Use of different metrics to generate training datasets for a numerical dispersion mitigation neural network

Гондюл Е.А., Лисица В.В.

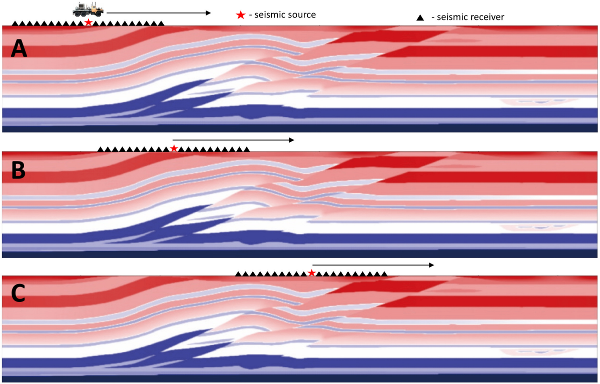
ИНГГ СО РАН, НГУ

September 26, 2023

Introduction: Seismic modeling

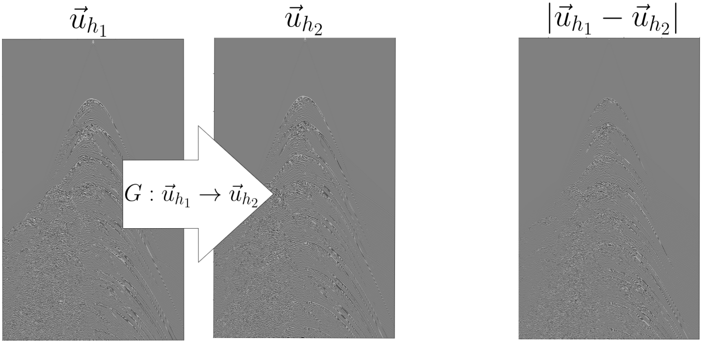
$$L[\vec{u}(t,\vec{x})] = \vec{f}(t)\delta(\vec{x}-\vec{x}_s),$$

where L is the differential operator governing seismic wave propagation, $\vec{u}(t,\vec{x})$ is the velocity, $\vec{f}$ is the pulse and $\vec{x}_s$ is the source position.



Introduction: Seismic modeling

$\vec{u}_{h_1}$ is the numerical solution (**seismogramm**) modeled on a **coarse grid**
$\vec{u}_{h_2}$ is the seismogram modeled on a **fine grid**



$$\frac{||\vec{u}_{h_1} - \vec{u}_{h_2}||}{||\vec{u}_{h_1}||} \times 100\% \approx 69 - 107\%$$

NDM-net

$$\begin{aligned} ||\vec{u}_{h_1} - \vec{u}_{h_2}|| &= \varepsilon^*, \text{ where } h_1 > h_2, \\ ||G(\vec{u}_{h_1}, \vec{\theta}) - \vec{u}_{h_2}|| &= \varepsilon << \varepsilon^*, \end{aligned} \tag{1}$$

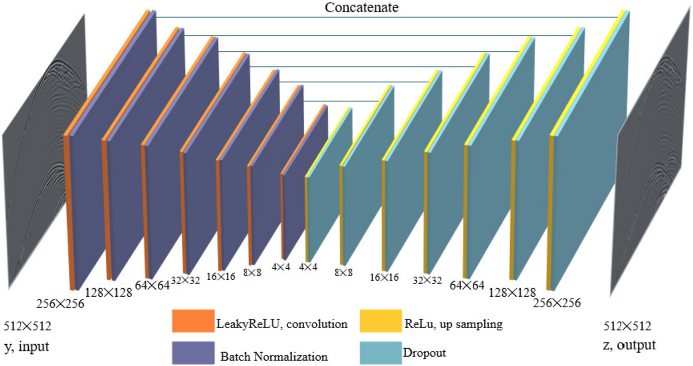
where $\vec{\theta} = \{\mathbf{W}, \mathbf{b}\}$ is the parameter, which includes the weight matrix $\mathbf{W}$ and bias $\mathbf{b}$.

