Compressive Sensing - Machine Learning combined for joint location and moment tensor estimation: a performance analysis

Introduction

Source location and source mechanism are fundamental parameters for the study of natural and induced seismicity (Aki and Richards, 2009). Their joint optimization presents challenges, on one hand, because global optimization approaches can be computationally expensive, and on the other, because local optimization methods suffer the known dangers of being caught in local minima.

Compressive sensing (CS) has been proposed as an alternative for fast global optimization of source location and moment tensor in datasets with large numbers of recording channels (Vera Rodriguez and Sacchi, 2017). The concept permits processing of the seismic records in compressed form, reducing thereby response time. The output of this Compressed Seismic Sensing (CoSSen) method is the location and moment tensor of detected events. However, CoSSen faced practical limitations, for example, in that the extraction of the source parameters required the use of time-consuming iterative solvers.

With the incorporation of machine learning, an upgraded version of the CS method, i.e., deepCoSSen, has showed promising results in simulations with synthetic data (Vera Rodriguez, 2021). Replacing the iterative solver in CoSSen with a deep neural network (DNN) books most of the computational effort to the training stage of the DNN, which afterwards can be used to obtain predictions of source parameters in a way that is consistent with the fast response objective of deepCoSSen. This work presents further simulation results that investigate the performance of deepCoSSen under controlled scenarios. These results are important in that they provide an idea of the requirements for the implementation with real data.

Theory

CS is a novel sampling paradigm that alleviates limitations posed by the Nyquist-Shannon sampling theorem (Donoho, 2006). Two important conditions for CS a signal are that 1) the signal must possess a sparse representation under an overcomplete dictionary of basis functions, and 2) that the compression operator must display restricted isometry (RIP, Candes et al., 2006).

If we set a grid over a monitoring region of interest and assume that every node in the grid is a virtual source, then, the joint location and moment tensor inversion problem can be casted as a block-sparse representation problem (Vera Rodriguez et al., 2012). This can be represented as a linear system with \( \mathbf{u} = \mathbf{Gm} \), where \( \mathbf{u} \) is a vector formed with the concatenation of the seismic records, \( \mathbf{G} \) is a dictionary of Green functions between each virtual source and the seismic receivers, and \( \mathbf{m} \) is a vector formed with the concatenation of the moment tensors of all the virtual sources. The sparsity of \( \mathbf{m} \) depends on the observations \( \mathbf{u} \) containing the arrivals of, ideally, only one seismic source. Using this parameterization, the first condition cited above is fulfilled so that CS can be applied to the seismic records in \( \mathbf{u} \).

For the second condition, we can select a compression matrix for which the RIP has been corroborated. Assuming that this matrix is \( \mathbf{\Phi} \in \mathbb{R}^{N_c \times N_u} \), the compression of the vector \( \mathbf{u} \in \mathbb{R}^{N_u \times 1} \) is represented with \( \mathbf{\Phi u} = \mathbf{\Phi Gm} \). The dimensions of \( \mathbf{\Phi} \) are defined as \( N_c << N_u \) such that its application results in a compression down to \( C_\Phi = 100 \frac{N_c}{N_u} \% \) of the original size of the observations. The recovery of the solution vector \( \mathbf{m} \) is then accomplished with a DNN to formulate the faster response method deepCoSSen (Vera Rodriguez, 2021). The following experiments report results with synthetic data that investigate the performance of this approach.

Examples

The setting for data simulations is a block with dimensions of 71 cm \( \times \) 71 cm \( \times \) 91 cm, instrumented with 38, one-component sensors distributed over all of its faces (Figure 1a). This corresponds with a laboratory experiment for which the real data will be used in future validation work. Synthetics were modeled using analytical solutions in a homogeneous medium with compressional (P) and shear (S) velocities randomly sampled from Gaussian distributions with means of \( v_P = 2750 \) m/s and \( v_S = 1800 \) m/s, respectively. The standard deviation in both cases was 3% of the mean. In this way, uncertainties about propagation velocities in the rock were taken into account.

The solution space for source mechanisms considered a general dislocation model, in which a source type is defined by the angles of strike, dip, rake and \( \alpha \) (\( \alpha \) measures the deviation of the displacement vector from the plane of the dislocation). The labels for the data examples incorporated normalized moment tensors.
Figure 1 (a) Setup used for data simulations. Arrows point in the direction of the sensors’ components. Colors are used to help discriminate the side of the block where a sensor is located. (b) Schematic of the compression of one synthetics to create a data example with \( C_\Phi = 12.5\% \). The compression matrix \( \Phi \) is formed with independent and identically distributed Gaussian samples.

the prediction of source parameters, the moment tensors were decomposed back to dislocation angles using the biaxial decomposition (Chapman and Leaney, 2012). The block was sampled at random to generate locations for the modeling of synthetics. Coordinates were transformed to a non-dimensional space with limits around -1 to 1 before adding them to the data labels. The parameters of the transform were stored and used to return predictions from the DNN to their original coordinate space. Three different simulation experiments were performed:

**Experiment 1.** Training sets with 50k, 100k, 200k and 300k data examples were prepared (k represents the 1000 multiplier). Each dislocation angle for one synthetics was randomly sampled from two possible Gaussian distributions. The mean values of the distributions were \( s_1 : \{135^\circ, 75^\circ, 90^\circ, 0^\circ\} \) and \( s_2 : \{90^\circ, 75^\circ, 90^\circ, 30^\circ\} \) for angles of strike, dip, rake and \( \alpha \), respectively. In all cases, the standard deviation was 5\(^{\circ}\). The source function was a Brune pulse with corner frequency of 10 kHz. The sampling rate was 1 \( \mu \)s. The synthetics were cut at lengths of 1028 samples and compressed at values of \( C_\Phi = 0.8\%, 3.1\% \) and 12.5\% to generate a total of 12 training sets of different sizes and compression (Figure 1b). A deepCoSSen system was created using each of the training sets. During training, the DNNs were tested every five epochs to generate predictions of location and dislocation angles using separate test sets of 10k data examples.

**Experiment 2.** Additional training sets of 300k, 500k and 700k were generated following the same procedure as in Experiment 1 but with a Brune pulse of 160 kHz corner frequency and sampling rate of 0.4 \( \mu \)s. The compression used was \( C_\Phi = 0.8\% \).

**Experiment 3** Additional training sets of 300k, 500k and 700k were generated, keeping corner frequency, sampling rate and compression at 10 kHz, 1 \( \mu \)s and 0.8\%, respectively. On the other hand, the solution space for source type was expanded with an additional set of angle distributions \( s_3 : \{135^\circ, 75^\circ, 0^\circ, 0^\circ\} \). The standard deviations were also increased from 5\(^{\circ}\) to 15\(^{\circ}\) in all cases.

**Deep Neural Network architectures.** The architecture of the DNN for each compression value had to be changed as a result of the changes in input dimensions. An example of the network architectures that were used is
presented in Vera Rodriguez (2021). They are deep hybrid neural networks consisting of convolutional and pooling layers, and that transition into dense layers as they prepare to give an output. Activation functions were rectified linear unit (relu) in all cases, except for the output layer, which did not require it.

Results

While the training error of the DNNs followed a stably declining trajectory during training, the validation error flattened around epoch 70 in all the investigated cases (e.g., Figure 2a). This flattening of the learning rate was also corroborated by the location and dislocation angle prediction errors evaluated over the testing sets (e.g., Figure 2b and c). Therefore, the results presented in the following correspond to DNNs trained up to the 70th epoch. The prediction error for location was estimated as the euclidean distance from the known location. The dislocation angles error was estimated as $\Delta_i = |\Delta_i^{\text{strike}}| + |\Delta_i^{\text{dip}}| + |\Delta_i^{\text{rake}}| + |\Delta_i^{\alpha}|$, where $\Delta_i$ represents a difference between predicted and true values for the $i$th example in one of the 10k-examples test sets.

![Figure 2](a) Training (solid lines) and validation (dashed lines) errors obtained with training sets of sizes 50k (blue), 100k (gray), 200k (red) and 300k (green) examples for deepCoSSen systems with $C_{\Phi} = 3.1\%$. Panels (b) and (c) present corresponding prediction errors and standard deviations evaluated every 5 epochs using a test set of 10k examples.

<table>
<thead>
<tr>
<th>$C_{\Phi}$</th>
<th>Type of error</th>
<th>Size #</th>
<th>50k</th>
<th>100k</th>
<th>200k</th>
<th>300k</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8 %</td>
<td>L $\pm \delta L$ (mm)</td>
<td>15.4 $\pm$ 9.6</td>
<td>10.6 $\pm$ 6.6</td>
<td>7.7 $\pm$ 4.9</td>
<td>6.3 $\pm$ 3.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A $\pm \delta A$ (°))</td>
<td>2.2 $\pm$ 1.6</td>
<td>1.6 $\pm$ 1.0</td>
<td>1.2 $\pm$ 0.8</td>
<td>1.0 $\pm$ 0.9</td>
<td></td>
</tr>
<tr>
<td>3.1 %</td>
<td>L $\pm \delta L$ (mm)</td>
<td>11.7 $\pm$ 7.0</td>
<td>8.1 $\pm$ 4.8</td>
<td>6.0 $\pm$ 3.6</td>
<td>5.2 $\pm$ 3.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A $\pm \delta A$ (°))</td>
<td>1.7 $\pm$ 1.5</td>
<td>1.2 $\pm$ 1.5</td>
<td>0.9 $\pm$ 1.2</td>
<td>0.8 $\pm$ 1.5</td>
<td></td>
</tr>
<tr>
<td>12.5 %</td>
<td>L $\pm \delta L$ (mm)</td>
<td>12.4 $\pm$ 7.3</td>
<td>8.3 $\pm$ 5.0</td>
<td>6.4 $\pm$ 3.7</td>
<td>5.5 $\pm$ 3.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A $\pm \delta A$ (°))</td>
<td>1.6 $\pm$ 1.1</td>
<td>1.2 $\pm$ 0.9</td>
<td>0.9 $\pm$ 1.0</td>
<td>0.8 $\pm$ 1.6</td>
<td></td>
</tr>
</tbody>
</table>

