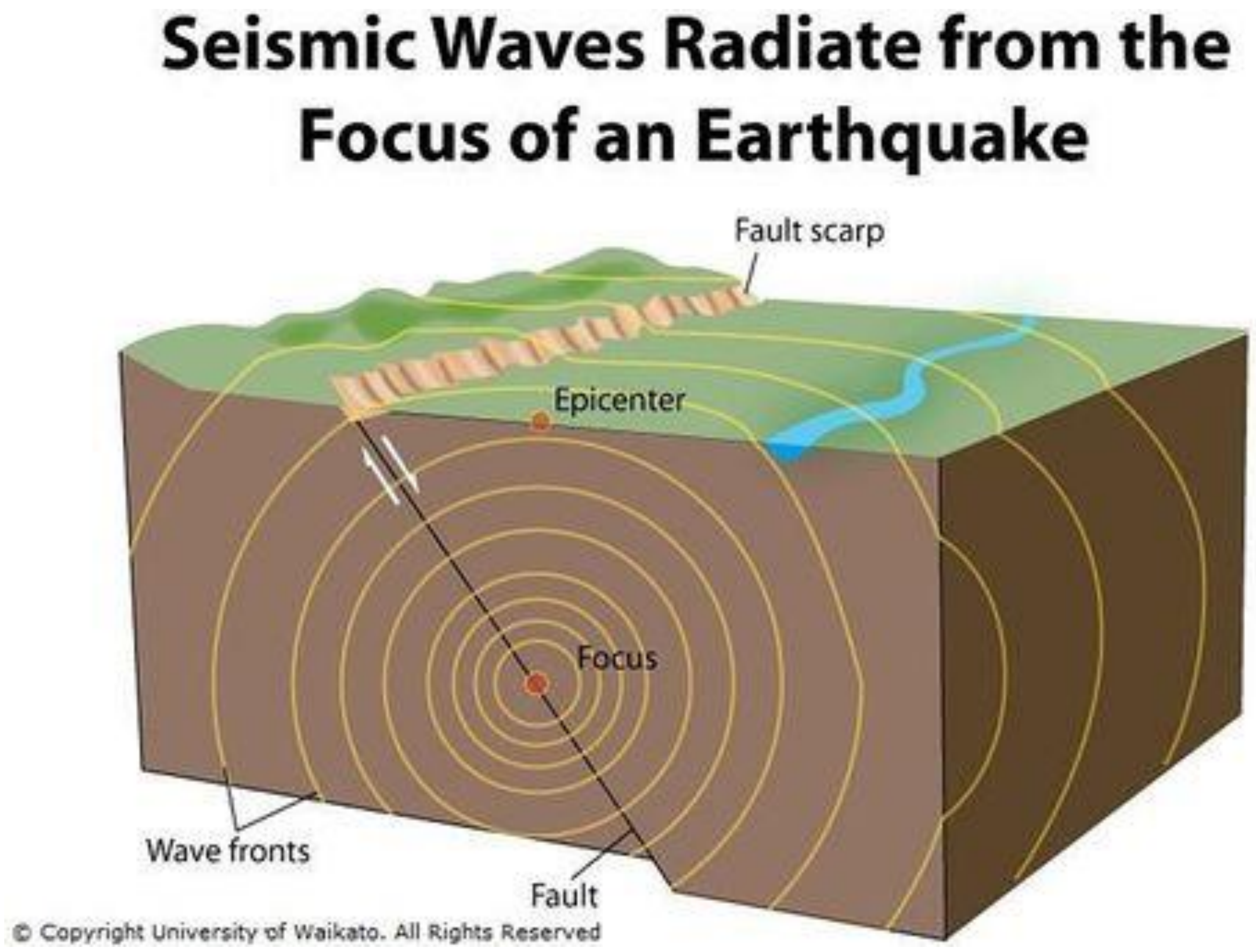
Neural network for seismic data (Data reduction을 중심으로)

서울대 물리탐사연구실
강혁

Index

Seismic data

http://sciencelearn.org.nz/Contexts/Earthquakes/Sci-Media/Images/Seismic-waves


Seismic Waves Radiate from the Focus of an Earthquake
Fault scarp
Epicenter
Focus
Wave fronts
Fault
© Copyright University of Waikato. All Rights Reserved

Seismic data

https://str.llnl.gov/str/Walter.html

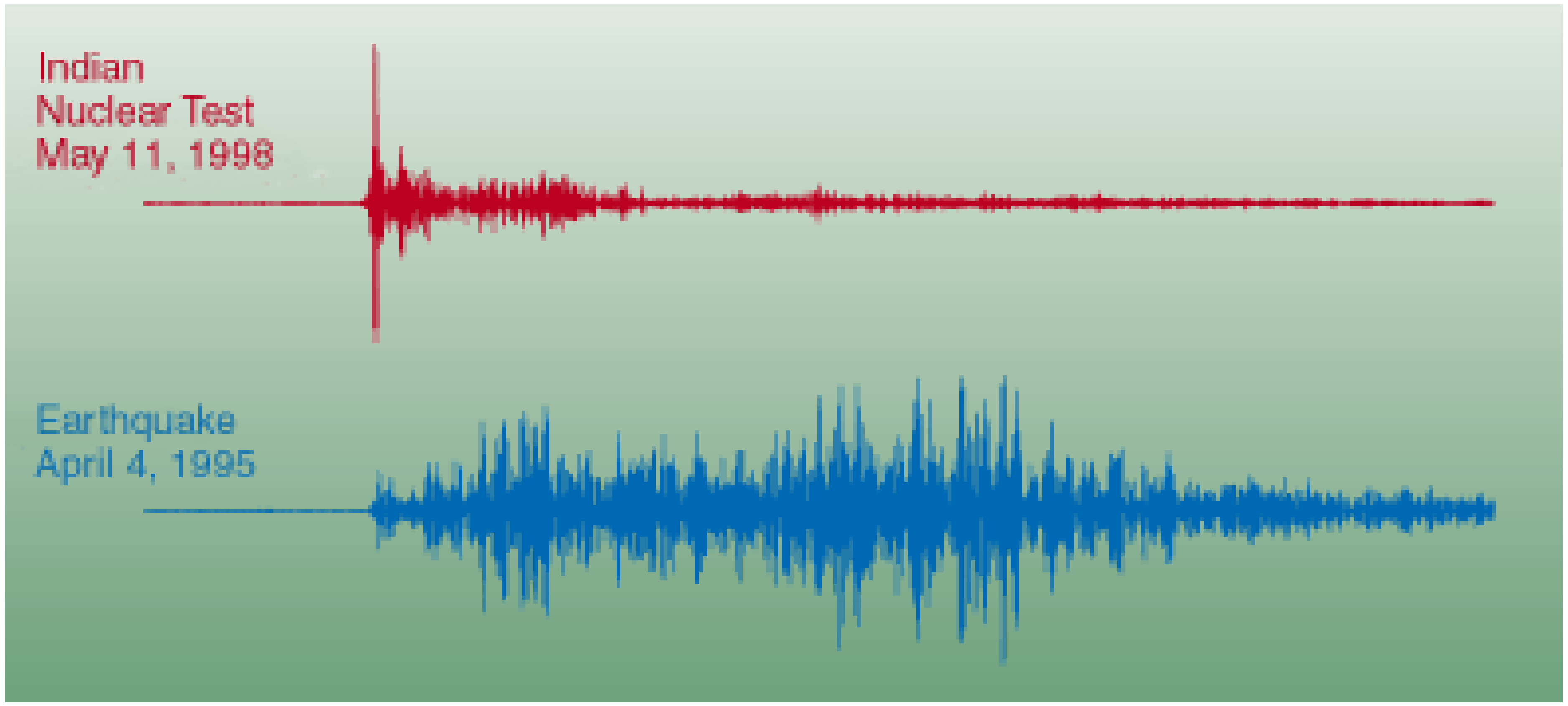
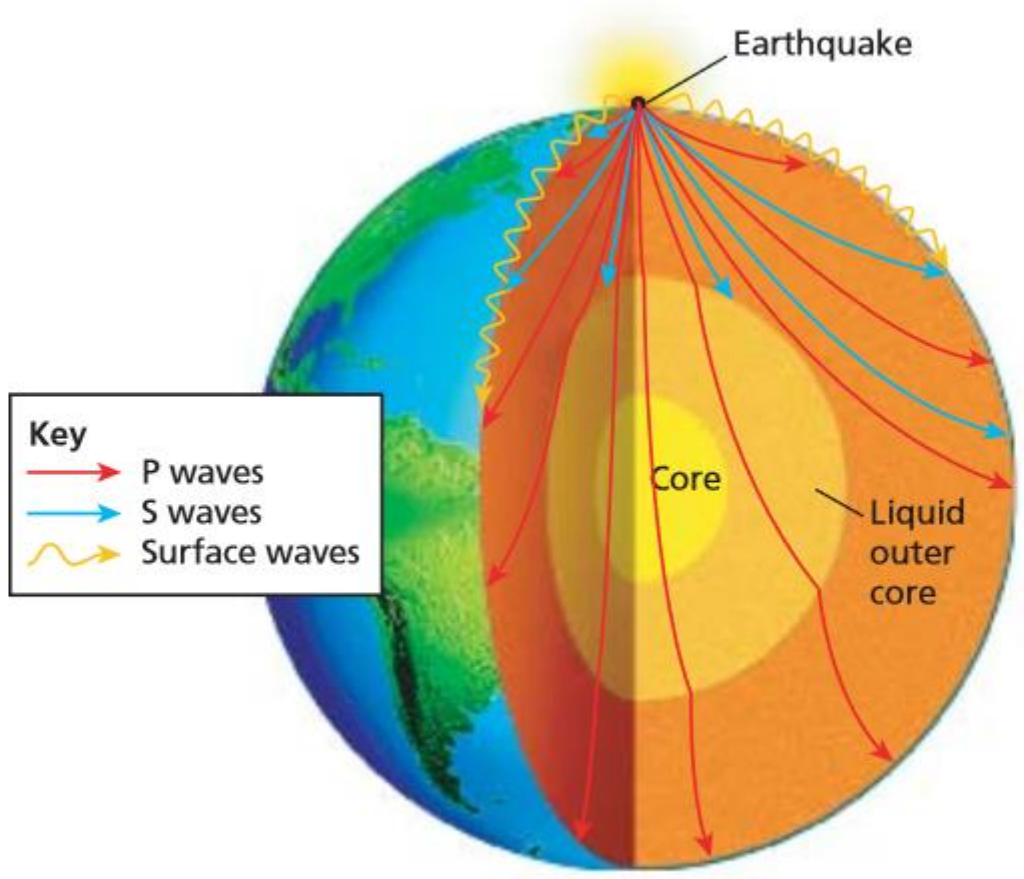
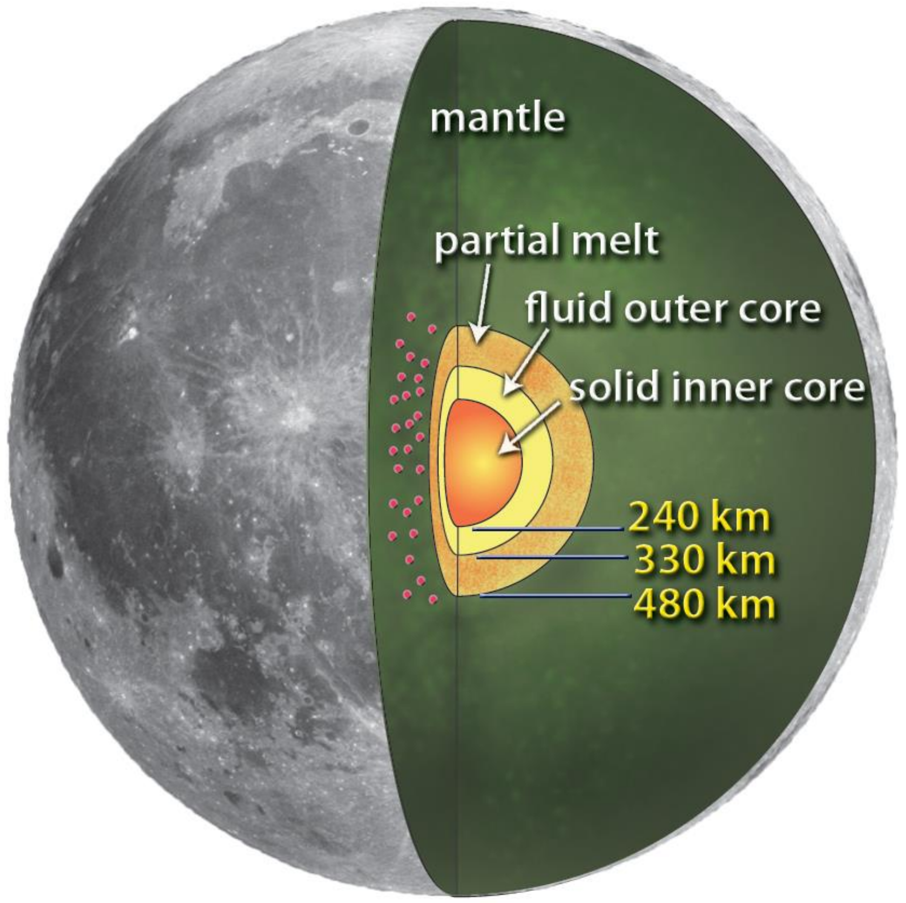


Figure 2. Seismograms of the Indian nuclear test (top) and a representative nearby earthquake (bottom) recorded at the seismic station at Nilore, Pakistan. These seismic signatures for an explosion and earthquake are typical and clearly distinguish one from the other.