During the training the parameters are optimized by minimizing the loss function:

$$L(\vec{\theta}) = \mathbb{E}_{x,y}[||\vec{u}_2 - G(\vec{u}_1, \vec{\theta})||_1],$$

where $\{\vec{u}_1, \vec{u}_2\}_{i=1}^{N}$ is the training set of size equal to N.

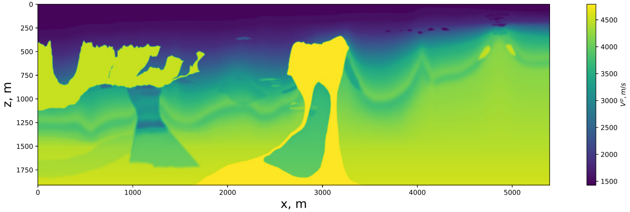
NDM-net


Concatenate
256×256
128×128
64×64
32×32
16×16
8×8
4×4
4×4
8×8
16×16
32×32
64×64
128×128
256×256
512×512
y, input
512×512
z, output
LeakyReLU, convolution
Batch Normalization
ReLu, up sampling
Dropout

Input data: BP Model

The size of the model is 67 km in horisontal direction and 12 km in vertical direction. The acquisition system consists of **2696** sources with a distance of 25 m. The wave field is recorded be 2401 receivers with the distance of 15 km. The distance between receivers is 12.5 km. We used Recker wavelet with a central frequency of 30 Hz.

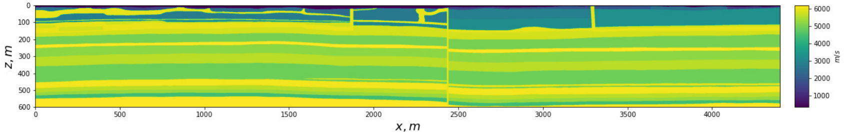
$\mathbf{h_1 = 6m}, \quad \mathbf{h_2 = 3m}$. The seismic data was cropped so that there was no direct wave and had a size of $\mathbf{512 \times 512}$.



Input data: Vanavar model

The size of the model is 220 km by 2.6 km. The aquisition included **1901** sources with a distance 100 m. The wavefield is recorded by **512** receivers with maximal source-receiver offsets of 6.4 km. The distance between the receivers is 25 m. The source is a Ricker wavelet with a central frequency of 30 Hz.

$\mathbf{h_1 = 5m}$, $\mathbf{h_2 = 2.5m}$. We have $\mathbf{512} \times \mathbf{512}$ cropped seismograms.



Training dataset options: Distance between the sources

The most obvious way to construct a training sample is an equidistant arrangement of sources:

$$J^{eq} = 1, k, 2k, ..., q(k)k,$$

where $k \in \mathbb{N}, k > 1$ and $q(k) = N_s/k$.
Then, the distance between the sources is

$$d_d^{ij} = |x_s^i - x_s^j| = |i - j|d,$$

where d is the distance between the two adjusted sources.

Training dataset options: Distance between the seismograms

The metric can be introduced on the base of L^2 norm:

$$d_s^{ij} = d_s(\vec{u}(t, x_o, x_s^j), \vec{u}(t, x_o, x_s^i)) = 2\frac{\|\vec{u}(t, x_o, x_s^j) - \vec{u}(t, x_o, x_s^i)\|_2}{\|\vec{u}(t, x_o, x_s^j)\|_2 + \|\vec{u}(t, x_o, x_s^i)\|_2},$$

where

$$\|\vec{u}(t, x_o, x_s^i)\|_2^2 = \sum_{n=1}^{N_t}\sum_{m=1}^{N_o}\left[u_x^2(t_n, x_o^m, x_s^i) + u_z^2(t_n, x_o^m, x_s^j)\right],$$

where N_t is the number of samples in time, N_o number of traces in the seismogramm.

