An encoder-decoder CNN for DAS-to-geophone transformation

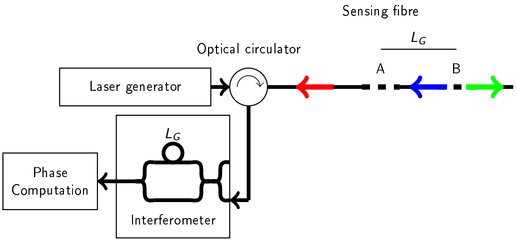
Jorge Monsegny*, Daniel Trad and Don Lawton
jorge.monsegnyparra@ucalgary.ca

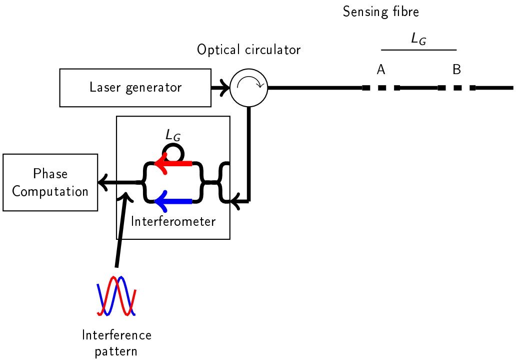
Calgary, December 2 2021

NSERC CRSNG
UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
Department of Geoscience
Carbon Management Canada

Introduction

Distributed acoustic sensing (DAS)



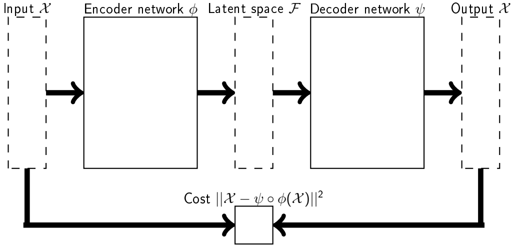


$$\Phi_{AB} = \Phi_A - \Phi_B + \frac{4\pi n \xi \delta l}{\lambda} \tag{1}$$

$$\epsilon_f(s) = \frac{\delta l}{L_G} \tag{2}$$

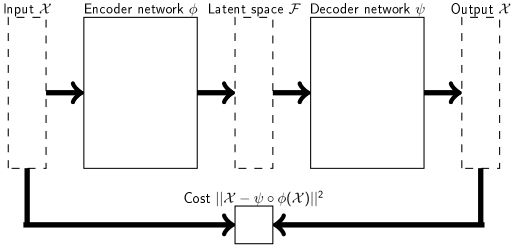
Autoencoder (Encoder-Decoder) Neural Networks

- $\mathcal{X}$ is the input and output space.
- $\mathcal{F}$ is the latent space.
- Encoder:$\phi : \mathcal{X} \rightarrow \mathcal{F}$,
  Decoder:$\psi : \mathcal{F} \rightarrow \mathcal{X}$, such that
  $\phi, \psi = argmin_{\phi,\psi} ||\mathcal{X} - \psi \circ \phi(\mathcal{X})||^2$
- Used to learn *efficient codings*.
- Useful for unlabelled data.



Autoencoder (Encoder-Decoder) Neural Networks

- Undercomplete autoencoders if $dim(\mathcal{F}) \leq dim(\mathcal{X})$.
- Overcomplete autoencoders if $dim(\mathcal{F}) \geq dim(\mathcal{X})$.
- Most approaches try to avoid the autoenoder to learn the identity function.
- Applications in Dimensionality reduction, PCA, Information retrieval, Anomaly detection, Image processing, among others.



Convolutional Neural Networks (CNN)

- $\mathcal{X}_i$, for $i = 1, \ldots, P+1$ are spaces.
- $\phi_i : \mathcal{X}_i \to \mathcal{X}_{i+1}$ is a convolutional layer.
- For $\vec{\boldsymbol{r}}^T = (\vec{\boldsymbol{r}}^1, \ldots, \vec{\boldsymbol{r}}^N)^T \in \mathcal{X}_i$:

$$\phi_i(\vec{\boldsymbol{r}}) = a_i\left(\begin{bmatrix} F_i^{11} & \cdots & F_i^{1N} \\ \vdots & \ddots & \vdots \\ F_i^{M1} & \cdots & F_i^{MN} \end{bmatrix}\begin{bmatrix} \vec{\boldsymbol{r}}^1 \\ \vdots \\ \vec{\boldsymbol{r}}^N \end{bmatrix} + \begin{bmatrix} \vec{\boldsymbol{b}}^1 \\ \vdots \\ \vec{\boldsymbol{b}}^N \end{bmatrix}\right) = \begin{bmatrix} a_i(\sum_j F_i^{1j} * \vec{\boldsymbol{r}}^j + \vec{\boldsymbol{b}}^1) \\ \vdots \\ a_i(\sum_j F_i^{Mj} * \vec{\boldsymbol{r}}^j + \vec{\boldsymbol{b}}^M) \end{bmatrix} \tag{3}$$

- An P layered CNN is:

$$\mathcal{X}_1 \xmapsto{\phi_1} \mathcal{X}_2 \xmapsto{\phi_2} \mathcal{X}_3 \cdots \mathcal{X}_P \xmapsto{\phi_P} \mathcal{X}_{P+1} \tag{4}$$

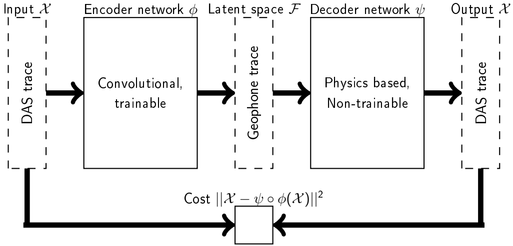
- Convolution makes network connectivity local.