Seismic data

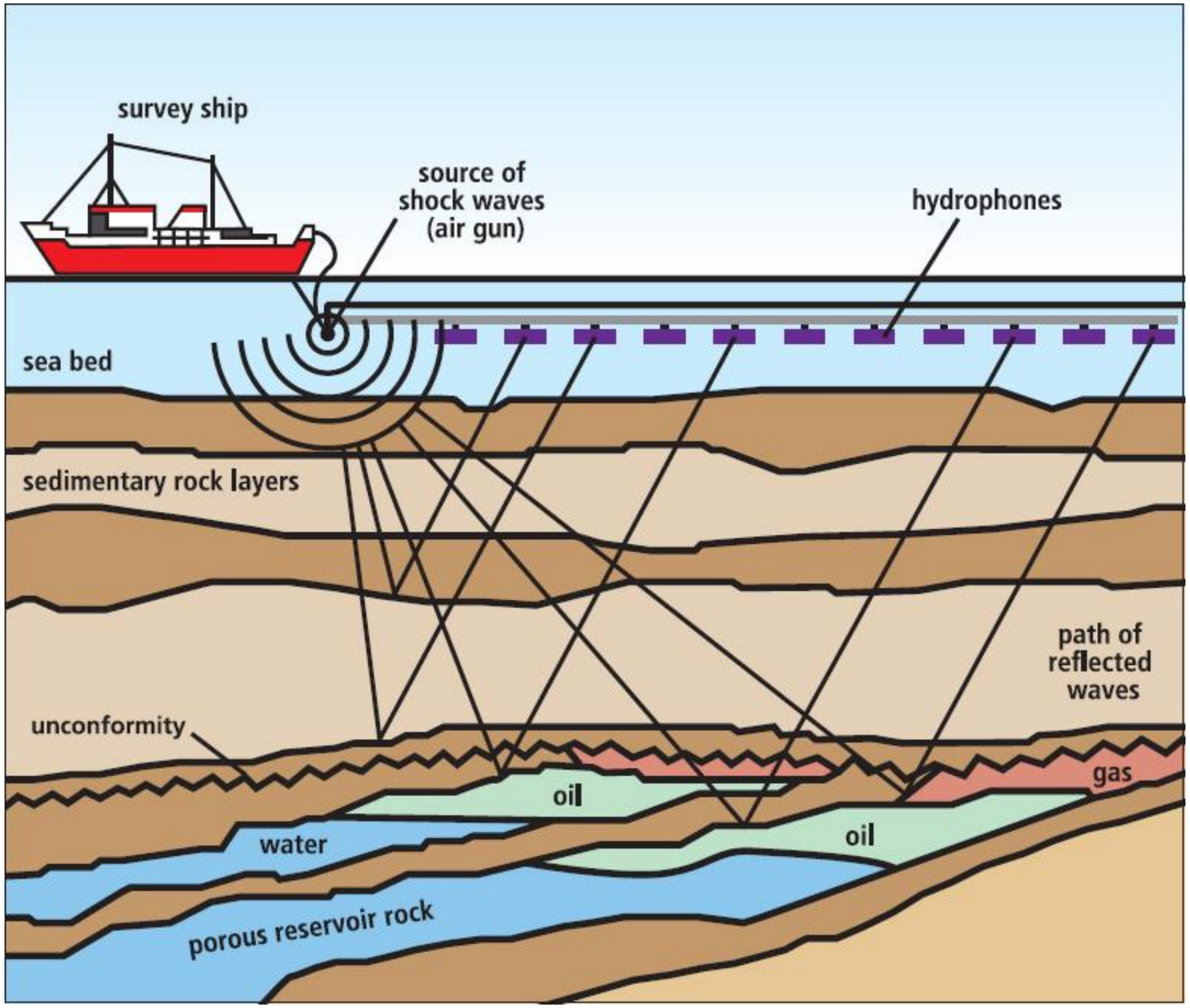
http://www.goodrichscience.com/3-seismic-waves.html
http://www.nasa.gov/images/content/508668main_lunar_core_lg.jpg



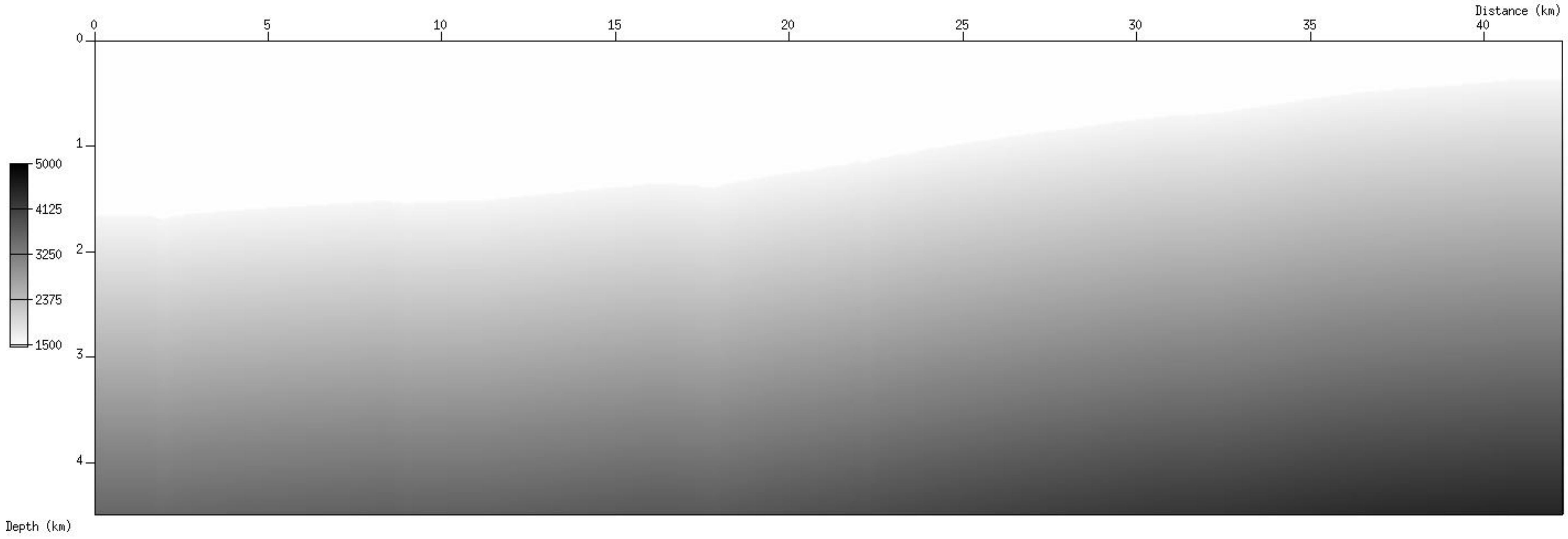


Seismic data

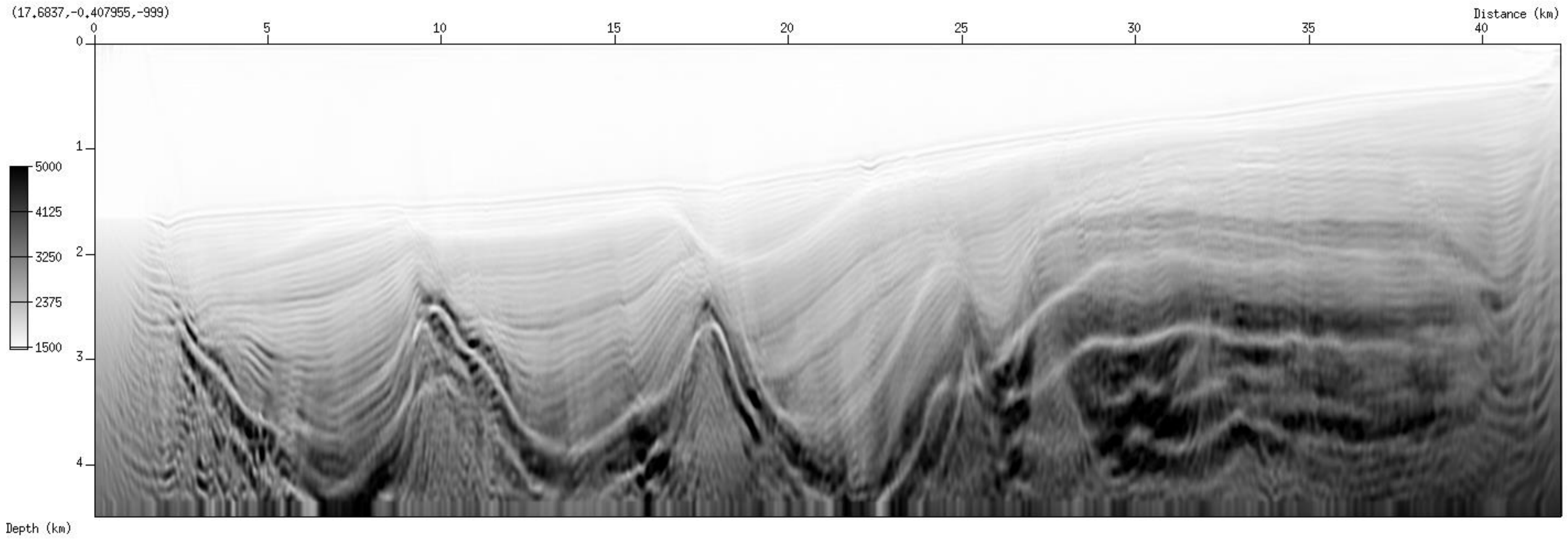
http://deliveryimages.acm.org/10.1145/2040000/2038877/figs/uf3.html