Training dataset options: Distance in the space of models

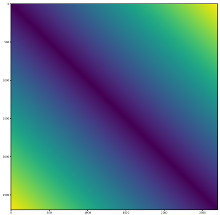
The distance between the models can be introduced as

$$d_m^{ij} = d_m(M(x,z,x_s^j), M(x,z,x_s^i)) = 2\frac{\|M(x,z,x_s^j) - M(x,z,x_s^i)\|_2}{\|M(x,z,x_s^j)\|_2 + \|M(x,z,x_s^i)\|_2},$$

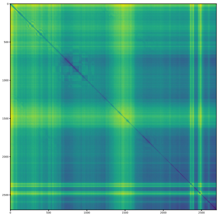
where

$$\|M(x,z,x_s^i)\|_2^2 = \int_0^Z \int_{x_s^i - L_x}^{x_s^i + L_x} \left[v_p^2(x,z) + v_s^2(x,z)\right] dxdz.$$

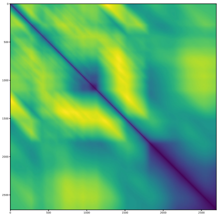
Training dataset options: Distance matrixes



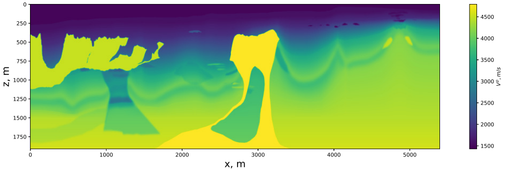
(a) d_d



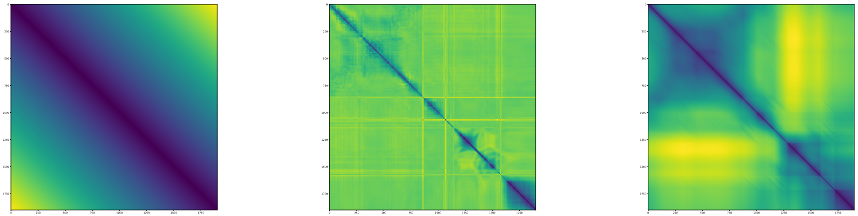
(b) d_s



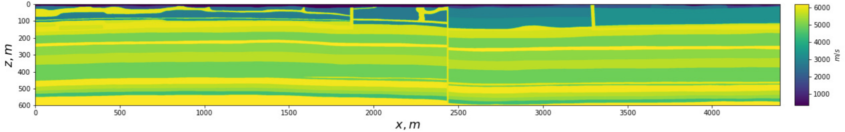
(c) d_m



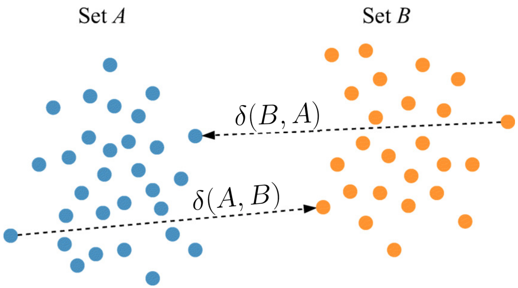
Training dataset options: Distance matrixes (Vanavar)



(a) d_d (b) d_s (c) d_m



Training dataset options: Hausdorff distance

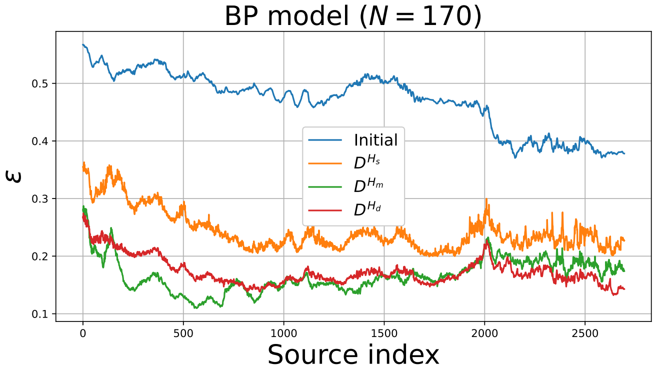


$$\delta_d = \max_{x_s^i \in S} \min_{x_s^j \in S_0} d_d^{ij}(x_s^j, x_s^i),$$

$$\delta_s = \max_{\vec{u}^i \in S} \min_{\vec{u}^j \in S_0} d_s^{ij}(\vec{u}(t, x_o, x_s^j), \vec{u}(t, x_o, x_s^j)),$$