Methods

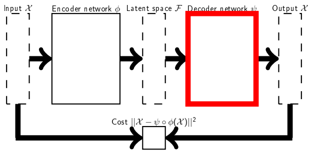
DAS-Geophone-DAS Encoder-Decoder Neural Network

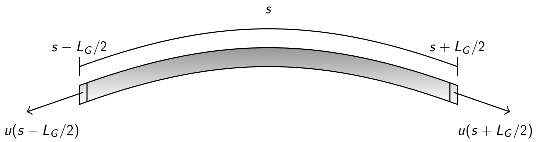
- $\mathcal{X}$ is the space of DAS traces.
- $\mathcal{F}$ is the space of the geophone traces.
- ϕ is a CNN.
- ψ is fixed and physics-based.
- Training objective: $\phi = argmin_{\phi}||\mathcal{X} - \psi \circ \phi(\mathcal{X})||^2$
- At the end ϕ does the desired DAS to geophone transformation:

$$\phi(\text{das trace}) = \text{geophone trace}$$



Geophone to DAS Decoder physical model



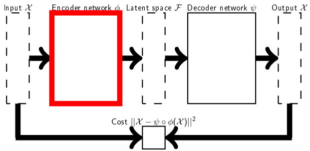


$$\delta l(s) = u(s + L_G/2) - u(s - L_G/2) \tag{5}$$

$$\dot{\epsilon}_f(s) = \frac{1}{L_G}(v(s + L_G/2) - v(s - L_G/2)) \tag{6}$$

$$\begin{bmatrix} \dot{\epsilon}_f(s_1) \\ \vdots \\ \dot{\epsilon}_f(s_i) \\ \vdots \\ \dot{\epsilon}_f(s_M) \end{bmatrix} = \frac{1}{L_G} \begin{bmatrix} -1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & & & & \vdots \\ 0 & \cdots & 0 & -1 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} v_z(s_{1-N/2}) \\ \vdots \\ v_z(s_i) \\ \vdots \\ v_z(s_{M+N/2}) \end{bmatrix} \tag{7}$$

DAS to Geophone Encoder CNN



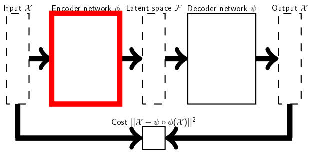
First CNN layer is the vector of 1D filters $F_1^T = (F_1^1, \ldots, F_1^M)^T$ and the activation function a_1:

$$\vec{d} \xmapsto{F_1} \begin{bmatrix} F_1^1 \\ F_1^2 \\ \vdots \\ F_1^M \end{bmatrix} \vec{d} = \begin{bmatrix} F_1^1 * \vec{d} \\ F_1^2 * \vec{d} \\ \vdots \\ F_1^M * \vec{d} \end{bmatrix} = \begin{bmatrix} \vec{r}^1 \\ \vec{r}^2 \\ \vdots \\ \vec{r}^M \end{bmatrix} \xmapsto{a_1} \begin{bmatrix} a_1(\vec{r}^1) \\ a_1(\vec{r}^2) \\ \vdots \\ a_1(\vec{r}^M) \end{bmatrix} \tag{8}$$

Second CNN layer is the vector of 1D filters $F_2 = (F_2^1, \ldots, F_2^M)$ and the activation function a_2:

$$\begin{bmatrix} a_1(\vec{r}^1) \\ a_1(\vec{r}^2) \\ \vdots \\ a_1(\vec{r}^M) \end{bmatrix} \xmapsto{F_2} \begin{bmatrix} F_2^1 \cdots F_2^M \end{bmatrix} \begin{bmatrix} a_1(\vec{r}^1) \\ a_1(\vec{r}^2) \\ \vdots \\ a_1(\vec{r}^M) \end{bmatrix} = \sum_{j=1}^{M} F_2^j * a_1(\vec{r}^j) \xmapsto{a_2} a_2\left(\sum_{j=1}^{M} F_2^j * a_1(\vec{r}^j)\right) = \vec{g} \tag{9}$$

DAS to Geophone Encoder CNN



- Activation function is $a_1 = a_2 = tanh$ because the output are traces with negative an positive values.
- Kernel initializer is Xavier normal initializer.
- Geophone trace is the same length as DAS trace so convolution padding is *same*.
- Bias was not used.