Seismic data

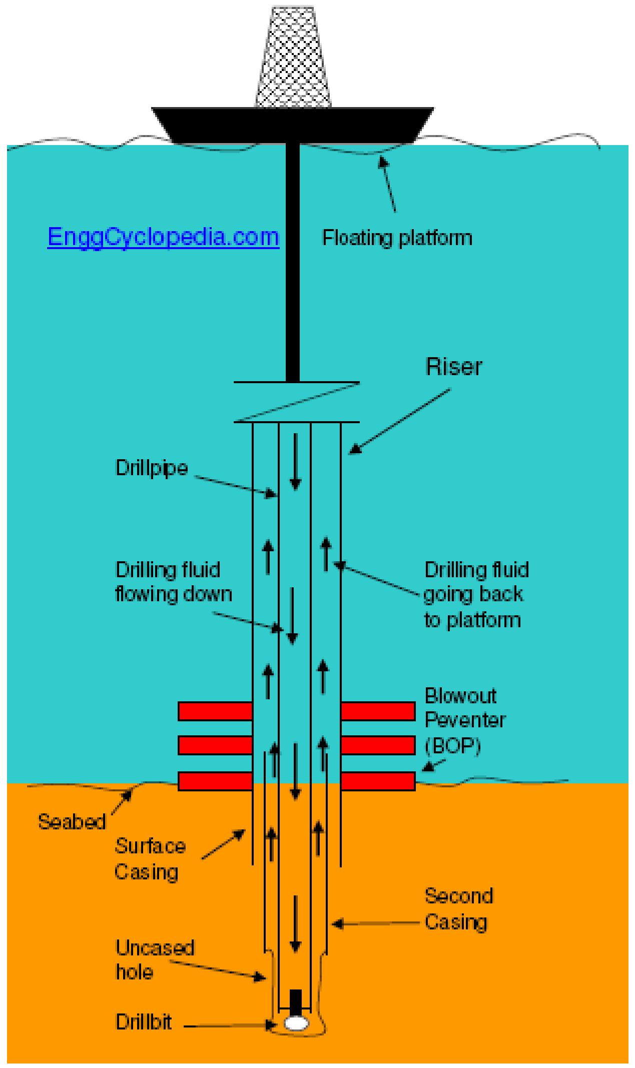


Seismic data



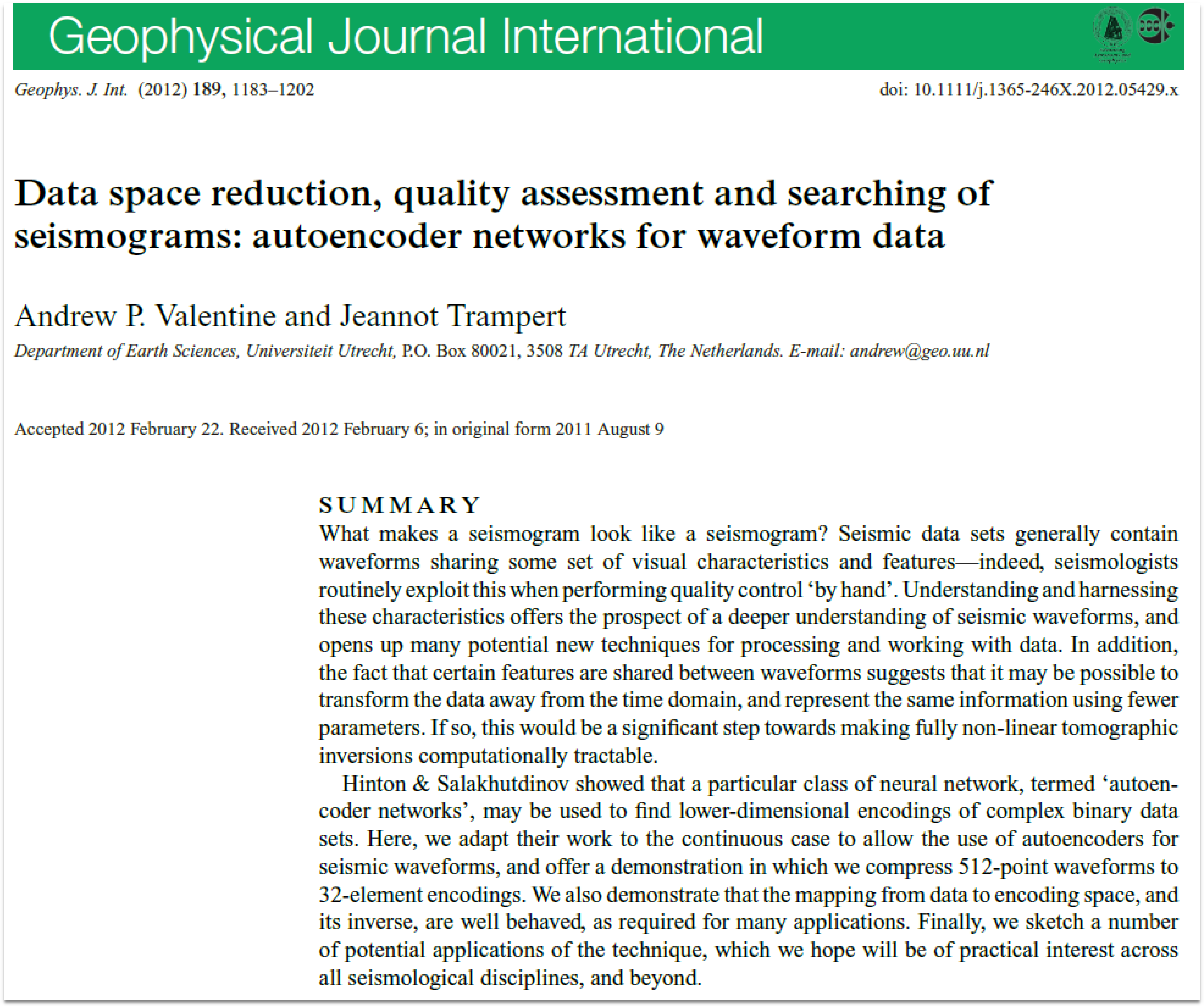
Seismic data

www.enggcyclopedia.com
www.offshoreenergytoday.com



Seismic data and… money

Paper 1


Geophysical Journal International

Geophys. J. Int. (2012) **189**, 1183–1202 doi: 10.1111/j.1365-246X.2012.05429.x

Data space reduction, quality assessment and searching of seismograms: autoencoder networks for waveform data

Andrew P. Valentine and Jeannot Trampert

Department of Earth Sciences, Universiteit Utrecht, P.O. Box 80021, 3508 *TA Utrecht, The Netherlands. E-mail: andrew@geo.uu.nl*

Accepted 2012 February 22. Received 2012 February 6; in original form 2011 August 9

SUMMARY

What makes a seismogram look like a seismogram? Seismic data sets generally contain waveforms sharing some set of visual characteristics and features—indeed, seismologists routinely exploit this when performing quality control ‘by hand’. Understanding and harnessing these characteristics offers the prospect of a deeper understanding of seismic waveforms, and opens up many potential new techniques for processing and working with data. In addition, the fact that certain features are shared between waveforms suggests that it may be possible to transform the data away from the time domain, and represent the same information using fewer parameters. If so, this would be a significant step towards making fully non-linear tomographic inversions computationally tractable.

Hinton & Salakhutdinov showed that a particular class of neural network, termed ‘autoencoder networks’, may be used to find lower-dimensional encodings of complex binary data sets. Here, we adapt their work to the continuous case to allow the use of autoencoders for seismic waveforms, and offer a demonstration in which we compress 512-point waveforms to 32-element encodings. We also demonstrate that the mapping from data to encoding space, and its inverse, are well behaved, as required for many applications. Finally, we sketch a number of potential applications of the technique, which we hope will be of practical interest across all seismological disciplines, and beyond.

선정이유

친절함

중요함

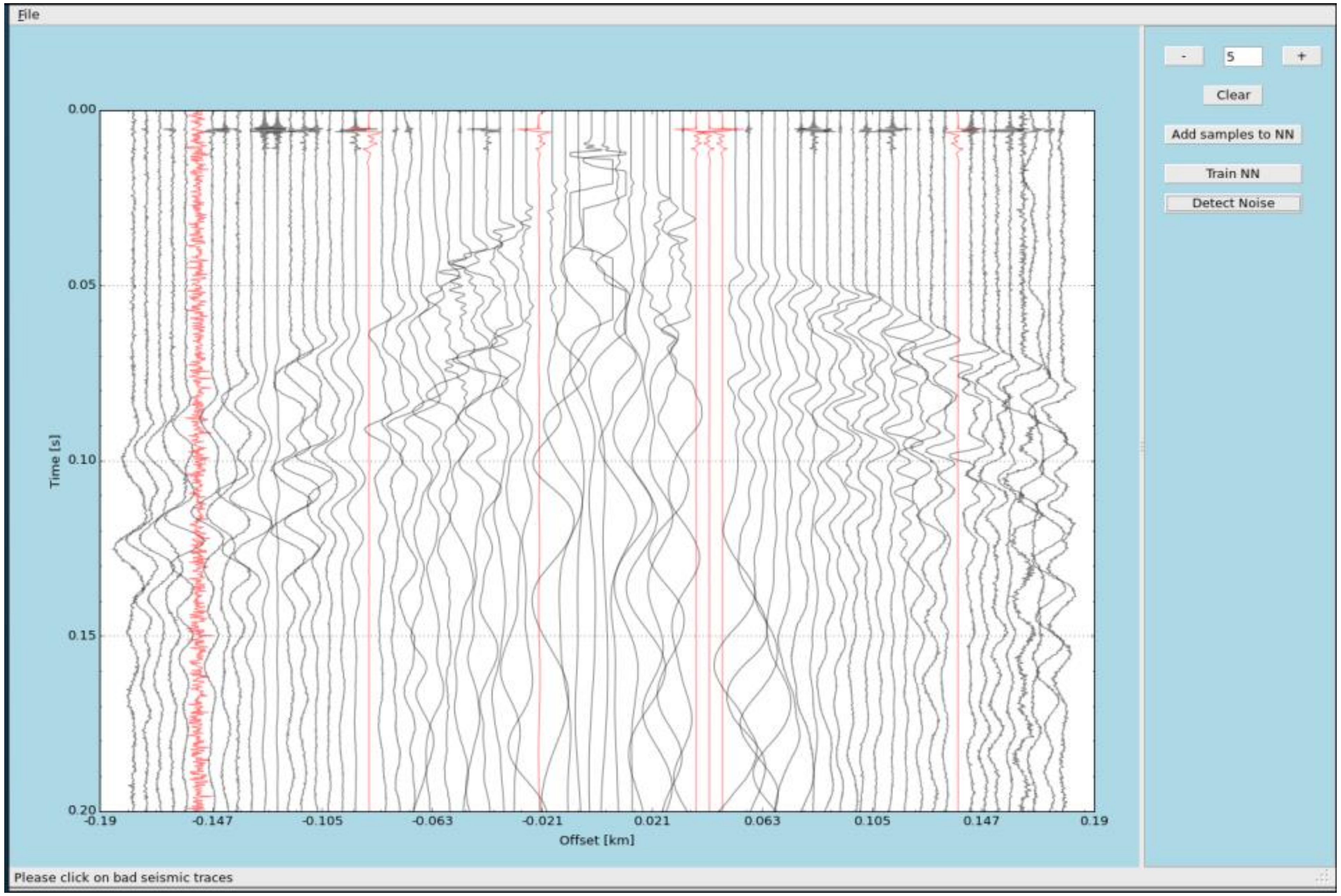
최신성

Paper 1: training data

https://geohanshacks.wordpress.com/2015/07/21/neural-networks-for-the-removal-of-bad-traces/

Training data is hand-picked

Raw trace :6000, Picked trace: 1700 (1000 training)