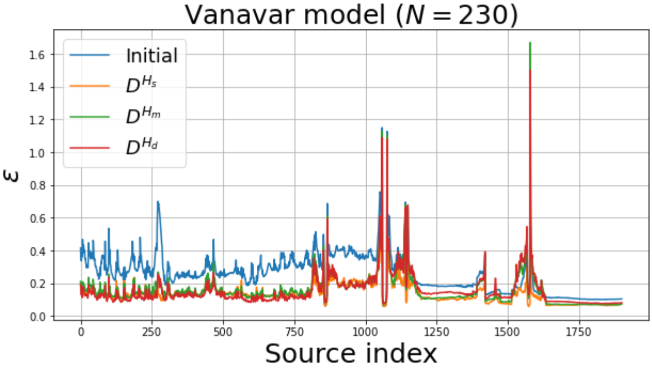
$$\delta_m = \max_{M^i \in S} \min_{M^j \in S_0} d_m^{ij}(M(x, z, x_s^j), M(x, z, x_s^j)).$$

Numerical results



$$\varepsilon = d_s(\vec{u}_{h_2}, G(\vec{u}_{h_1}))$$

Numerical results

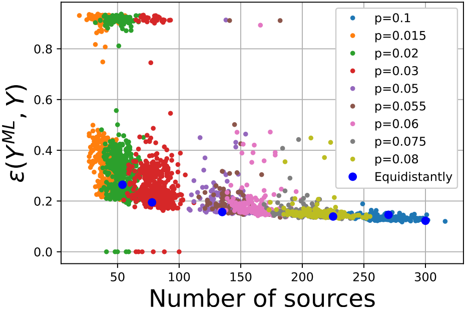


$$\varepsilon = d_s(\vec{u}_{h_2}, G(\vec{u}_{h_1}))$$

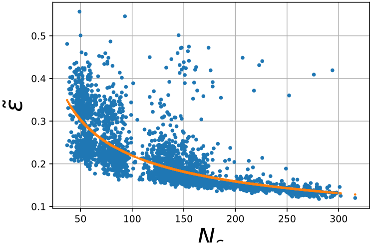
Statistical analysis: random training dataset

We trained a set of neural networks for each randomly generated samples and calculate the errors between all generated seismograms and seismograms modeled on a fine grid:

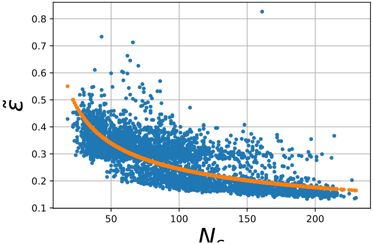
$$\tilde{\varepsilon} = \frac{1}{N_s} \sum_{i=1}^{N_s} d_s \left(G \left[\vec{u}_{h_2}(t, x_o, x_i^s) \right], \vec{u}_{h_1}(t, x_o, x_i^s) \right).$$



Statistical analysis



(a) BP model



(b) Vanavar model

Let $\tilde{\varepsilon} = f(\vec{y})$, where $\vec{y} = (N, \delta_d, \delta_s, \delta_m)^T$.

$$S_i = \frac{Var(\mathbb{E}_{\vec{y}\sim i}(\tilde{\varepsilon}|y_i))}{Var(\tilde{\varepsilon})}, \quad S_i \in [0,1], \quad \sum_i S_i = 1.$$

For BP and Vanavar models $\mathbf{S_N} = \mathbf{0.63}$.