DAS to Geophone Encoder CNN

- Regularization can be applied to kernel weights:

$$\epsilon_1 |F_1|_p^p + \epsilon_2 |F_2|_p^p$$

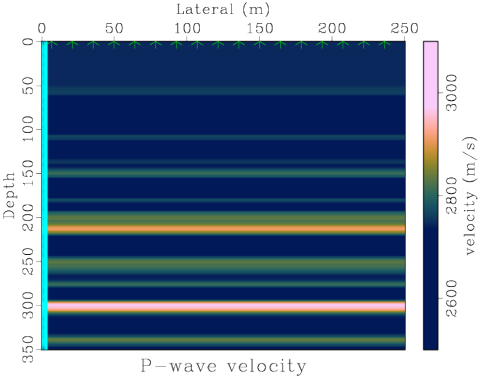
- To layer outputs:

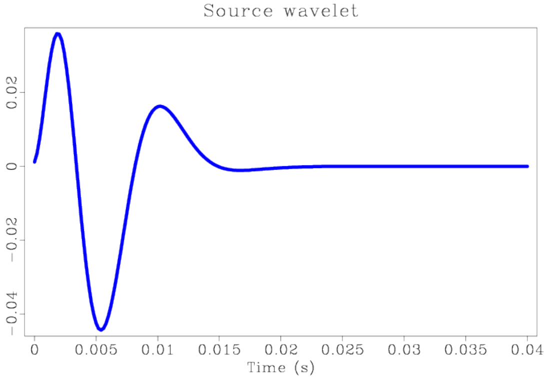
$$\epsilon_1 |F_1 \vec{\boldsymbol{d}}|_p^p + \epsilon_2 |F_2 \vec{\boldsymbol{r}}|_p^p$$

- (It can also be applied to bias vectors).

Synthetic data

Acquisition

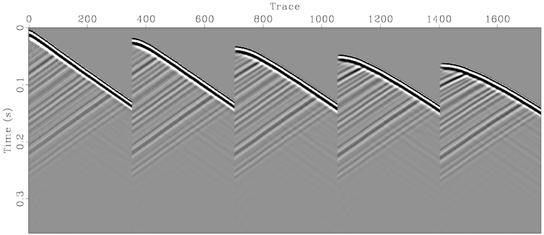




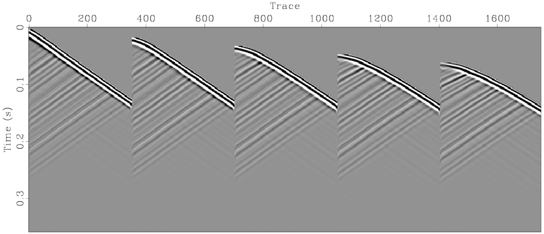
Geophone and DAS data

- Vertical particle velocity acoustic modelling.
- DAS obtained from geophone using the neural network decoder with $L_G = 10$m:

$$\dot{\epsilon}_f(s) = \frac{1}{L_G}(v_z(s+L_G/2)-v_z(s-L_G/2))$$



Geophone

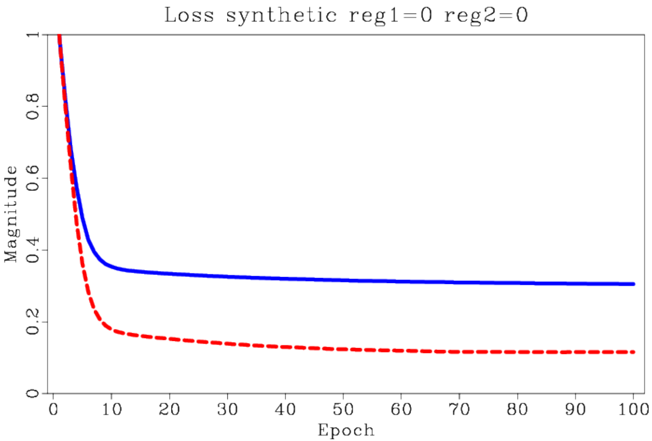


DAS

Results without regularization

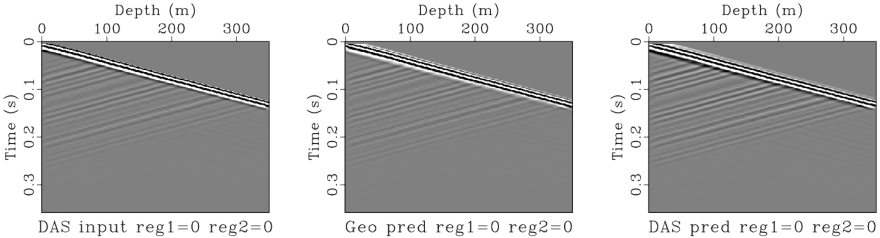
Parameters:

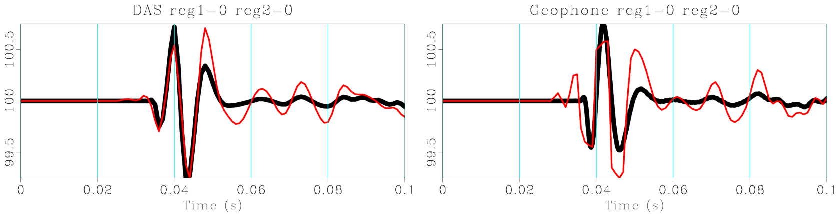
- Algorithm: Adam.
- Number of epochs: 100
- Training decimation: 10
- Learning rate: 0.001
- Batch size: 32
- Validation split: 0.5
- Number of filters: 20
- Filter length: $2L_G$
- No regularization



Continuous line is training MSE. Dashed is validation MSE.