Please click on bad seismic traces

Paper 1 : Motivation

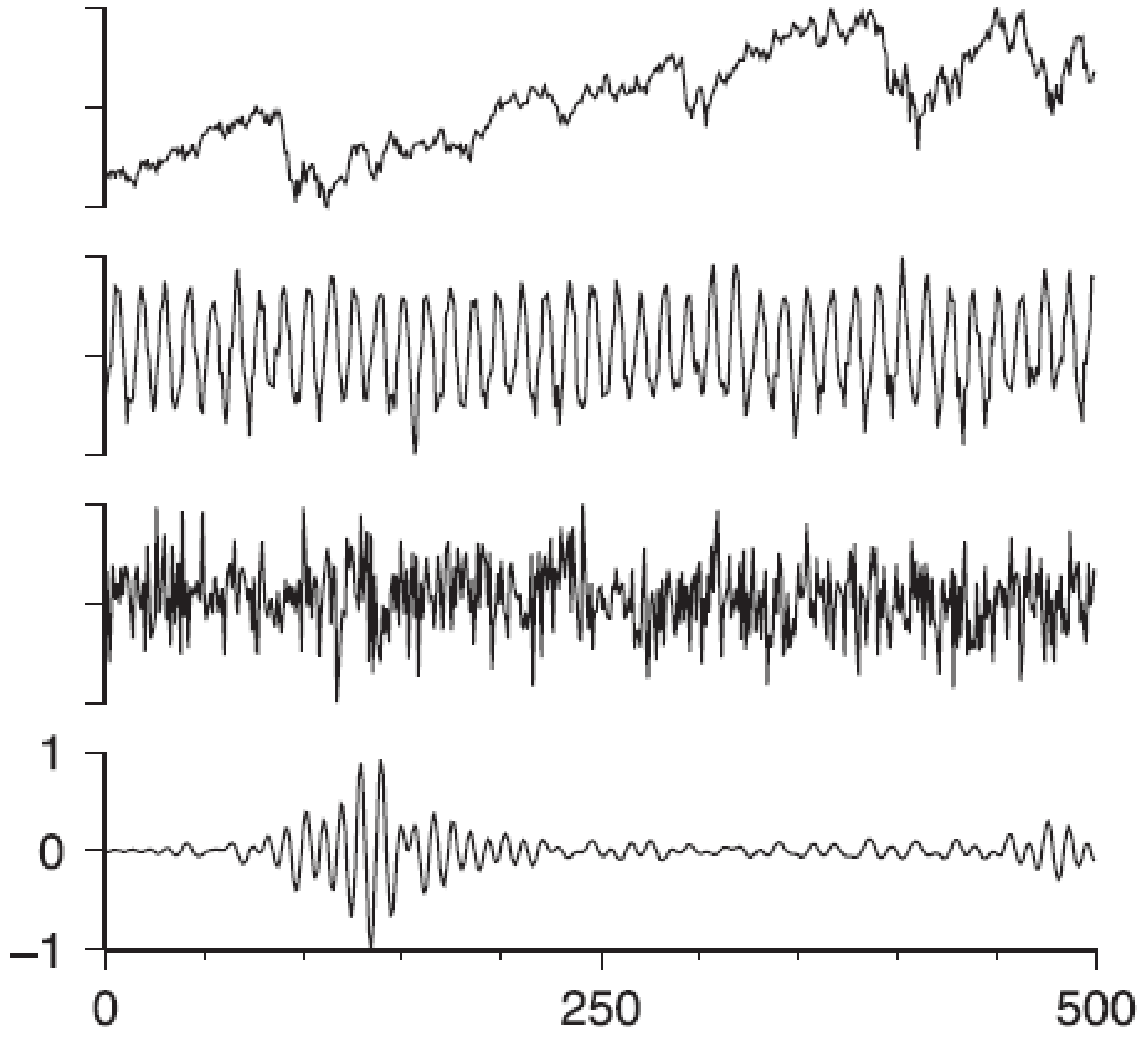
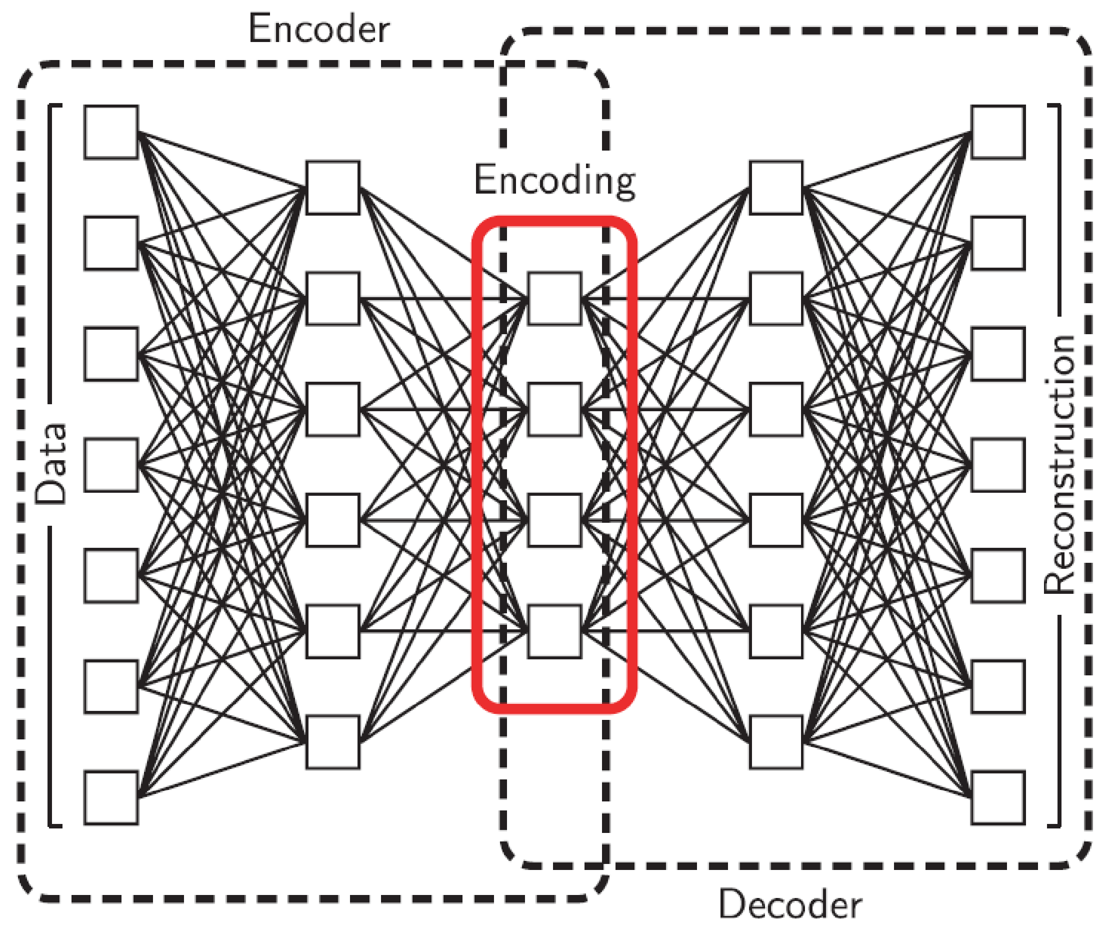


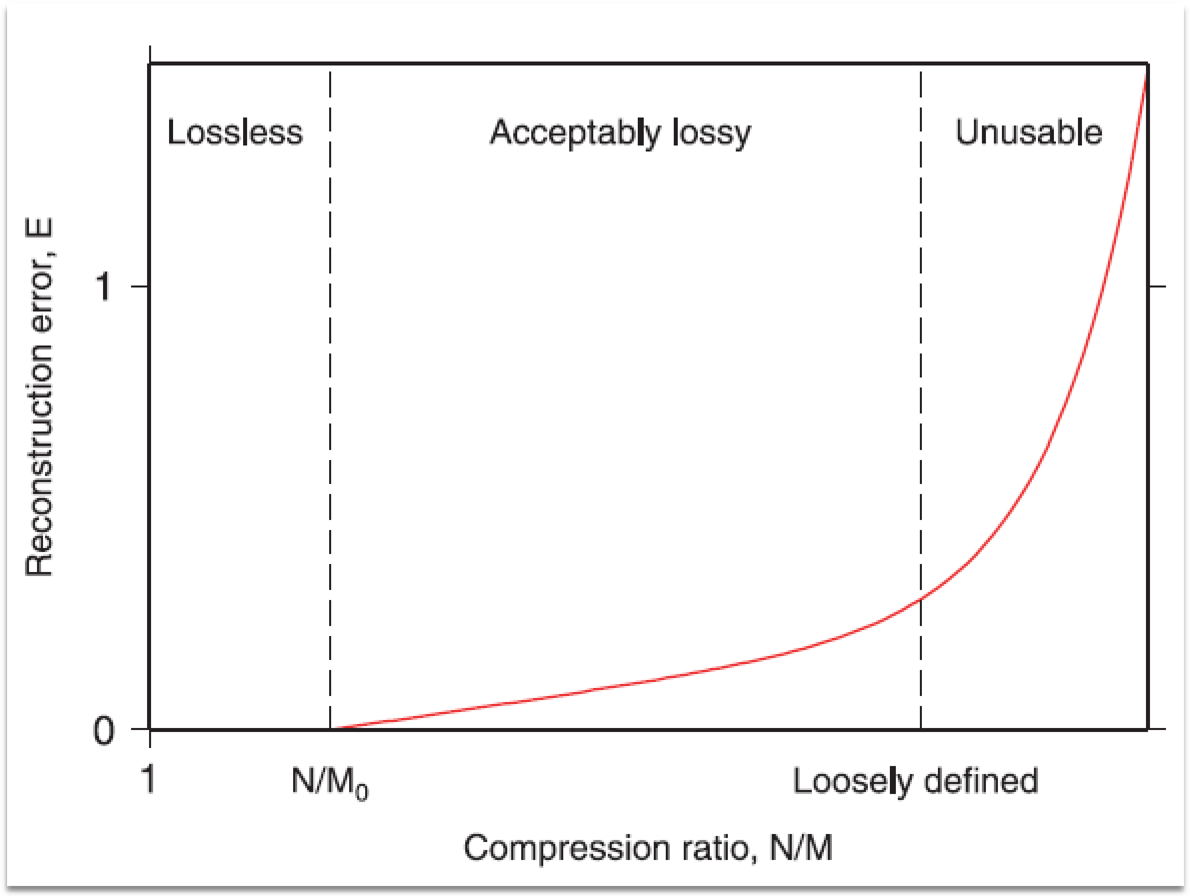
Figure 1. 'Spot the seismogram'. Four 500-point time-series, normalized to take amplitudes in the range [−1, 1]—but the seismogram has sufficient characteristic features to make it instantly recognizable. From top: FTSE 100 closing prices, 2009 June–2011 May; monthly mean temperature for central England, 1950–1991 (Parker *et al.* 1992); Gaussian random noise; long-period surface wave seismogram.



Reduction to encoded space

512-256-128-64-32-62-128-256-512

Paper 1 : Background



Compression-Error trade-off

$$\mathbf{t}_i = \underset{N\to M}{\mathrm{enc}}(\mathbf{s}_i)\,.$$

$$\mathbf{s}'_i = \underset{M\to N}{\mathrm{dec}}(\mathbf{t}_i)\,,$$

$$E_i = \frac{1}{2}(\mathbf{s}'_i - \mathbf{s}_i)\cdot(\mathbf{s}'_i - \mathbf{s}_i)\,.$$

$$\bar{E} = \lim_{Q\to\infty}\frac{1}{Q}\sum_{i=1}^{Q} E_i\,,$$

Paper 1 : Background

$$y = f(ab + a(\mathbf{w} \cdot \mathbf{x})),$$

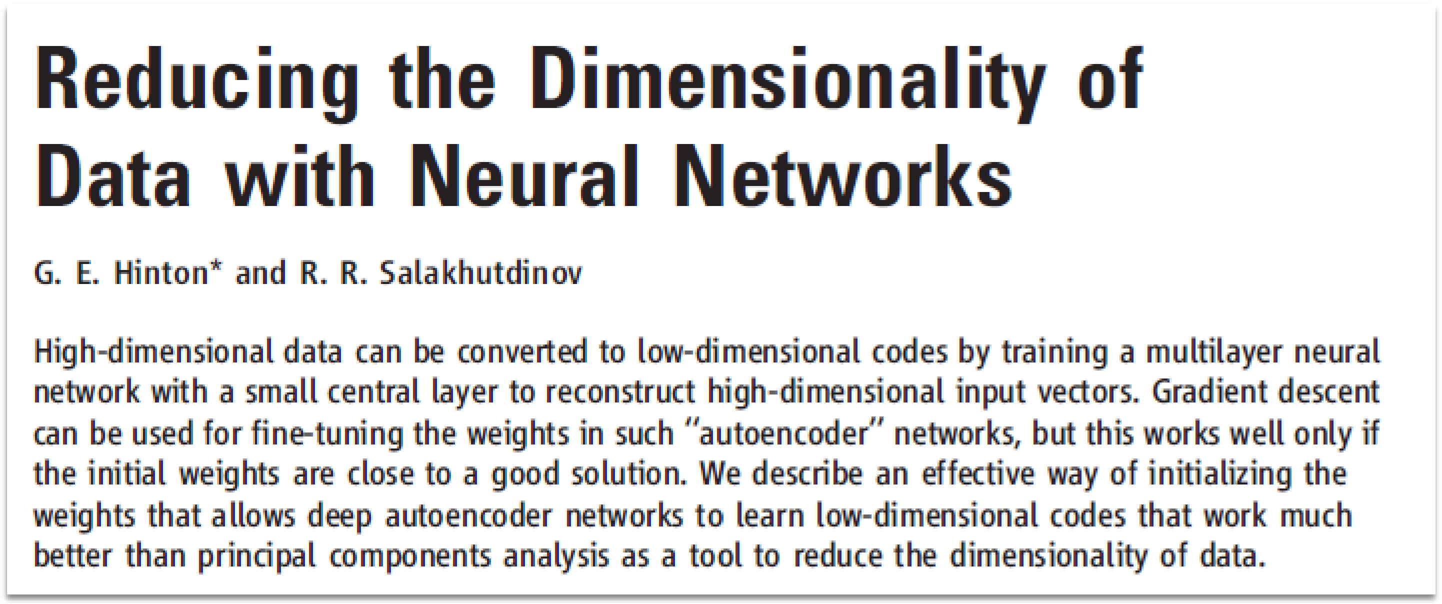
$$f(x) = f_0 + \frac{f_1 - f_0}{1 + \exp(-x)},$$