Statistical analysis: for fixed N

We assume that the error can be represented as

$$\tilde{\varepsilon}_{N_c}(\delta_d, \delta_s, \delta_m) = \tilde{\varepsilon}_0 + \alpha_1\delta_1 + \alpha_2\delta_2 + \alpha_3\delta_3 + o(\delta_1, \delta_2, \delta_3),$$

where $\tilde{\varepsilon}_{N_c}$ is the mean error for fixed size of training sample, $N_c = const$, $(\delta_1, \delta_2, \delta_3) = (\delta_d, \delta_s, \delta_m)$.

$$q(\alpha_1\delta_1 + \alpha_2\delta_2 + \alpha_3\delta_3) \to \min, \quad \text{for } \forall q > 0 \in \mathbb{R}.$$

Statistical analysis: for fixed N

$$S_{\delta_i} = \frac{Var(\mathbb{E}_{\vec{\delta}\sim i}(\tilde{\varepsilon}|\delta_i))}{Var(\tilde{\varepsilon})} = \frac{(\alpha_i L_{\delta_i})^2}{\sum_j (\alpha_j L_{\delta_j})^2}, \quad \sum_i S_{\delta_i} = 1,$$

where L_{δ_i} is the extent of δ_i.

$$\widetilde{\alpha}_i = \frac{1}{N_r} \sum_{k=0}^{N_r} \frac{\sqrt{S_i^k \sum_j (\alpha_j^k L_j^k)^2}}{L_i^k}$$

Statistical analysis: for fixed N

	BP	Vanavar
S_{δ_d}	0.499	0.489
S_{δ_s}	0.0442	0.139
S_{δ_m}	0.4565	0.3714

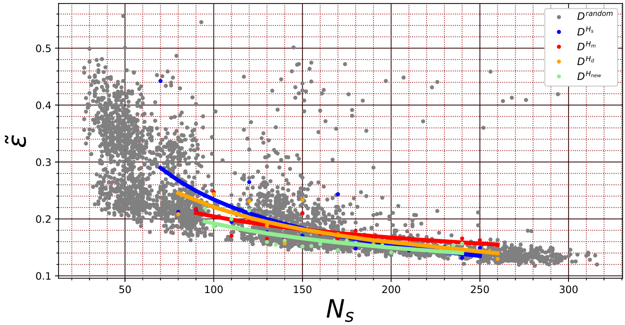
	BP	Vanavar
$\widetilde{\alpha}_d$	0.815	0.812
$\widetilde{\alpha}_s$	0.089	0.112
$\widetilde{\alpha}_m$	0.095	0.075

Statistical analysis: for fixed N

We consudered the minimization problem:

$$0.813\delta_d + 0.1\delta_s + 0.085\delta_m \to \min .$$

BP model

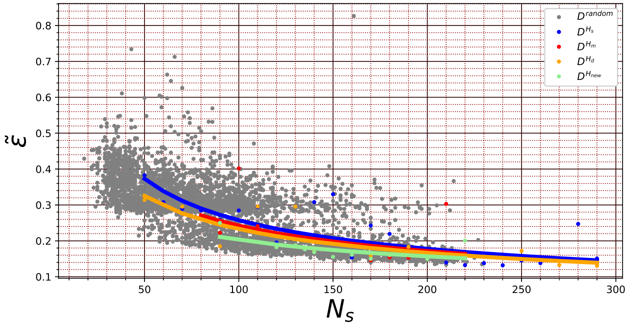


Statistical analysis: for fixed N

We consudered the minimization problem:

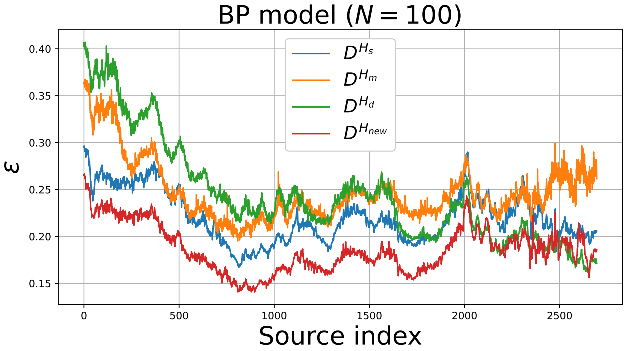
$$0.813\delta_d + 0.1\delta_s + 0.085\delta_m \to \min .$$