Results without regularization

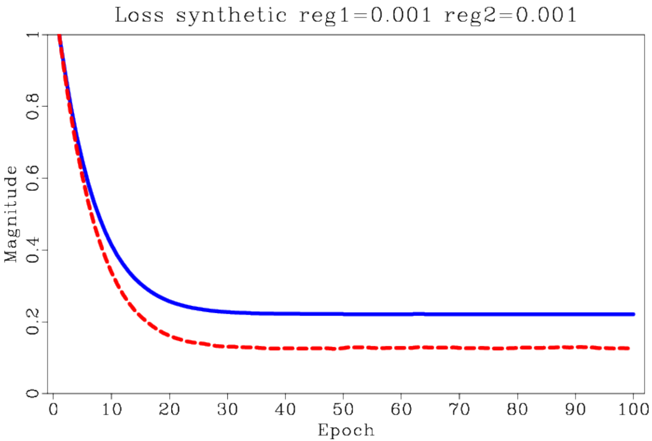




Regularization reg1=0.001 and reg2=0.001

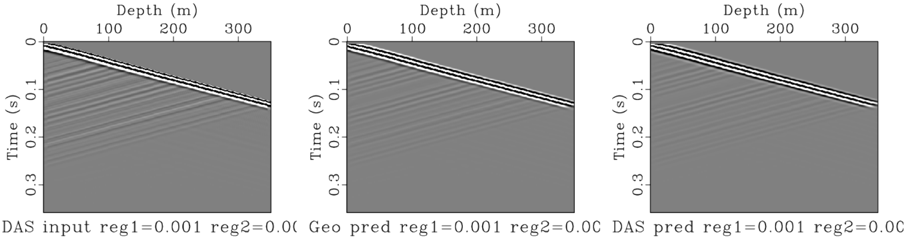
Parameters:

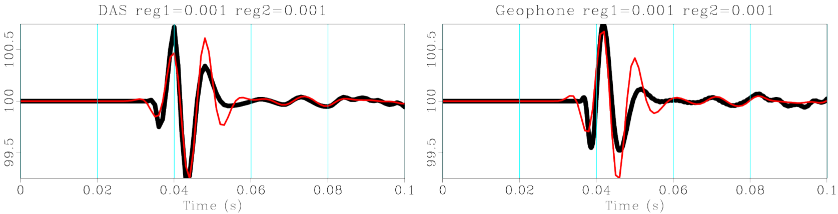
- Idem, except
- L2 kernel regularization: reg1=0.001 and reg2=0.001



Continuous line is training MSE. Dashed is validation MSE.

Results reg1=0.001 and reg2=0.001

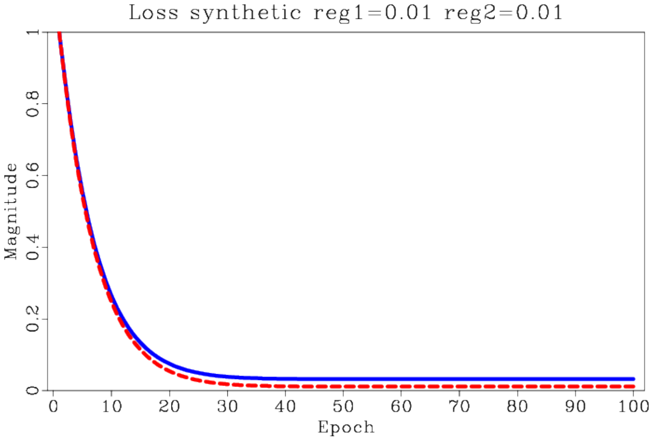




Regularization reg1=0.01 and reg2=0.01

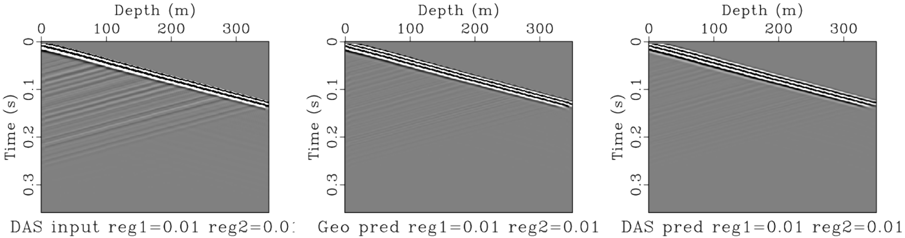
Parameters:

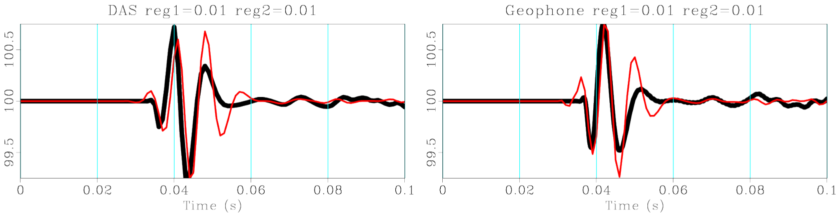
- Idem, except
- L2 kernel regularization: reg1=0.01 and reg2=0.01



Continuous line is training MSE. Dashed is validation MSE.