Sigmoid activation function

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}x} &= (f_1 - f_0)\frac{\exp(-x)}{[1 + \exp(-x)]^2} \\ &= \frac{1}{f_1 - f_0}[f(x) - f_0][f_1 - f(x)],\end{aligned}$$

Easy to compute back propagation

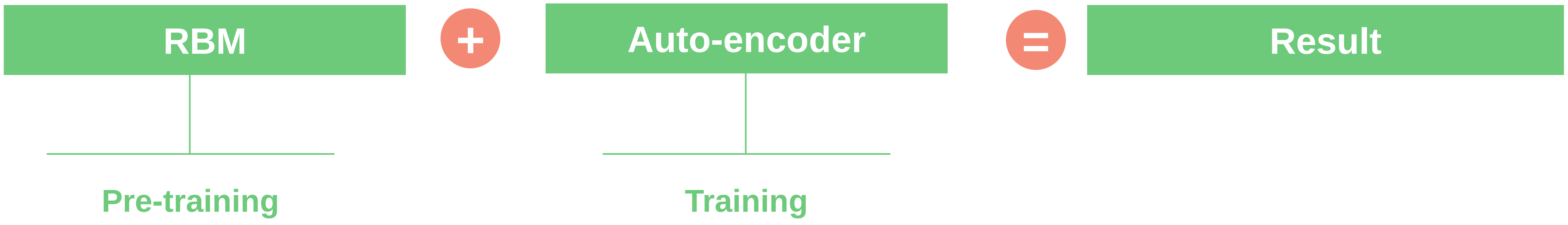
Paper 2



Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.



Paper 2

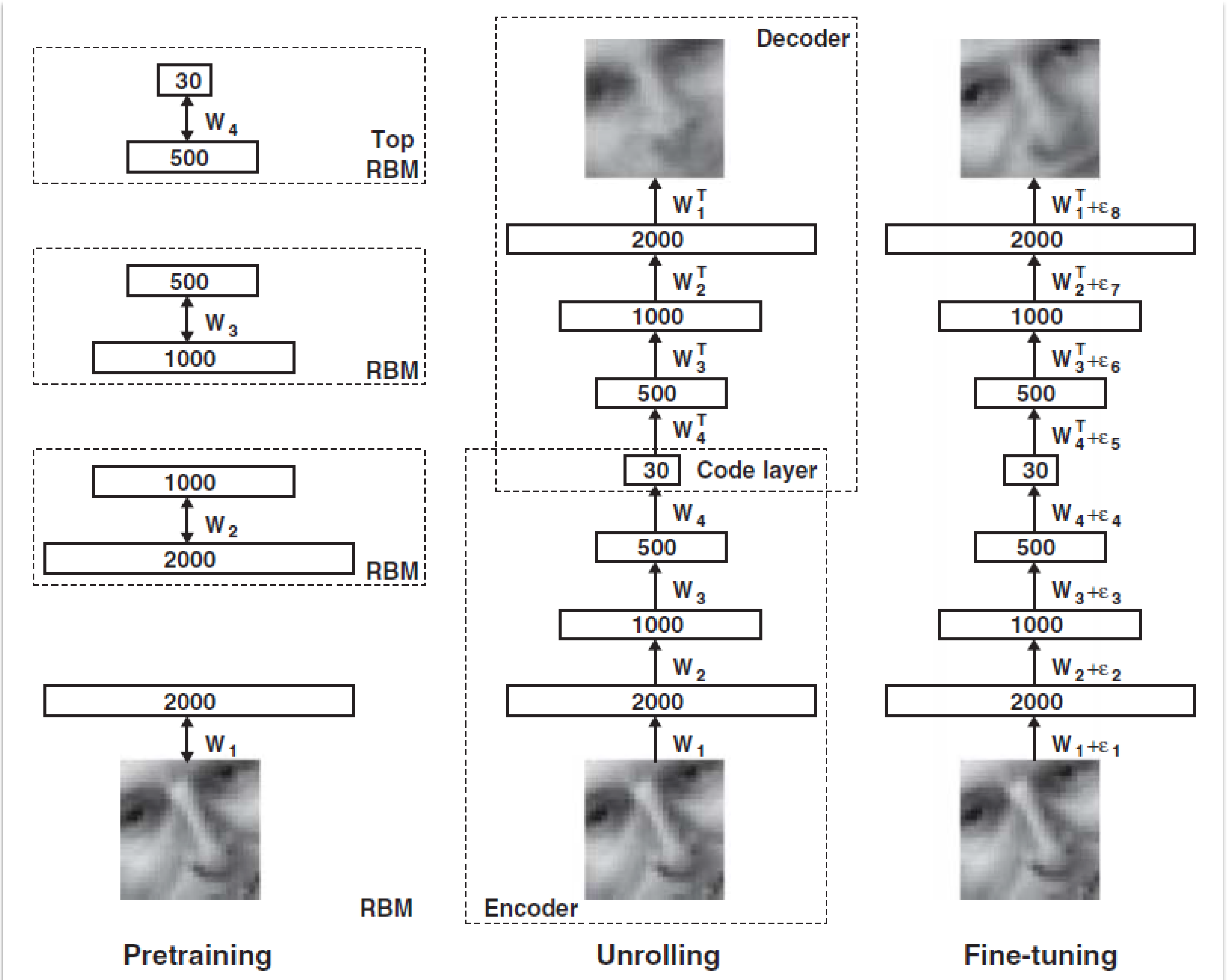


Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the "data" for training the next RBM in the stack. After the pretraining, the RBMs are "unrolled" to create a deep autoencoder, which is then fine-tuned using backpropagation of error derivatives.

Paper 2

Supporting figures

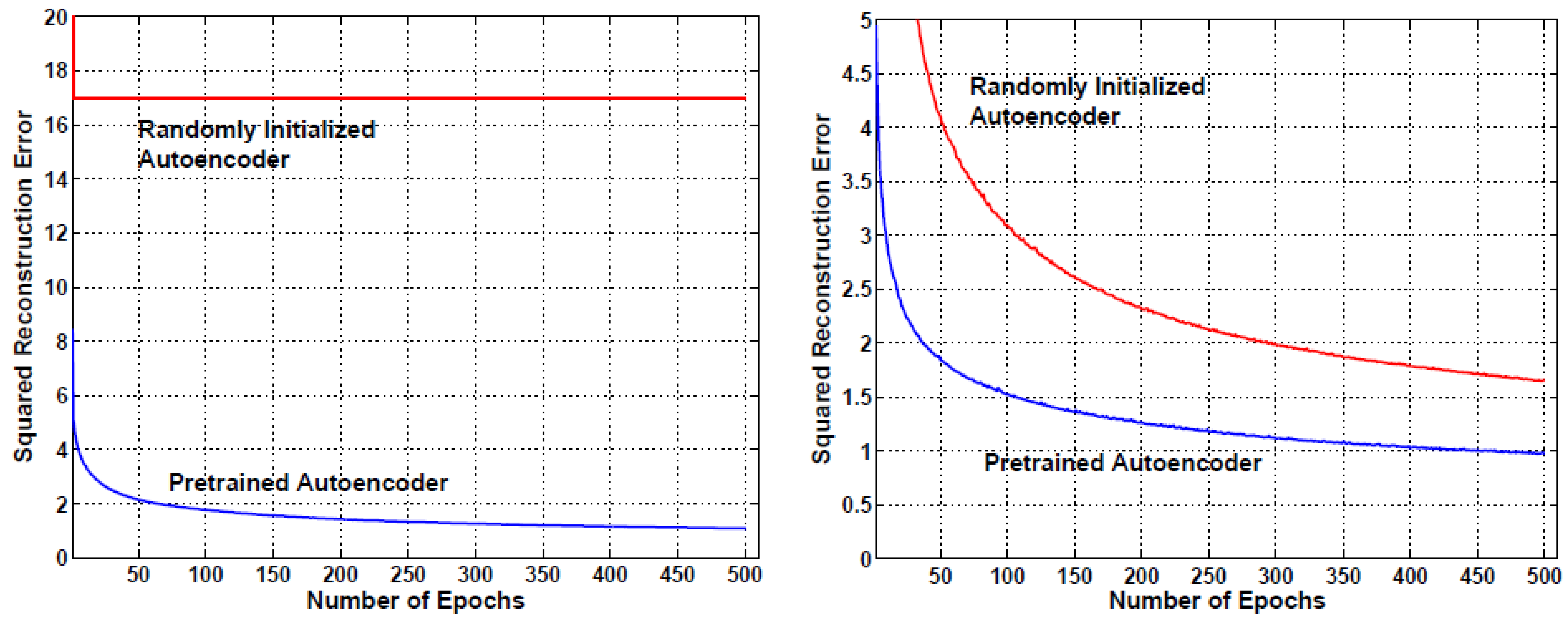


Fig. S1: The average squared reconstruction error per test image during fine-tuning on the curves training data. Left panel: The deep 784-400-200-100-50-25-6 autoencoder makes rapid progress after pretraining but no progress without pretraining. Right panel: A shallow 784-532-6 autoencoder can learn without pretraining but pretraining makes the fine-tuning much faster, and the pretraining takes less time than 10 iterations of fine-tuning.