Vanavar model



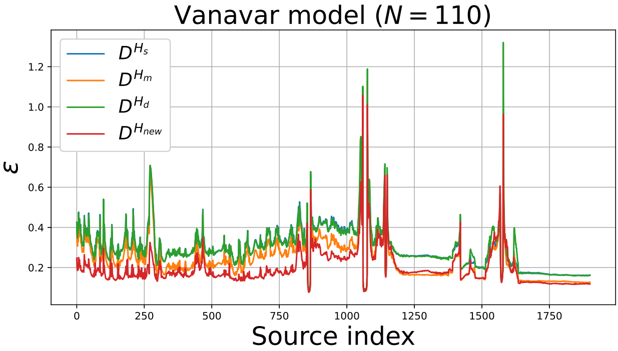
Statistical analysis: for fixed N

$$0.813\delta_d + 0.1\delta_s + 0.085\delta_m \to \min\,.$$

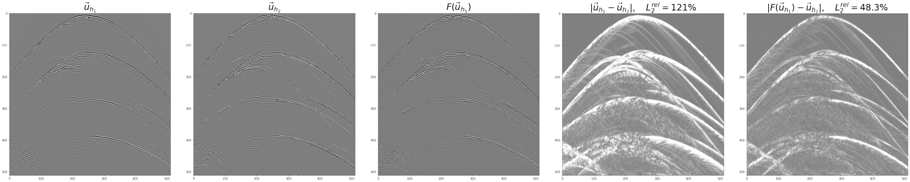


Statistical analysis: for fixed N

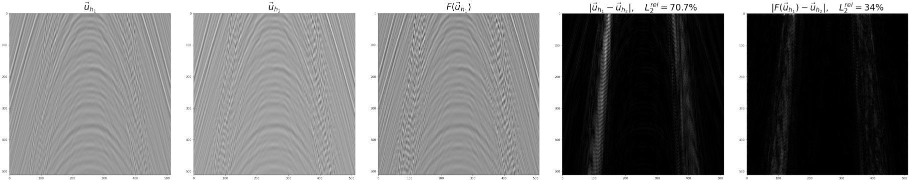
$$0.813\delta_d + 0.1\delta_s + 0.085\delta_m \to \min .$$



Training dataset size is 295!



Vanavar model, training dataset size is 200!



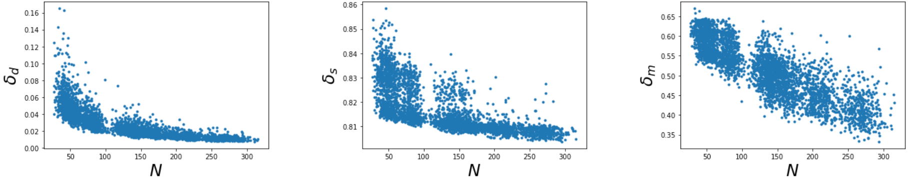
Conclusion

- The NDM-net was presented
- According the provided global sensitivity analysis the most significant parameter affecting the output error is the distance between the source positions
- The results of training on all samples showed that using the new metric, we can suppress the numerical variance for the BP model by 70%, while training on other datasets suppresses the variance by up to 60%

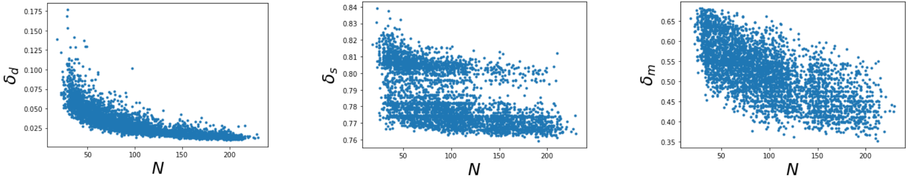
Thank you!