Results reg1=0.01 and reg2=0.01

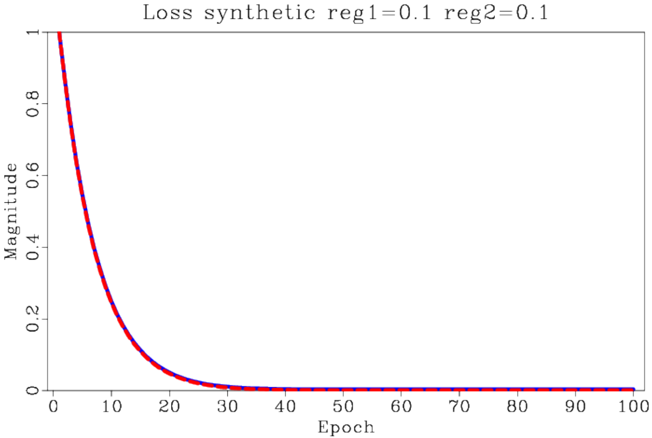




Regularization reg1=0.1 and reg2=0.1

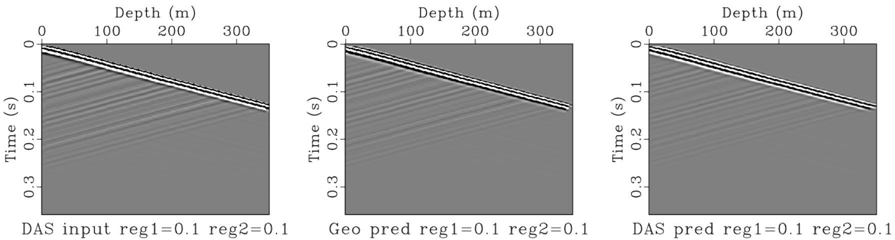
Parameters:

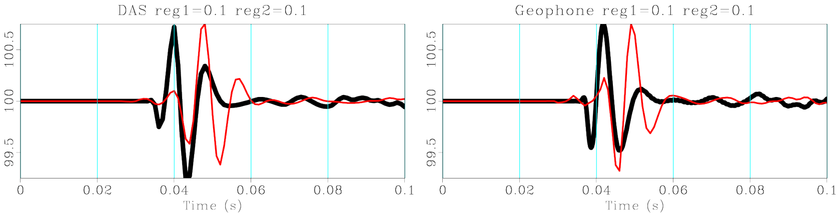
- Idem, except
- L2 kernel regularization: reg1=0.1 and reg2=0.1



Continuous line is training MSE. Dashed is validation MSE.

Results reg1=0.1 and reg2=0.1

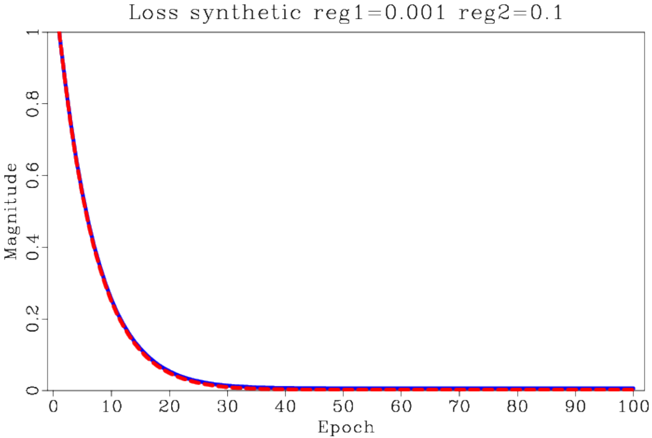




Regularization reg1=0.001 and reg2=0.1

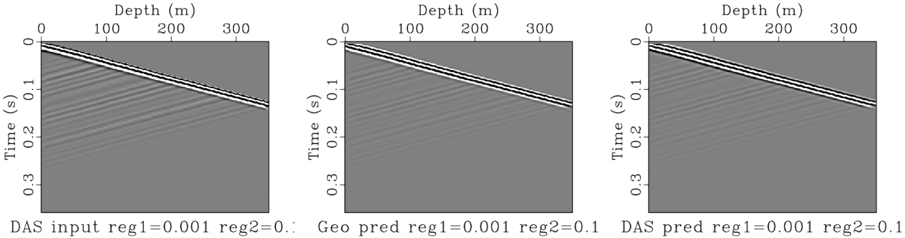
Parameters:

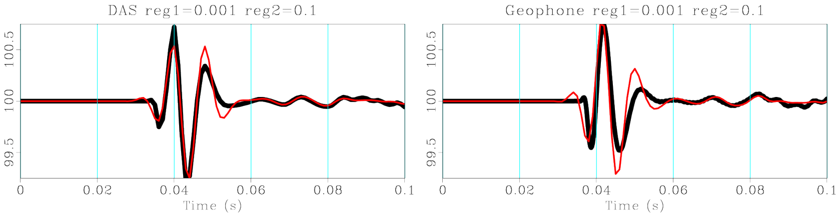
- Idem, except
- L2 kernel regularization: reg1=0.001 and reg2=0.1