Paper 2

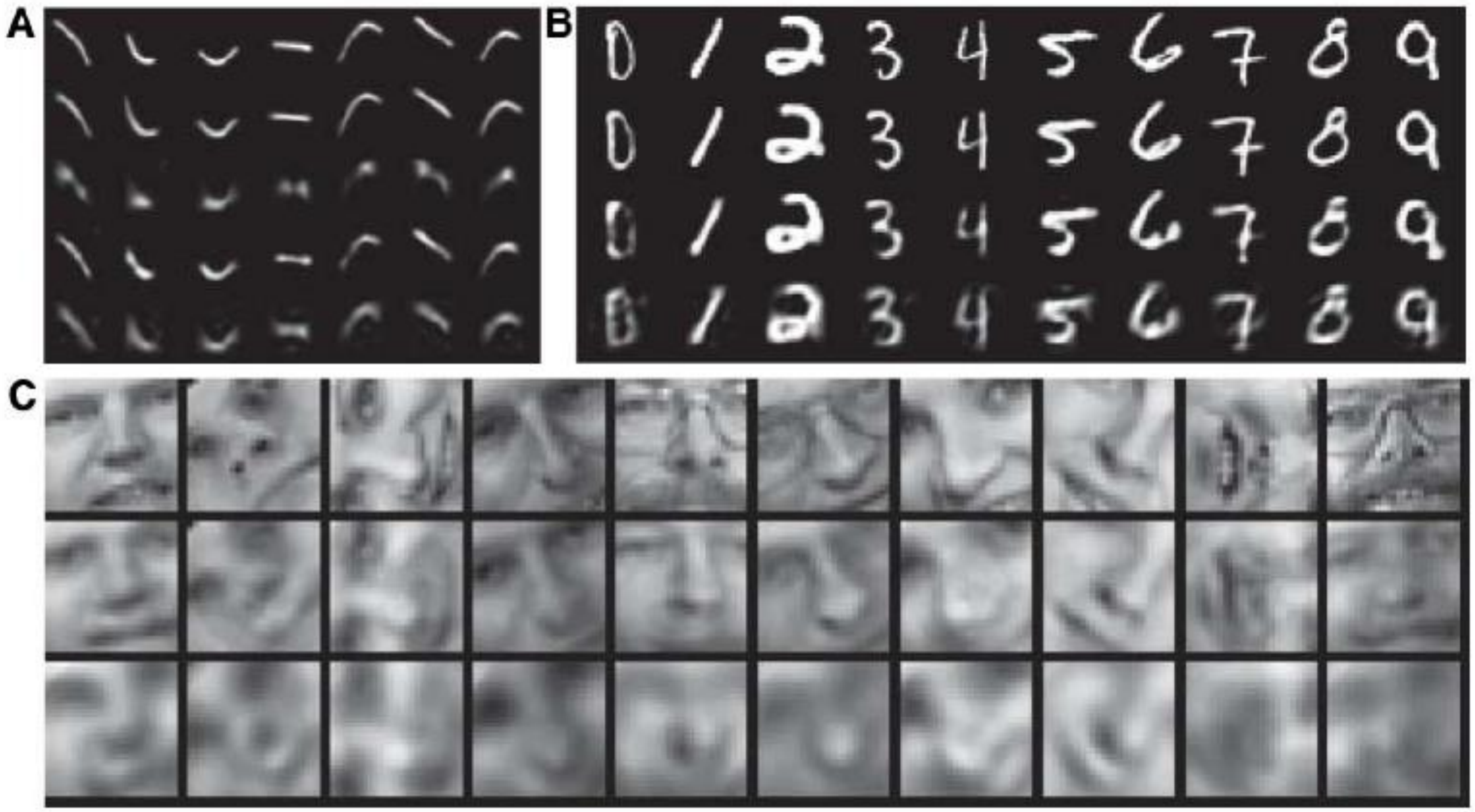


Fig. 2. (**A**) Top to bottom: Random samples of curves from the test data set; reconstructions produced by the six-dimensional deep autoencoder; reconstructions by "logistic PCA" (*8*) using six components; reconstructions by logistic PCA and standard PCA using 18 components. The average squared error per image for the last four rows is 1.44, 7.64, 2.45, 5.90. (**B**) Top to bottom: A random test image from each class; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional logistic PCA and standard PCA. The average squared errors for the last three rows are 3.00, 8.01, and 13.87. (**C**) Top to bottom: Random samples from the test data set; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional PCA. The average squared errors are 126 and 135.

Paper 2

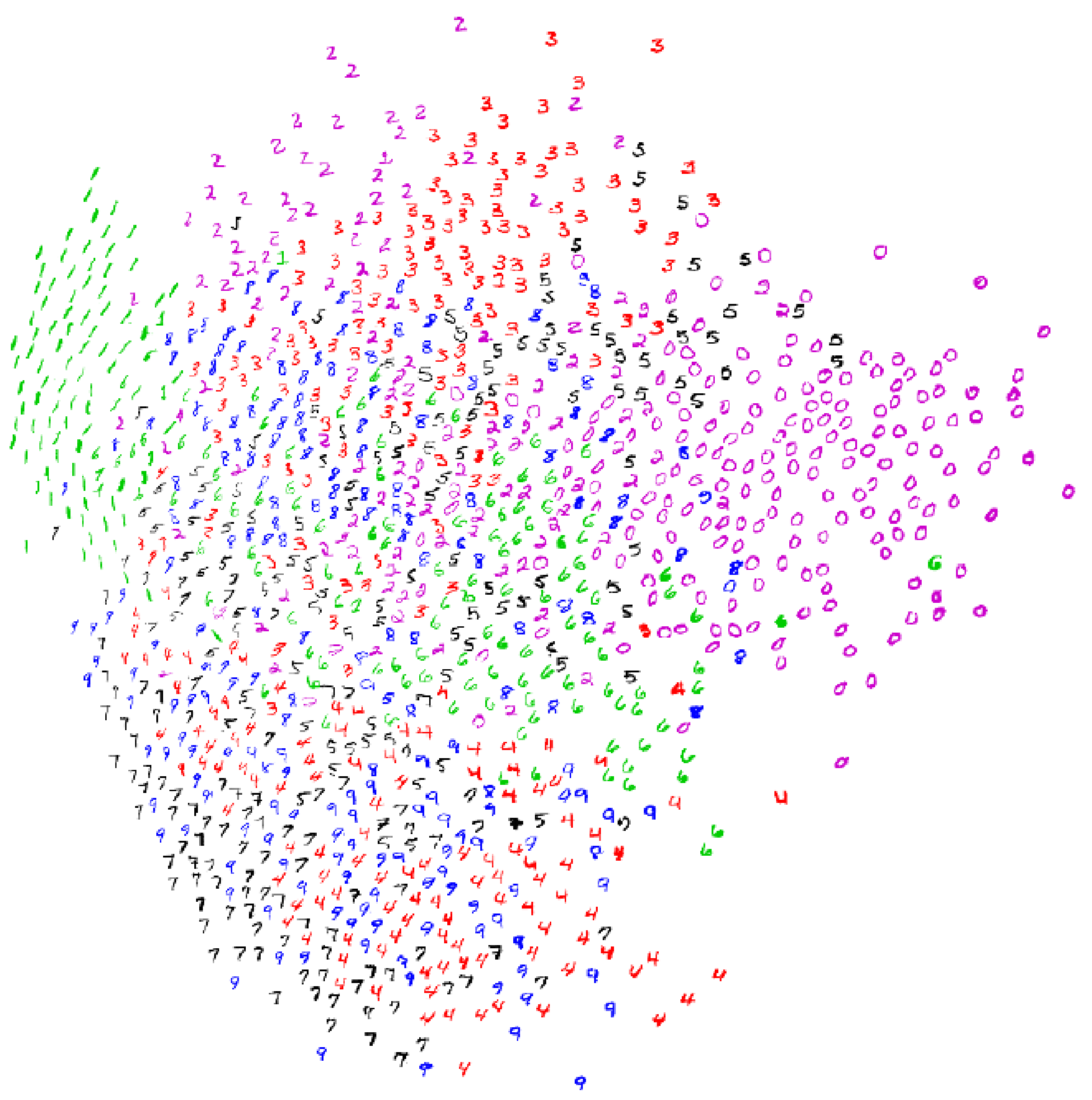


Fig. S3: An alternative visualization of the 2-D codes produced by taking the first two principal components of all 60,000 training images. 5,000 images of digits (500 per class) are sampled in random order. Each image is displayed if it does not overlap any of the images that have already been displayed.

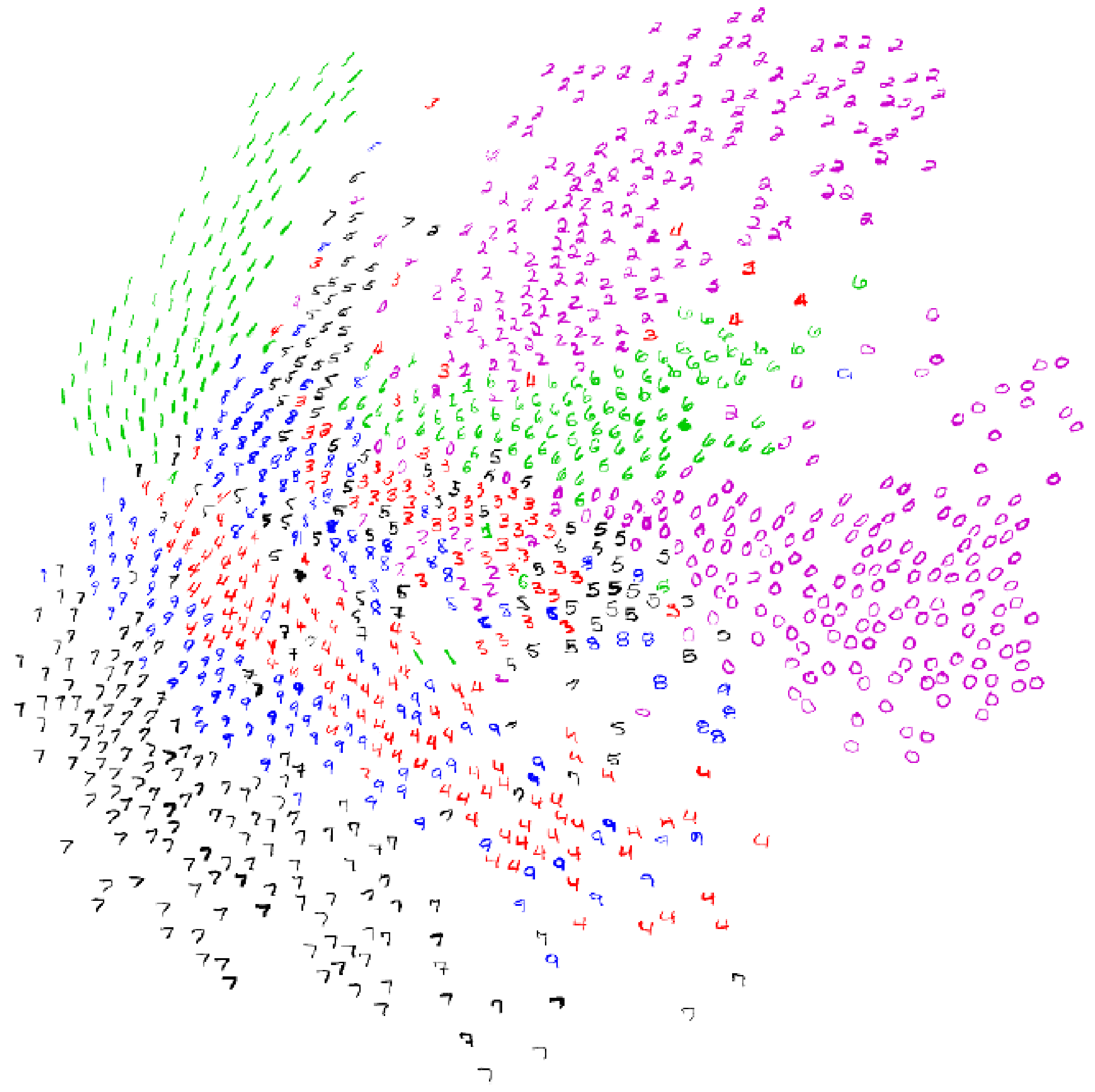


Fig. S4: An alternative visualization of the 2-D codes produced by a 784-1000-500-250-2 autoencoder trained on all 60,000 training images. 5,000 images of digits (500 per class) are sampled in random order. Each image is displayed if it does not overlap any of the images that have already been displayed.