Statistical analysis: for fixed N

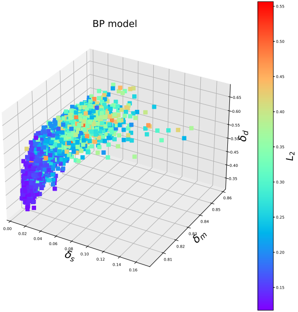
BP model

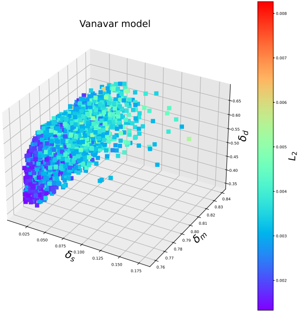


Vanavar model



Statisitcal analysis: for rixed N

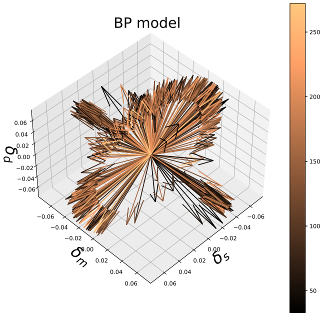


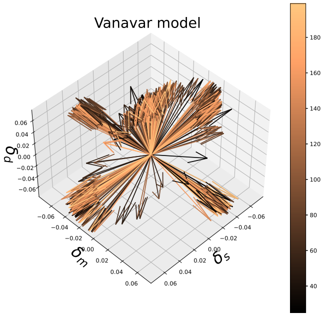


Principal component analysis

$$\mathbf{\Sigma}_{ij}^N = \mathrm{Cov}(\delta_i^N, \delta_j^N) = \mathbb{E}[(\delta_i^N - \mathbb{E}[\delta_i^N])(\delta_j^N - \mathbb{E}[\delta_j^N])],$$

where $\mathbb{E}$ stands for the mean value, δ_i and δ_j can be any of the three distances δ_d, δ_s, or δ_m, and N denotes the number of seismogramms in the considered datasets.





Global Sensitivity Analysis

We consider a continuous scalar function $\phi : \mathbb{R}^p \ rightarrow \mathbb{R}$ defined on the interval $[0, 1]^p$. The input is a $p-$dimensional random variable $X = (X^{(1)}, X^{(2)}, ..., X^{(p)})$ and the output is defined as $Y = \phi(X)$. $X^{(i)}$ is independent.

In the consedered framework, it is thus possible to show that ϕ can be decomposed into elementary functions:

$$\phi(X) = \phi_0 + \sum_{i=1}^{p} \phi_i(X^{(i)}) + \sum_{1 \le i \le j \le p} \phi_{ij}(X^{(i)}, X^{(j)}) + ... + \phi_{1...p}(X^{(1)}, ..., X^{(p)}),$$

where ϕ is assumed to be integrable, ϕ_0 is a constant.

$$Var(Y) = V = \sum_{i=1}^{p} V_i + \sum_{1 \le i \le j \le p} V_{ij} + ... + V_{1,...,p},$$

where

$$V_i = Var(\mathbb{E}(Y|X^{(i)})),$$

$$V_{ij} = Var(\mathbb{E}(Y|X^{(i)}, X^{(j)})) - V_i - V_j, \text{and etc.}$$

Global sensitivity analysis

Sobol sensitivity indices at first order S_i for the $X^{(i)}$ are then defined with

$$S_i = \frac{V_i}{V} = \frac{Var(\mathbb{E}(Y|X^{(i)}))}{Var(Y)}$$

Sensitivity indices at second order S_{ij}:

$$S_{ij} = \frac{V_{ij}}{V}.$$

Global Sensitivity Analysis

$$f(x,y) = f_0 + \frac{\partial f}{\partial x}(x - x_0) + \frac{\partial f}{\partial y}(y - y_0) + o(\dots),$$

$$Var(f) = \int_{(x,y)} (f(x,y) - f_0)^2 dxdy = \int_{(x,y)} [\frac{\partial f}{\partial x}(x - x_0) + \frac{\partial f}{\partial y}(y - y_0)]^2 dxdy$$

$$= q[(\frac{\partial f}{\partial x})^2 L_x^2 + (\frac{\partial f}{\partial y})^2 L_y^2], \quad q > 0$$

$$S_x = \frac{V_x}{Var(f)} = \frac{(\frac{\partial f}{\partial x} L_x)^2}{(\frac{\partial f}{\partial x} L_x)^2 + (\frac{\partial f}{\partial y} L_y)^2}$$

