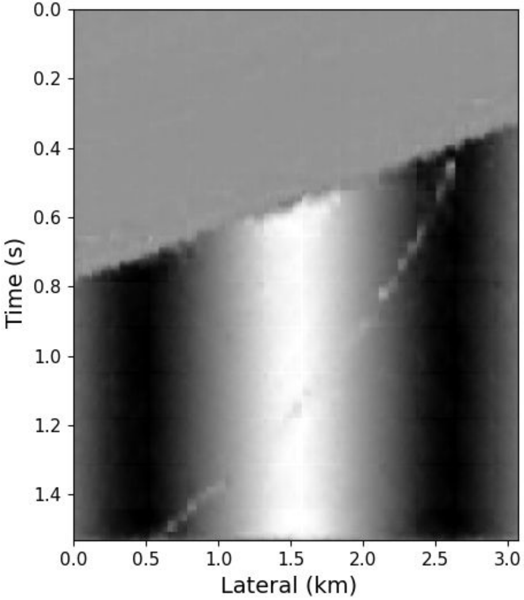


Patched data and their respective slope fields obtained through neural PWD.

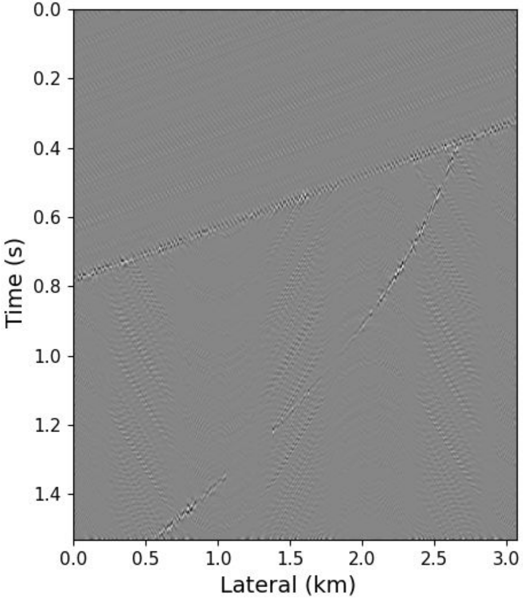
Example: Slope estimation through PDE residual minimization