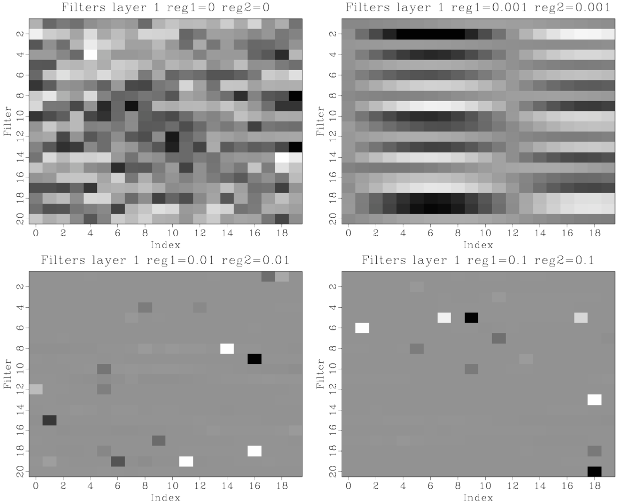
Continuous line is training MSE. Dashed is validation MSE.

Results reg1=0.001 and reg2=0.1

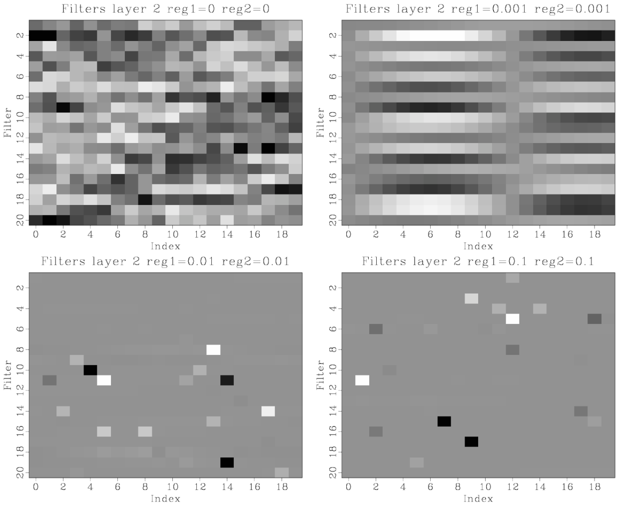




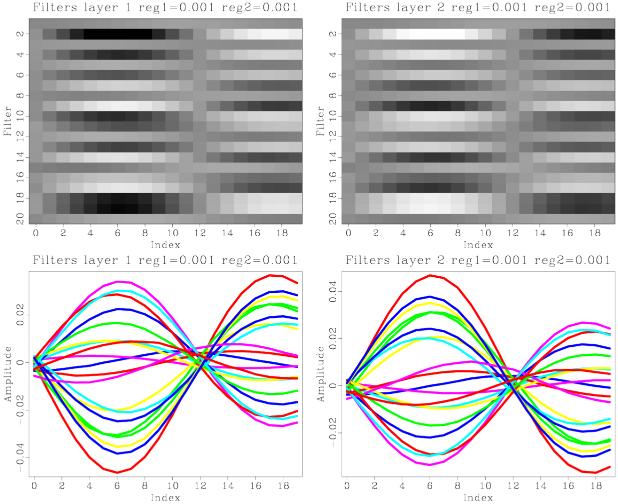
Layer 1 convolution filters



Layer 2 convolution filters

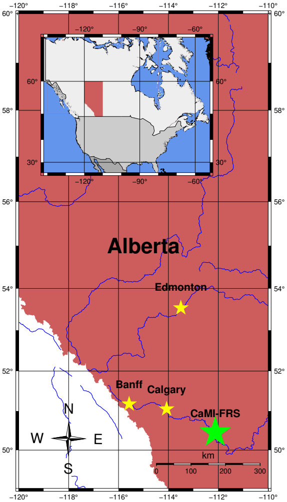


Convolution filters for reg1=0.001 and reg2=0.001

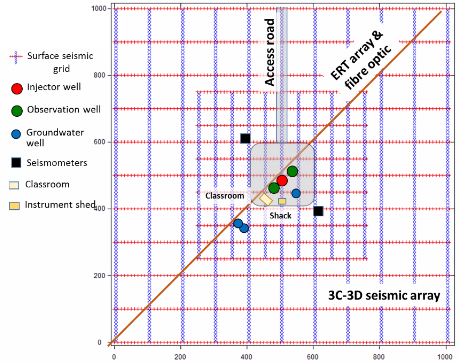


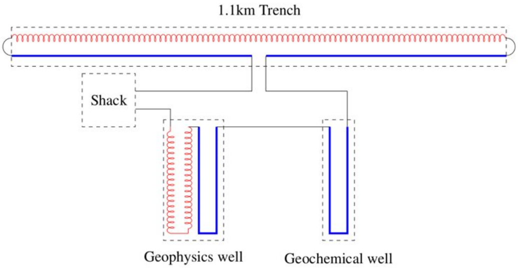
Field data

Containment and Monitoring Institute Field Research Station (CaMI-FRS)