Paper 2

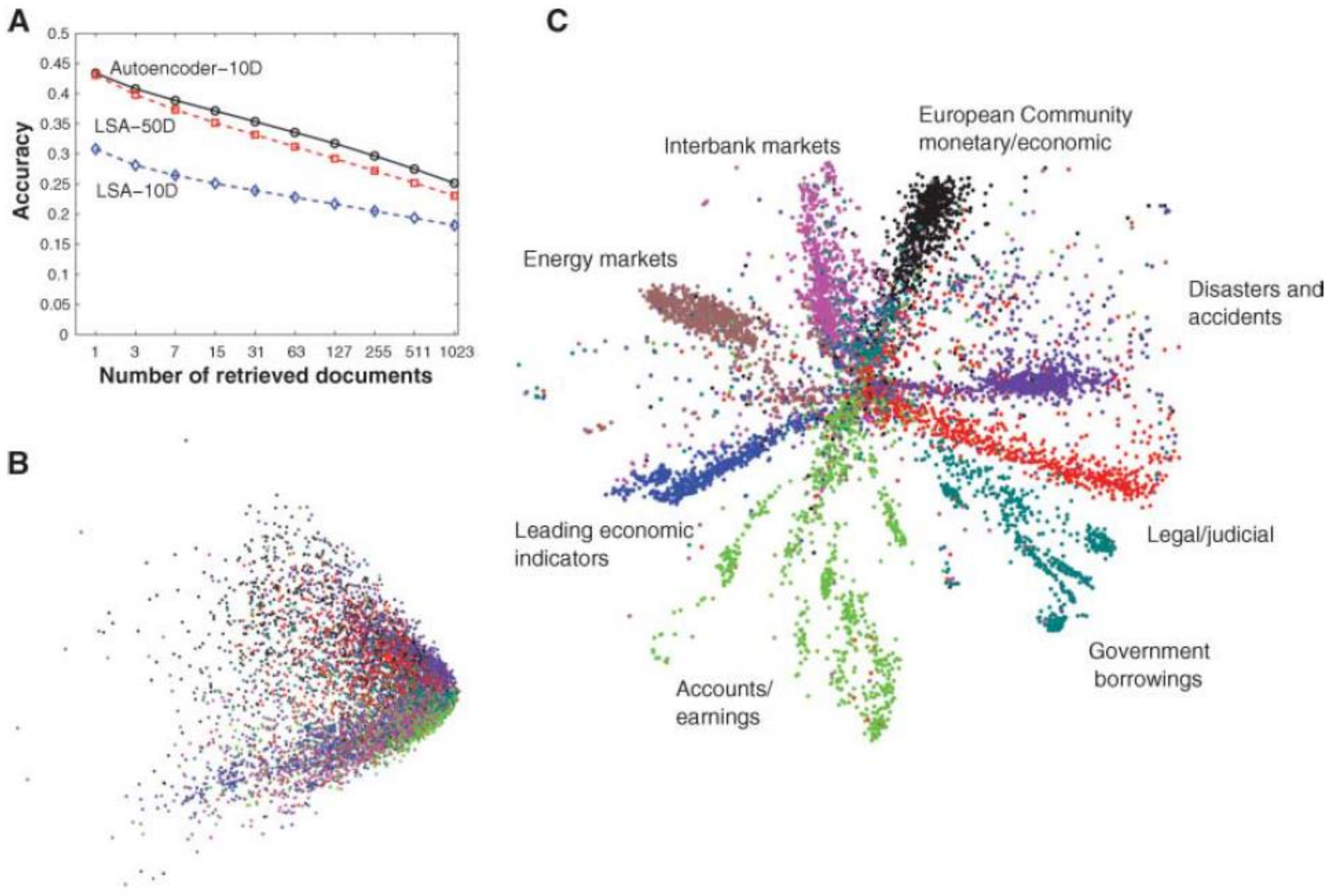


Fig. 4. (**A**) The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. (**B**) The codes produced by two-dimensional LSA. (**C**) The codes produced by a 2000-500-250-125-2 autoencoder.

Paper 1 : Method

Noise

$$x_{ij}^{(0)} \to x_{ij}^{(0)} + \mathcal{G}(0, \sigma^{(A)}) \, ,$$

Initialization

$$w_{ij}^{\mathrm{v,h}} = \mathcal{G}(0, 0.01) \, ; \qquad b_i^{\mathrm{v,h}} = \mathcal{G}(0, 0.01) \, ; \qquad a_i^{\mathrm{v,h}} = 1 \, , \quad (39)$$

Learning rate

$$\eta \to \eta_0 + \frac{i}{I} (\eta - \eta_0) \, ,$$

Paper 1 : Result

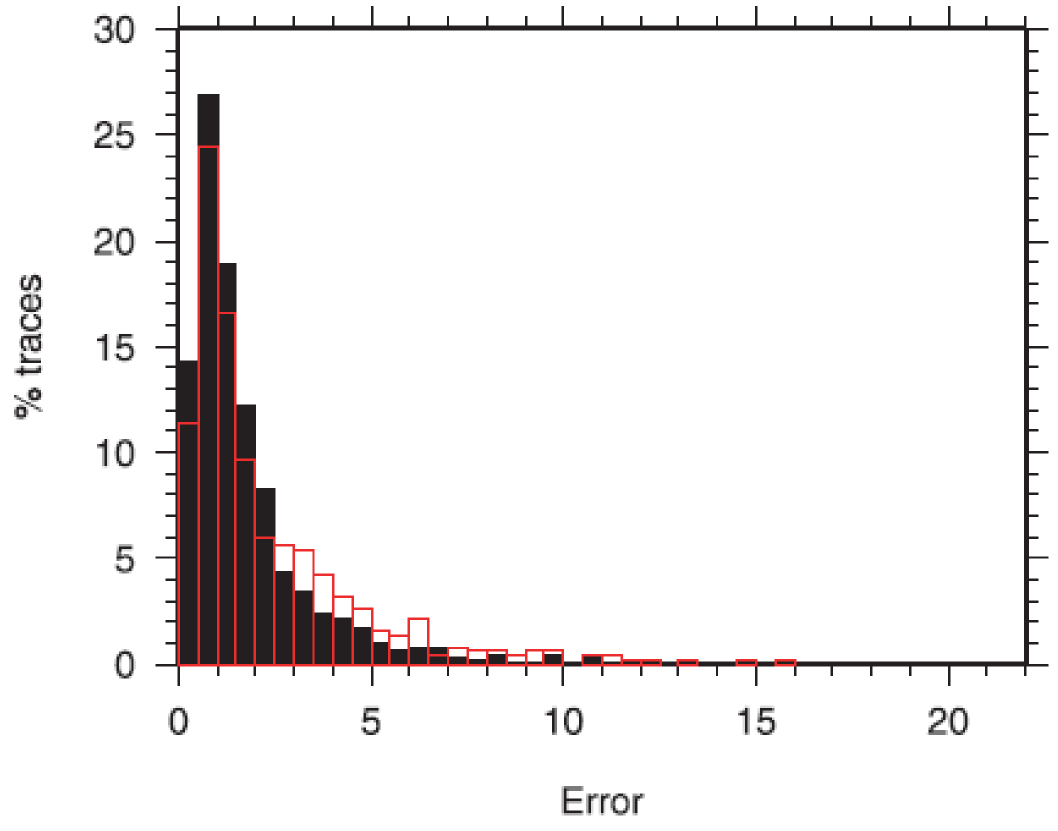


Figure 8. Errors in recovery of original traces from encoding. Histogram showing trace-by-trace error computed according to eq. (5) for training (solid black bars) and monitoring (red outline) data sets. Some examples from the monitoring data set can be found in Fig. 9, to allow error levels to be related to waveform discrepancies.

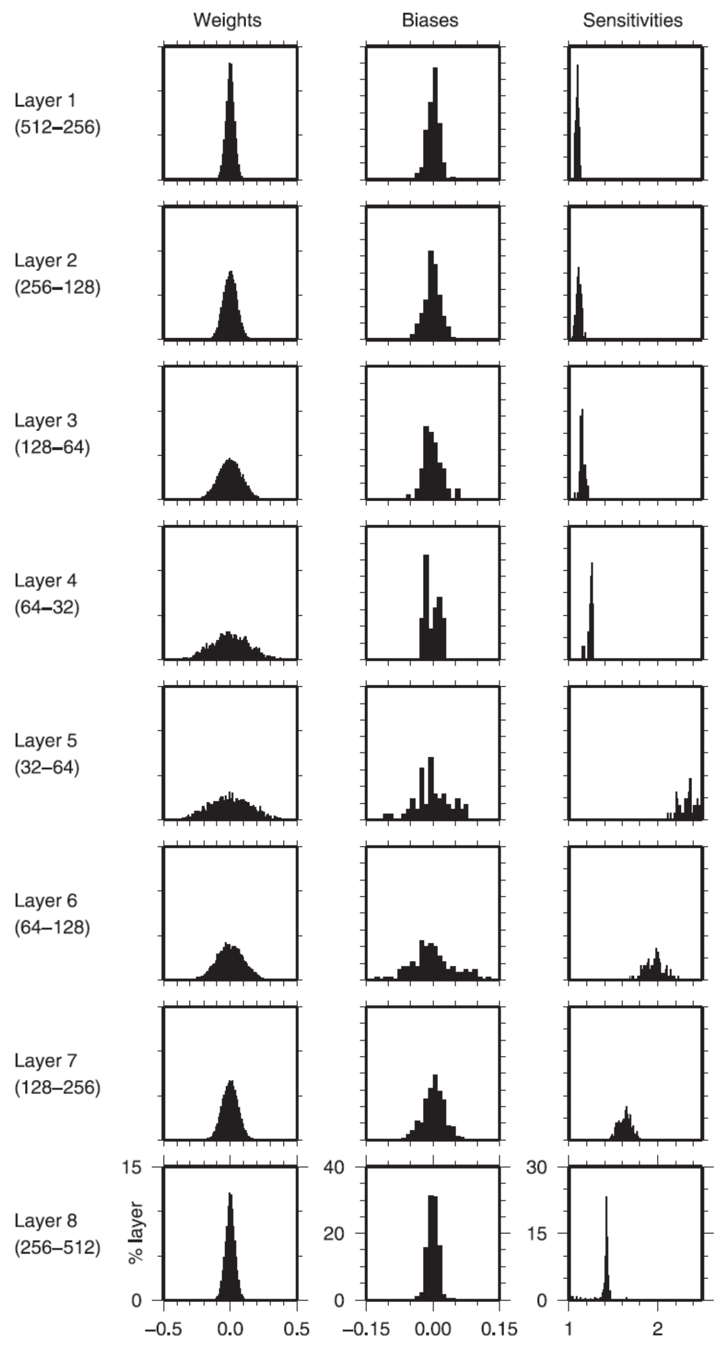


Figure 10. Weights, biases and sensitivities of trained autoencoder. Histograms showing distribution of values of elements of $\mathbf{W}^{(n)}$, $\mathbf{b}^{(n)}$ and $\mathbf{a}^{(n)}$ (as defined in eq. 14) for a trained autoencoder. All histograms in a given column share the same horizontal and vertical scales. Note that the number of elements varies between layers.