Input section

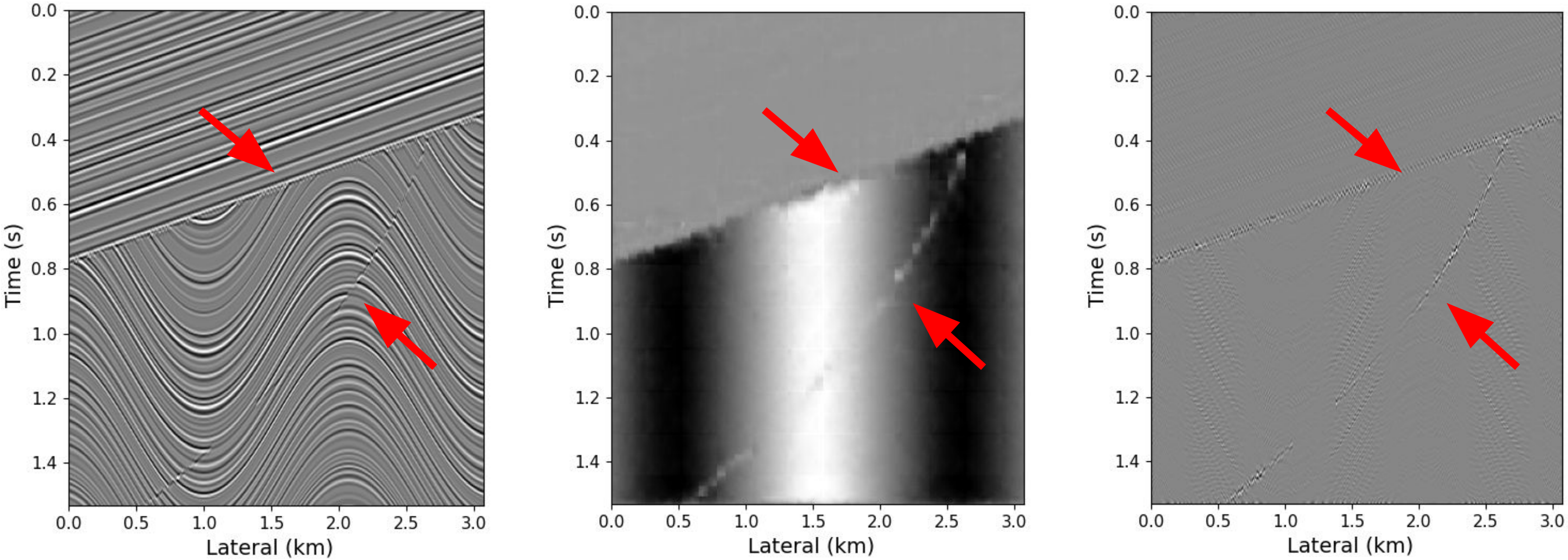


Estimate slope field after 500 epochs of ADAM



PWD result with obtained slope field

Example: Slope estimation through PDE residual minimization

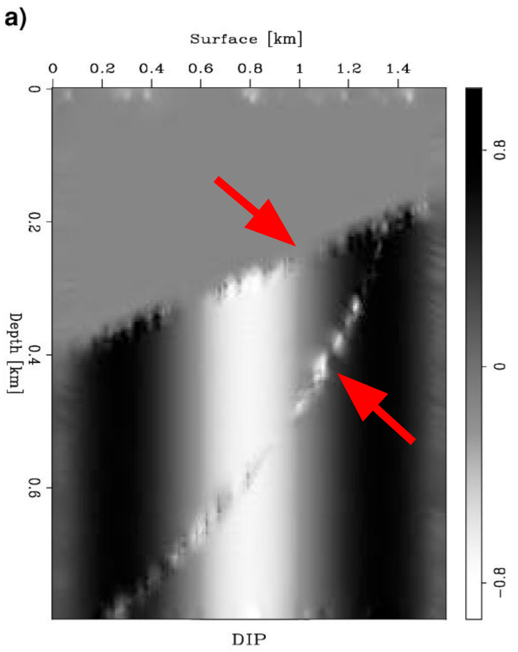


Input section

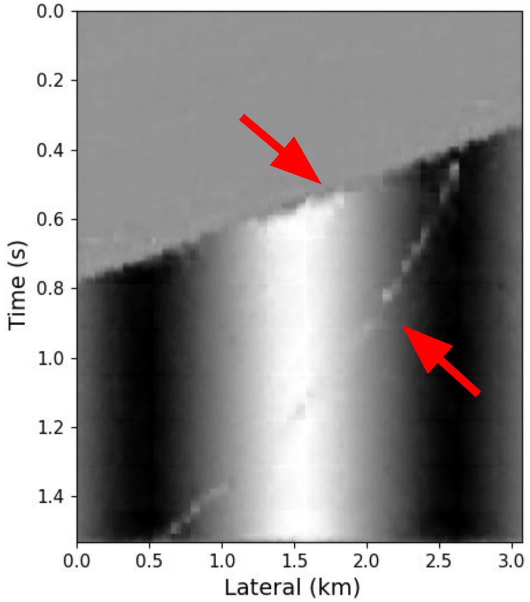
Estimate slope field after 500 epochs of ADAM

PWD result with obtained slope field

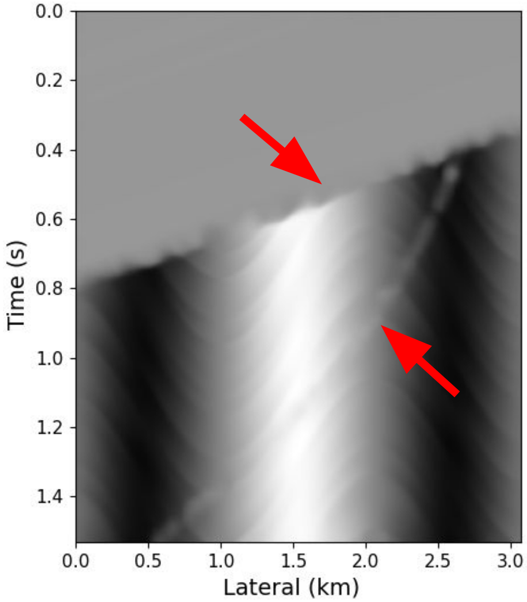
Example: Slope estimation through PDE residual minimization