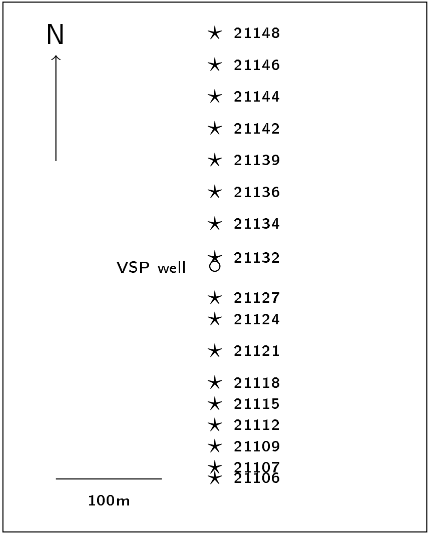


Containment and Monitoring Institute Field Research Station (CaMI-FRS)

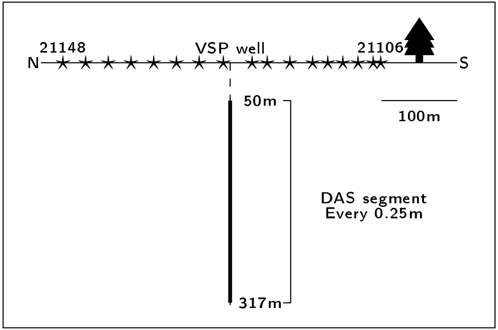




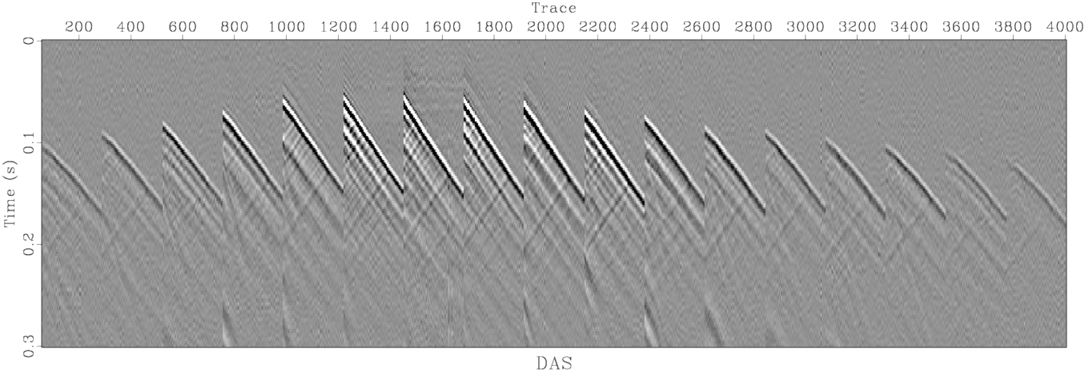
Field data



- 17 shot gathers.
- Source was IVI EnviroVibe with linear sweep 10-150Hz.
- 338m DAS fibre.
- Gauge length is 10m with traces every 0.25m



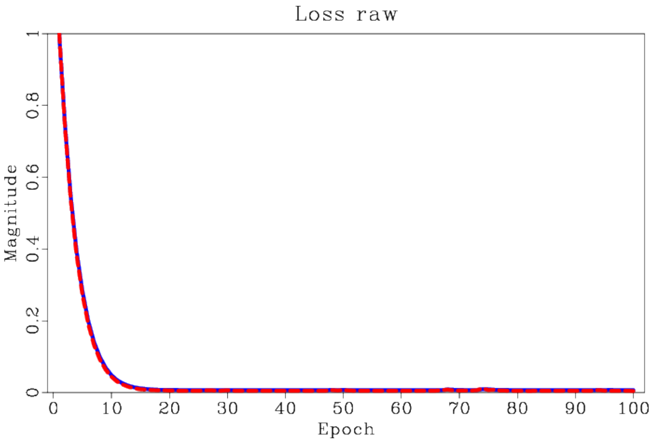
Raw data


Trace
200
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
3000
3200
3400
3600
3800
4000
Time (s)
0
0.1
0.2
0.3
DAS

Training and loss

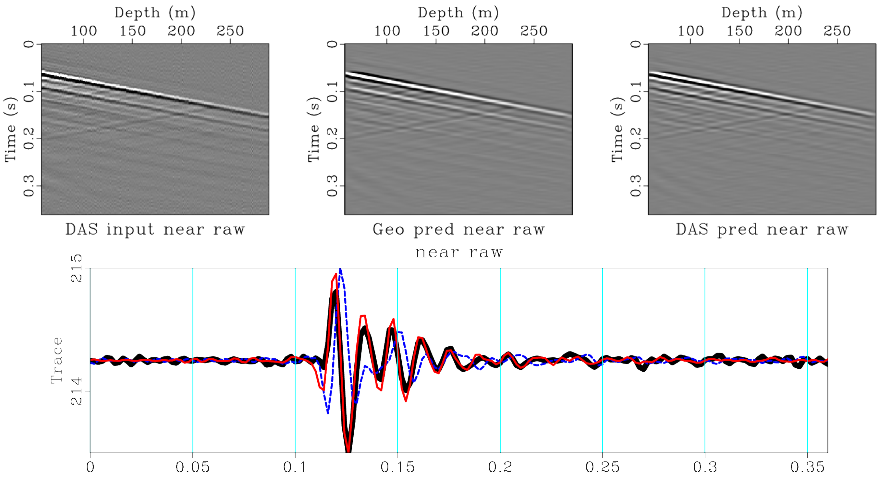
Parameters:

- Number of epochs: 100
- Training decimation: 10
- Learn rate: 0.001
- Batch size: 32
- Validation split: 0.5
- Number of filters: 20
- Filter length: $2L_G$
- L2 kernel regularization: reg1=0.001 and reg2=0.1



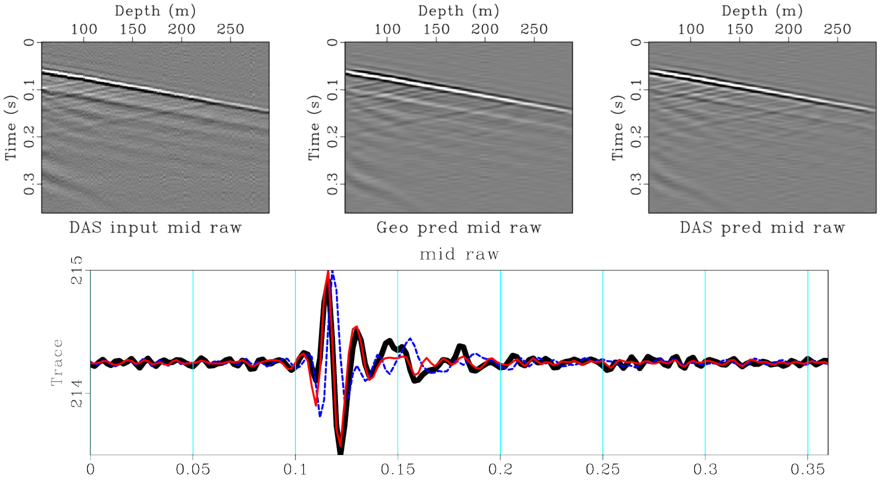
Continuous line is training MSE. Dashed is validation MSE.

Near offsets



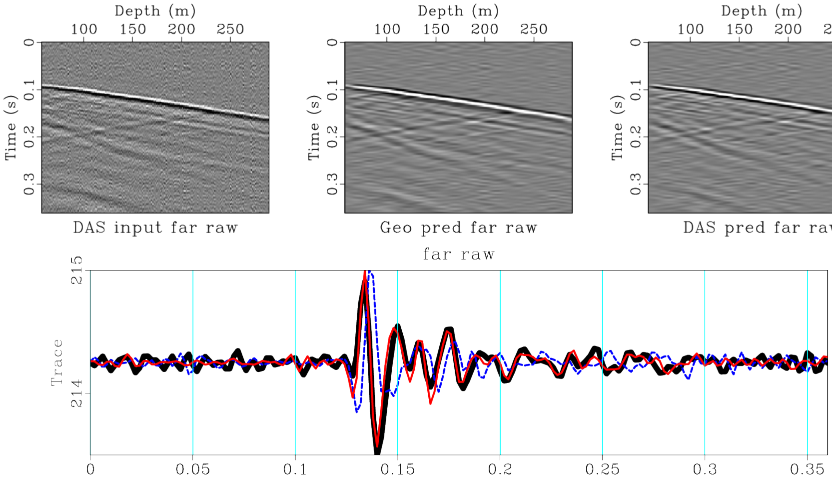
Thick line is input DAS, thin line is predicted DAS and dashed line is predicted geophone.

Mid offsets



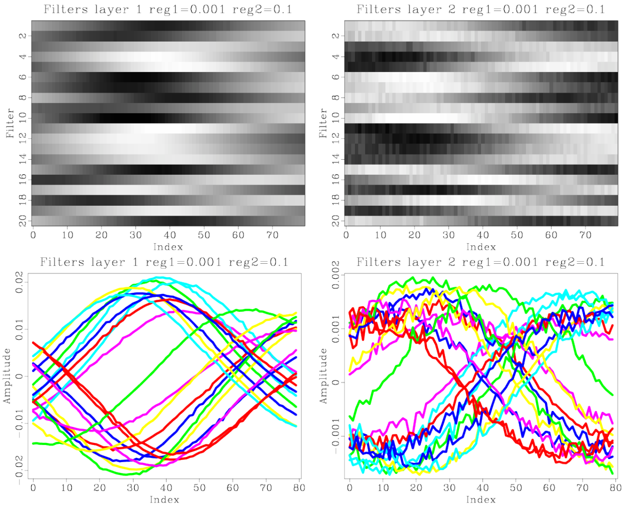
Thick line is input DAS, thin line is predicted DAS and dashed line is predicted geophone.

Far offsets



Thick line is input DAS, thin line is predicted DAS and dashed line is predicted geophone.

Convolution filters



Summary

Summary

- The DAS-to-geophone encoder decoder CNN is an example of a physics-based unsupervised neural network.
- Regularization in the kernel coefficients of the neural network were needed to adjust the relative amplitudes of the output DAS trace and the geophone trace in the latent space.
- Some regularization values produced convolution filters with sinusoidal shapes.
- The DAS-to-geophone encoder-decoder CNN was successful when tested with the synthetic data. More work is needed to examine its results with the field data.
- The physics part of the neural network in the decoder can be improved and the encoder will adjust itself with the neural network training.

Acknowledgements

- CREWES Sponsors.
- University of Calgary Global Research Initiative.
- The Canada First Research Excellence Fund.
- The Containment and Monitoring Institute.
- CREWES students, faculty and staff.