The results from Experiment 1 (Table 1) confirm that the replacement of the iterative solver with the DNN does not change the general behavior of the deepCoSSen system. For instance, reducing $C_{\Phi}$ increases prediction errors, which amounts to information loses during compression. Nevertheless, the loses are not considered significant compared to the amount of data compression that is achieved. Moreover, the degradation appears to start somewhere with $C_{\Phi} < 3.1\%$ as the results with $C_{\Phi} \geq 3.1\%$ are both comparable.

Increasing the size of the training set also improved prediction errors. In this respect, training sets on the order of 300k examples appeared to converge to the resolution limits of the corresponding deepCoSSen versions (see Figure 2 for an example). The resolution limit for location is close to the location uncertainties reported for real data in similar experiments (e.g., Vera Rodriguez et al., 2017). Notice here that the error cannot be zero even if the data are noiseless synthetics; this is because of the uncertainties introduced by the sampling of velocities from Gaussian distributions during the generation of data examples.

Increasing the frequency content in the data increased prediction errors (i.e., Experiment 2, Table 2). This results from a reduction in waveform similarity amongst nearby sources, which requires larger training sets to feed the DNN with a comprehensive overview of the solution space during training. In this experiment with 160kHz frequency content, increasing the size of the training set up to 700k examples is not yet enough to achieve the prediction error levels observed at 10kHz frequency content with the same compression and a
training set with 300k examples (compare results in Tables 1 and 2). The increase in frequency content in the observations can be seen as equivalent to an increase of the solution space for source location.

### Table 2 Results for mean location errors (\(L\)), mean errors in dislocation angles (\(A\)), and their standard deviations. Errors were estimated with test sets of 10k examples. \(C_\Phi = 0.8\%\) in both experiments.

<table>
<thead>
<tr>
<th>Type of error</th>
<th>Size of training set</th>
<th>300k</th>
<th>500k</th>
<th>700k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 2</td>
<td>(L \pm \delta L) (mm)</td>
<td>26.3 ± 13.7</td>
<td>22.1 ± 11.5</td>
<td>20.0 ± 10.3</td>
</tr>
<tr>
<td></td>
<td>(A \pm \delta A) (°)</td>
<td>2.6 ± 1.3</td>
<td>2.3 ± 1.3</td>
<td>2.2 ± 1.4</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>(L \pm \delta L) (mm)</td>
<td>11.3 ± 7.4</td>
<td>9.0 ± 5.8</td>
<td>8.1 ± 5.3</td>
</tr>
<tr>
<td></td>
<td>(A \pm \delta A) (°)</td>
<td>4.9 ± 11.9</td>
<td>4.5 ± 12.1</td>
<td>4.6 ± 12.4</td>
</tr>
</tbody>
</table>

As can be expected from the previous results, the increase in size of the solution space for the source mechanism (Experiment 3) also increased prediction errors for same sizes of the training set (compare results in Tables 1 and 2). In this case, adding one more distribution for the sampling of dislocation angles and increasing the standard deviations required training sets on the order of 700k examples to come closer to the prediction errors for location observed in Experiment 1 for the same compression but smaller training set of 300k examples. Dislocation angles, however, remained with significant prediction errors, mainly concentrated on strikes and rakes. These results suggests that the learning of the DNN is robust to recognise the (compressed) pattern of moveouts in the synthetics to determine location, although it still requires more data examples to learn the distribution of (compressed) amplitudes that corresponds to different dislocation solutions.

### Conclusions

Compressed seismic sensing (CoSSen) opens up new possibilities for fast analysis of the records from dense passive seismic monitoring networks. The results presented here demonstrate that the projection of the seismic data onto a compressed space preserves the relative distances between the transformed signals in the new space. In other words, the transformed signals retain unique patterns that can be learned by neural networks (e.g., deepCoSSen). The main benefit of this is that large volumes of continuous data, for example, those produced by DAS, can be quickly scanned to detect time periods of interest where the fully sampled data can then be analysed in more detail. The application presented here goes a step farther by reporting source parameters of the detected seismic events. Nevertheless, other applications of the protocol can be devised to output only event detections or source locations. Further work related to the application presented here entails the implementation with a continuous data flow rather than individual triggers, and the analysis of the impact of noise in prediction errors and permissible compression.

### Acknowledgements

This work is supported by NORSAR institutional funds.

### References

Vera Rodriguez, I. [2021] Towards fast DAS passive seismic monitoring combining Compressive Sensing with a Deep Learning decoder. 2nd EAGE Workshop on Fiber Optic Sensing, EAGE, Online event.