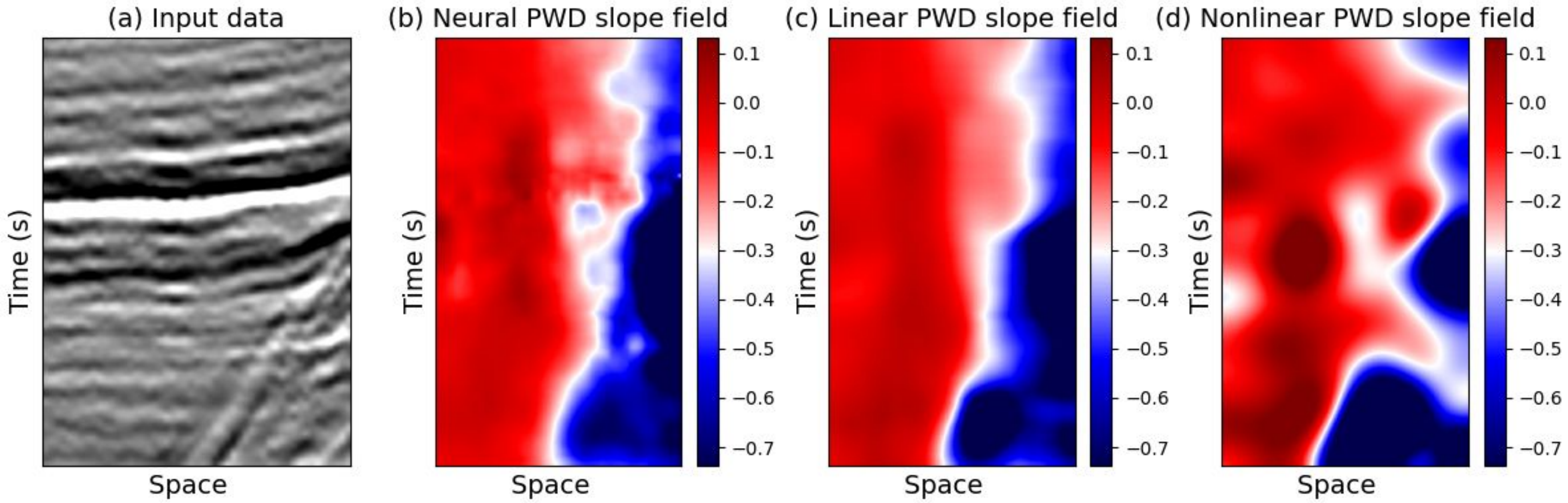
Estimate slope taken from Fomel (2002).



Estimate slope field after 500 epochs of ADAM



Estimate slope by linear least-squares (Claerbout, 1992)



Resulting slope fields for input data (a) obtained with (b) neural, (c) linear, and (d) nonlinear plane-wave destruction filters.

Deep Image Prior (DIP)

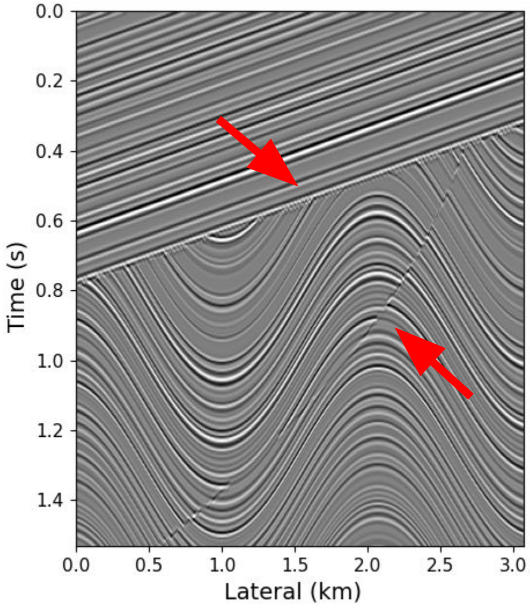
$$J(\theta) = \sum_i \left(\sum_{ix} \sum_{it} \left(\mathbf{U}_x^{(i)} + \hat{\mathbf{P}}^{(i)} \mathbf{U}_t^{(i)} \right)^2 + \lambda_x \| \mathbf{D}_x \hat{\mathbf{P}}^{(i)} \|_2^2 + \lambda_t \| \mathbf{D}_t \hat{\mathbf{P}}^{(i)} \|_2^2 \right)$$