Wave equation

$$
\begin{aligned}
\rho\frac{\partial u_x}{\partial t} &= \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xz}}{\partial z}, \\
\rho\frac{\partial u_z}{\partial t} &= \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z}, \\
\frac{\partial \sigma_{xx}}{\partial t} &= (\lambda + 2\mu)\frac{\partial u_x}{\partial x} + \lambda\frac{\partial u_z}{\partial z} + \hat{f}_{xx}, \\
\frac{\partial \sigma_{zz}}{\partial t} &= \lambda\frac{\partial u_x}{\partial x} + (\lambda + 2\mu)\frac{\partial u_z}{\partial z} + \hat{f}_{zz}, \\
\frac{\partial \sigma_{xz}}{\partial t} &= \mu\frac{\partial u_x}{\partial z} + \mu\frac{\partial u_z}{\partial x} + \hat{f}_{xz},
\end{aligned}
\tag{2}
$$

where $\vec{u} = (u_x, u_z)^T$ is the particle velocity vector, σ_{xx}, σ_{zz}, σ_{xz} are the stress tensor components, $\rho = \rho(x, z)$ is the mass density, $\lambda = \lambda(x, z)$ and $\mu = \mu(x, z)$ are the Lame's parameters, and $\hat{f}_{xx} = f_{xx}(t)\delta(x - x_s)\delta(z - z_s)$, $\hat{f}_{zz} = f_{zz}(t)\delta(x - x_s)\delta(z - z_s)$, $\hat{f}_{xz} = f_{xz}(t)\delta(x - x_s)\delta(z - z_s)$ are the seismic moments tensor. Typically, the point sources are considered, thus $\delta(x - x_s)$ is the Kronecker's delta function, where (x_s, z_s) is the source location coordinates. The time wavelets $f_{xx}(t)$, $f_{zz}(t)$,$f_{xz}(t)$ are defined by a band-limited impulse.

Numerical dispersion

for the considered fourth-order in space second-order in time scheme the phase velocity of the numerical solution is

$$
\begin{aligned}
c^{fd} &= \pm \frac{Nc}{\alpha\pi} arcsin\left(\alpha\sqrt{\hat{k}_1^2 + \hat{k}_2^2}\right), \\
\hat{k}_1 &= \frac{9}{8}\sin\left(\frac{\pi\cos(\beta)}{N}\right) - \frac{1}{24}\sin\left(\frac{3\pi\cos(\beta)}{N}\right), \\
\hat{k}_2 &= \frac{9}{8}\sin\left(\frac{\pi\sin(\beta)}{N}\right) - \frac{1}{24}\sin\left(\frac{3\pi\sin(\beta)}{N}\right),
\end{aligned} \tag{3}
$$

where N is the number of points per wavelength, α is the Courant ratio, β is the angle defining the propagation direction, and c is the true phase velocity. It is clear, that the phase velocity c^{fd} depends on the signal frequency, thus leads to the change of the signal shape and discrepancy in the wave propagation time.

NDM-net

Downsampling and upsampling layers include the convolutional layer with kernel size $(4 \times 4 \times 4)$, activation function $ReLU(\cdot) = \max(0, \cdot)$ for decoder and $LeakyReLU(\cdot) = \max(0, \cdot) + c \min(0, \cdot)$ with negative slope coefficient $c = 0.2$ for encoder.

Parameter	Value
Learning rate	from 0.01 to 0.0001 during epochs
Momentum parameters	$\beta_1 = 0.5, \beta_2 = 0.999$
Batch size	10
The number of epochs	500
Optimization algorithm	Adaptive Momentum (Adam)
Optimality criterion	Mean Absolute Error (MAE)