$$J(\theta) = \|\mathbf{y} - \mathbf{A} f_\theta(\mathbf{z})\|_2^2 + \lambda \mathcal{R}(f_\theta(\mathbf{z}))$$

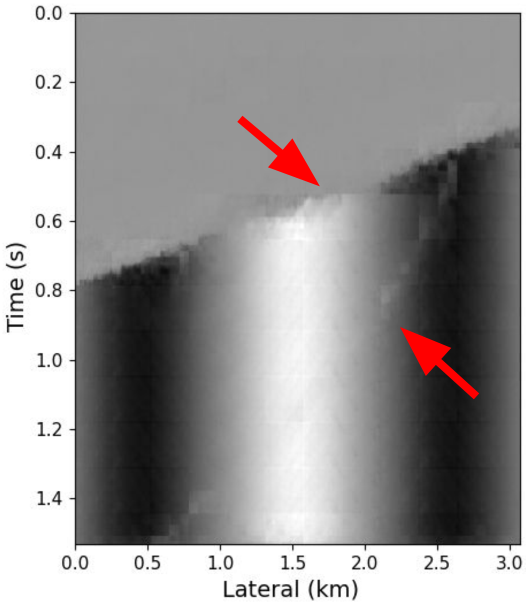
$$\hat{\mathbf{x}} = f_{\theta^*}(\mathbf{z})$$

Ulyanov et al. (2018)

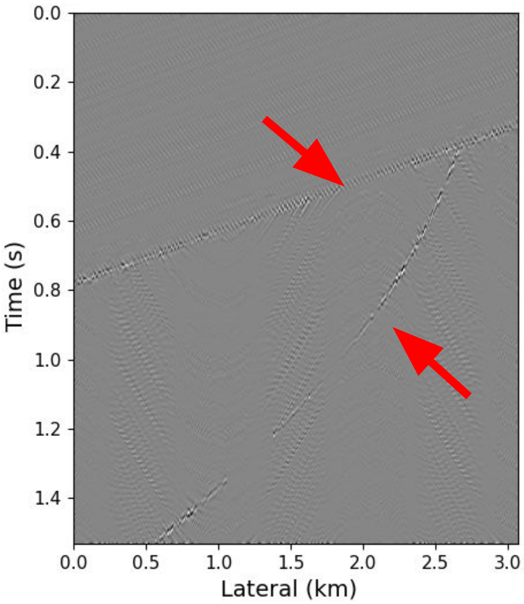
Example: Slope estimation through PDE residual minimization



Input section

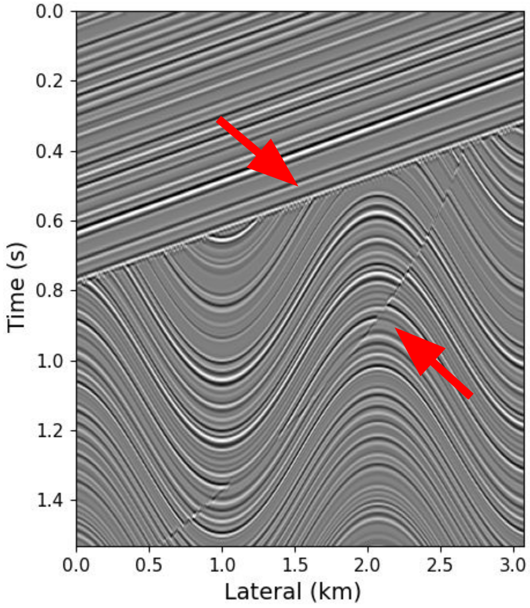


Estimate slope field after 1000 epochs of ADAM

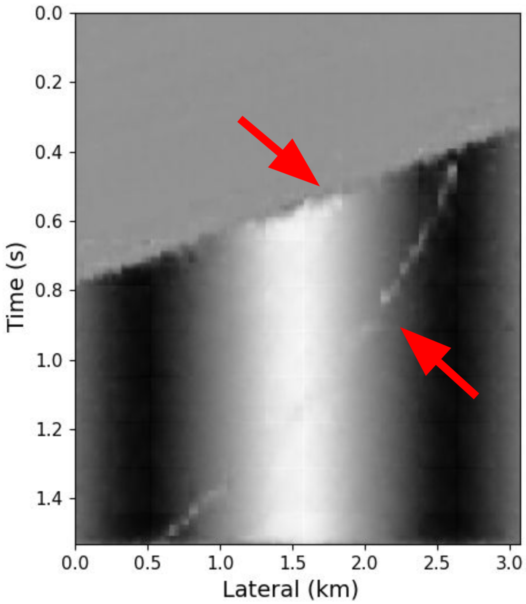


PWD result with obtained slope field

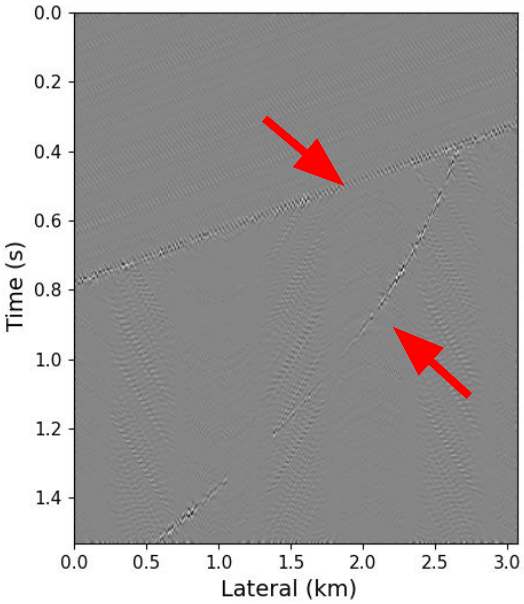
Example: Slope estimation through PDE residual minimization



Input section



Estimate slope field after 500 epochs of ADAM



PWD result with obtained slope field

Conclusions

- Machine Learning is a powerful tool but arguably lacks
    - Generalization
    - Domain knowledge

Conclusions

- Machine Learning is a powerful tool but arguably lacks
  - Generalization
  - Domain knowledge

- SciML: Leveraging ML frameworks and tools for scientific computing
  - Parameter estimation through PDE residual minimization
    - Deep image prior (DIP)

Conclusions

- Machine Learning is a powerful tool but arguably lacks
  - Generalization
  - Domain knowledge

- SciML: Leveraging ML frameworks and tools for scientific computing
  - Parameter estimation through PDE residual minimization
    - Deep image prior (DIP)

- Slope estimation
  - Plane-wave destruction filters
    - Linear least-squares (Claerbout, 1992)
    - Non-linear least-squares (Fomel, 2002)
    - ***Neural network parametrization***
      - Physics-guided
      - Unsupervised
      - Task specific
      - Efficient

SIGNAL
ANALYSIS &
IMAGING GROUP

Local slope estimation

Hale (2007) - Local dip filtering with directional Laplacians. *CWP Report*.

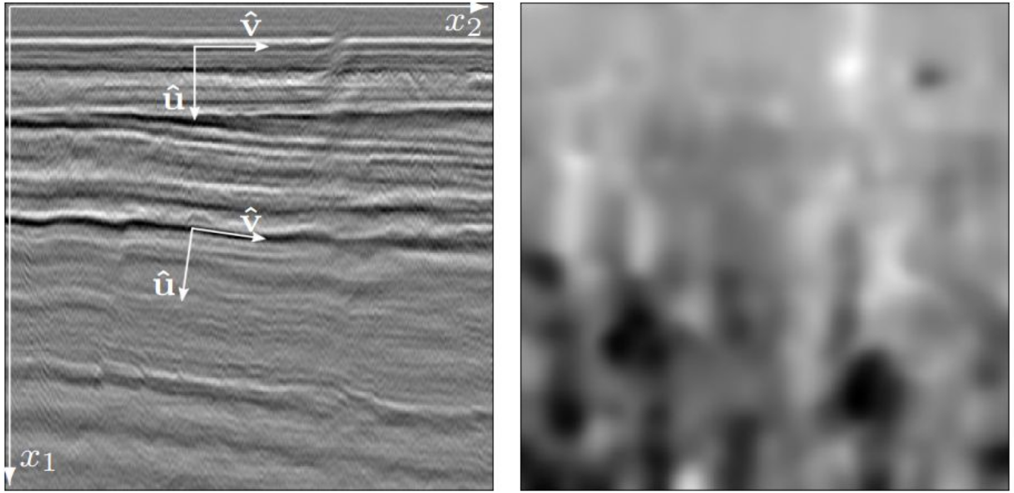
Dip parametrization instead of local slopes.

$$\mathbf{G} = \begin{pmatrix} \langle \mathbf{U}_t, \mathbf{U}_t \rangle & \langle \mathbf{U}_t, \mathbf{U}_x \rangle \\ \langle \mathbf{U}_t, \mathbf{U}_x \rangle & \langle \mathbf{U}_x, \mathbf{U}_x \rangle \end{pmatrix}$$

$$\mathbf{u}, \sigma, \mathbf{v} = \mathtt{svd}(\mathbf{G})$$

$$\mathbf{u} = \begin{pmatrix} \mathtt{cos}\theta \\ -\mathtt{sin}\theta \end{pmatrix} \mathbf{v} = \begin{pmatrix} \mathtt{sin}\theta \\ \mathtt{cos}\theta \end{pmatrix}$$

$$p = \mathtt{tan}\theta$$



(a)

(b)

Future work

Missing data: Penalty method and Physics-constrained Learning

$$J(\mathbf{u},\theta) = \|\mathbf{u}_{obs} - \mathbf{T}\mathbf{u}\|_2^2 + \mu\|(\partial_x + f_\theta(\mathbf{z})\partial_t)\mathbf{u}\|_2^2 + \lambda_x\|\mathbf{D}_x f_\theta(\mathbf{z})\|_2^2 + \lambda_t\|\mathbf{D}_t f_\theta(\mathbf{z})\|_2^2$$

Fix **u** and solve for the NN parameters through back-propagation

$$J_1(\theta) = \mu\|(\partial_x + f_\theta(\mathbf{z})\partial_t)\mathbf{u}\|_2^2 + \lambda_x\|\mathbf{D}_x f_\theta(\mathbf{z})\|_2^2 + \lambda_t\|\mathbf{D}_t f_\theta(\mathbf{z})\|_2^2$$

Fix theta and solve for **u** using the PWD as a model preconditioner

$$J_2(\mathbf{u}) = \|\mathbf{u}_{obs} - \mathbf{T}\mathbf{u}\|_2^2 + \mu\|\mathbf{L}\mathbf{u}\|_2^2$$

$$\mathbf{L} = (\partial_x + \hat{\mathbf{p}}\partial_t)$$

Future work

Similar to Physics-Informed Neural Networks (PINNs)

$$J = \|\mathbf{u}_{obs} - \hat{\mathbf{u}}\|_2^2 + \|(\partial_x + \hat{\mathbf{p}}\partial_t)\hat{\mathbf{u}}\|_2^2$$

$$\hat{\mathbf{p}}(t, x) = f_\theta(t, \mathbf{x}, \mathbf{u}(t, \mathbf{x}))$$

$$\hat{\mathbf{u}}(t, x) = g_\gamma(t, \mathbf{x})$$

Thank you for your attention!

Muito obrigado!