Paper 1 : Result

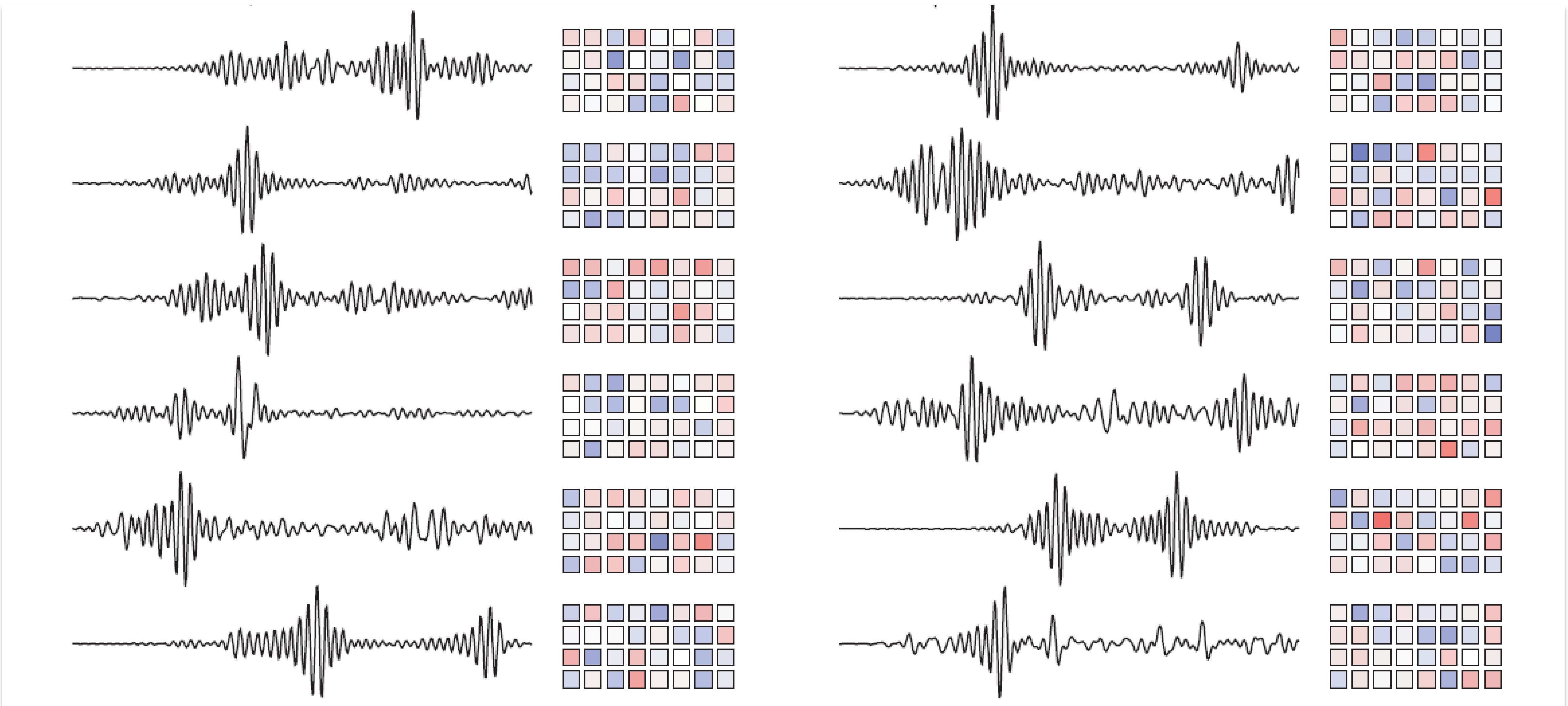
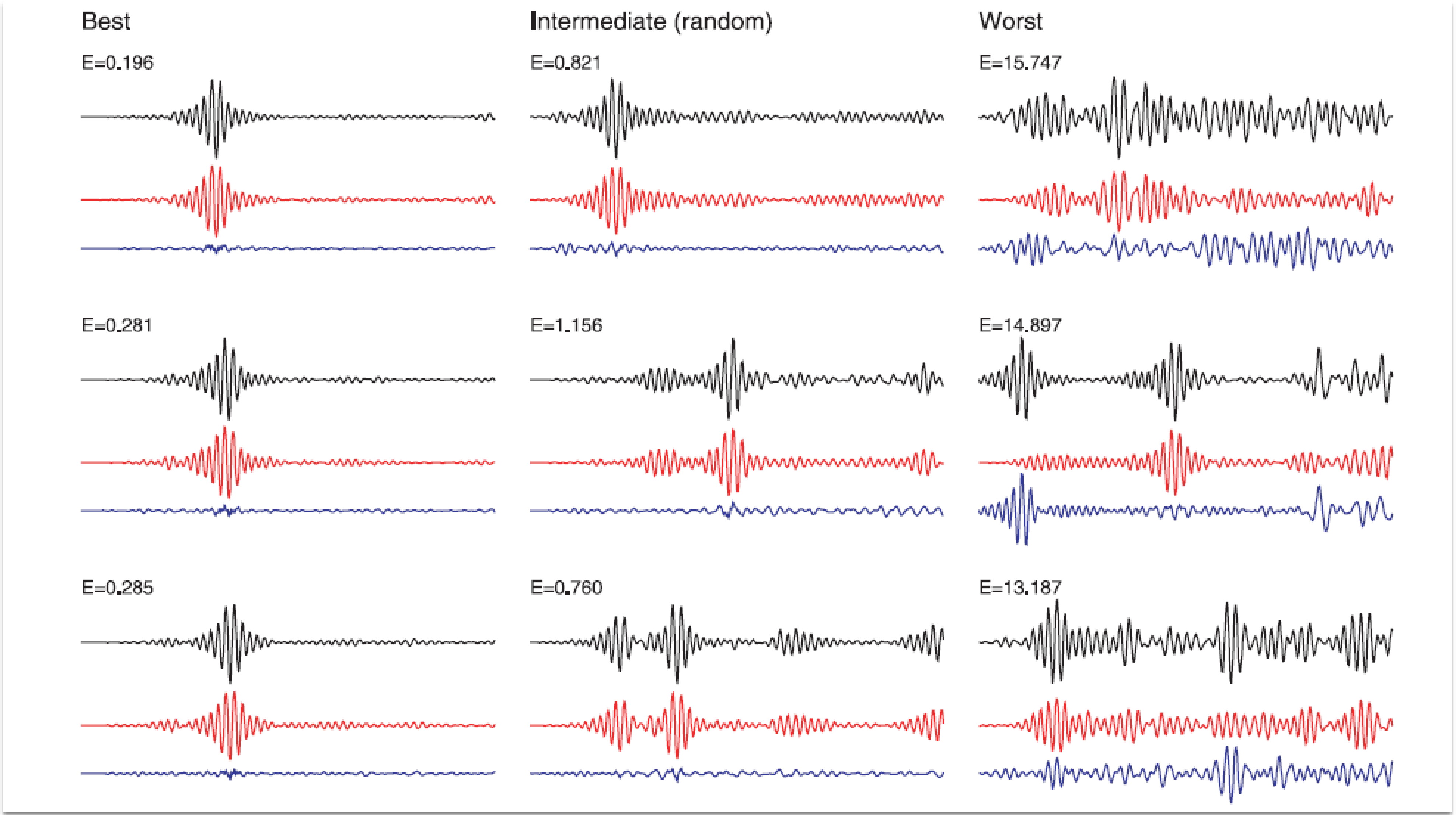


Figure 7. Examples of encodings. Waveforms from the monitoring data set and their encoding using a trained autoencoder. As there is no inherent temporal relationship between the elements of each encoding, we represent them in grid form; red colours represent positive numbers, and blues are negative. The colour scale is equivalent to that shown in Fig. 12.

Paper 1 : Result



Paper 1 : Result

Orthogonality

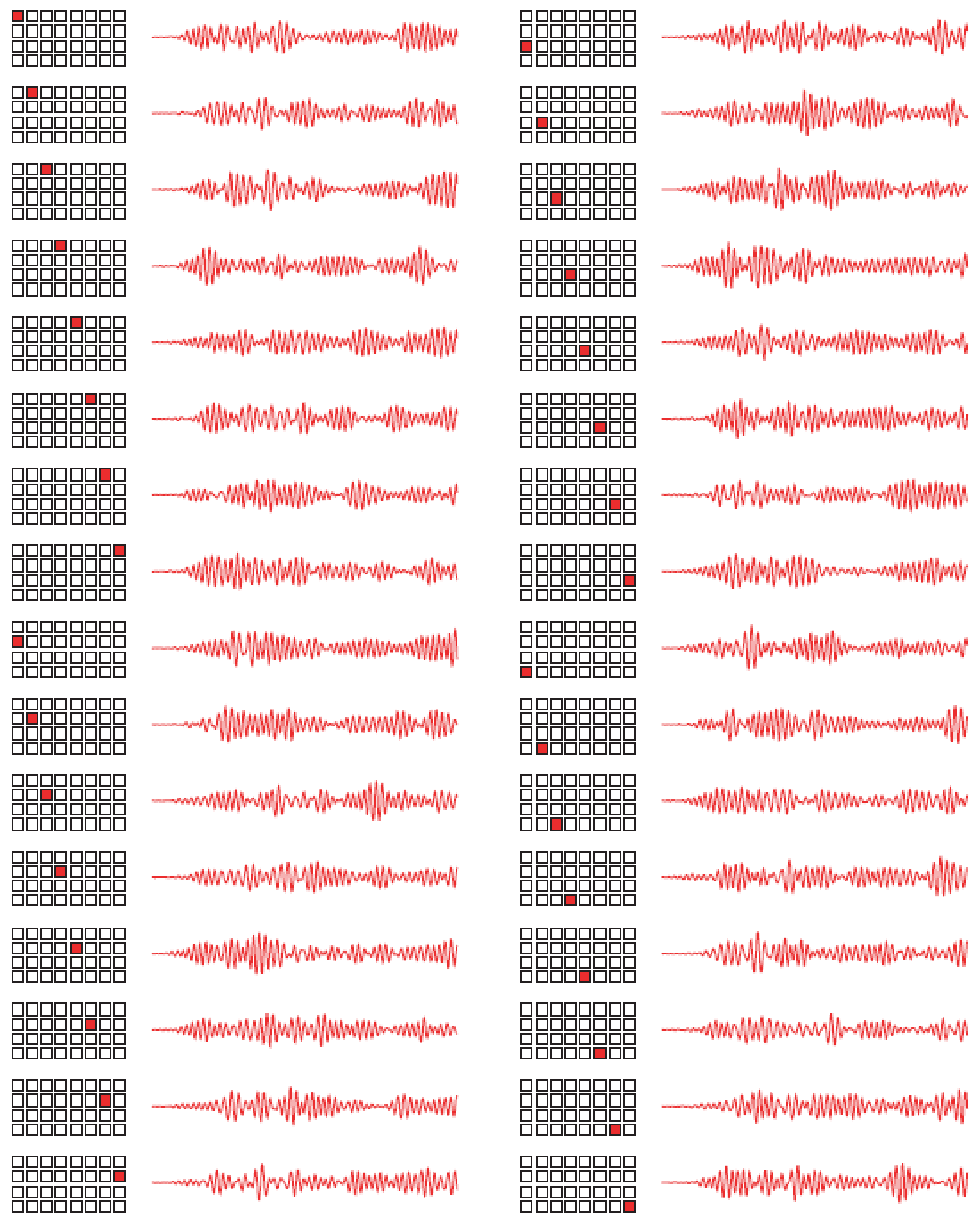


Figure 11. Waveforms corresponding to unit encodings. 32 waveforms obtained by decoding the unit vectors $\mathbf{x}^{(L/2)} = (1, 0, \ldots, 0)$, $\mathbf{x}^{(L/2)} = (0, 1, \ldots, 0)$, etc. We observe that each element of the encoding affects the entire length of seismogram.

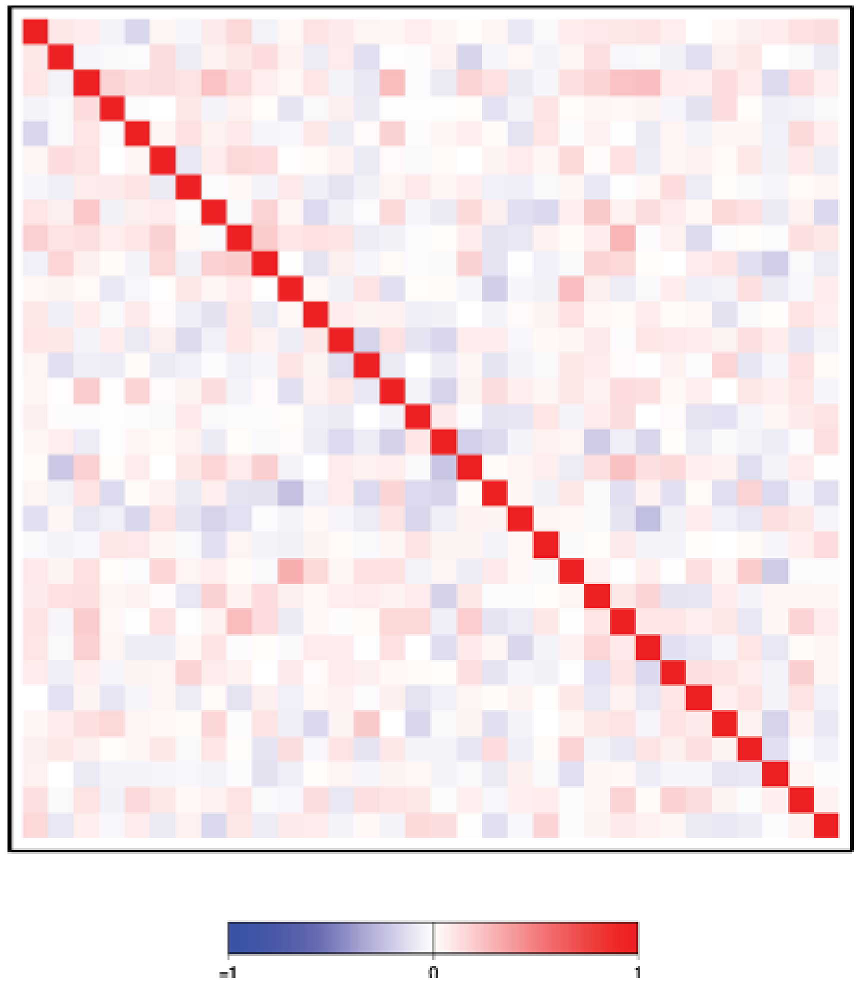


Figure 12. Orthogonal encodings produce (almost) orthogonal decodings. Inner product matrix **M** (as in eq. 40) formed from the 32 waveforms shown in Fig. 11. Clearly, these are close to forming an orthogonal set.

Paper 1 : Result

Closeness

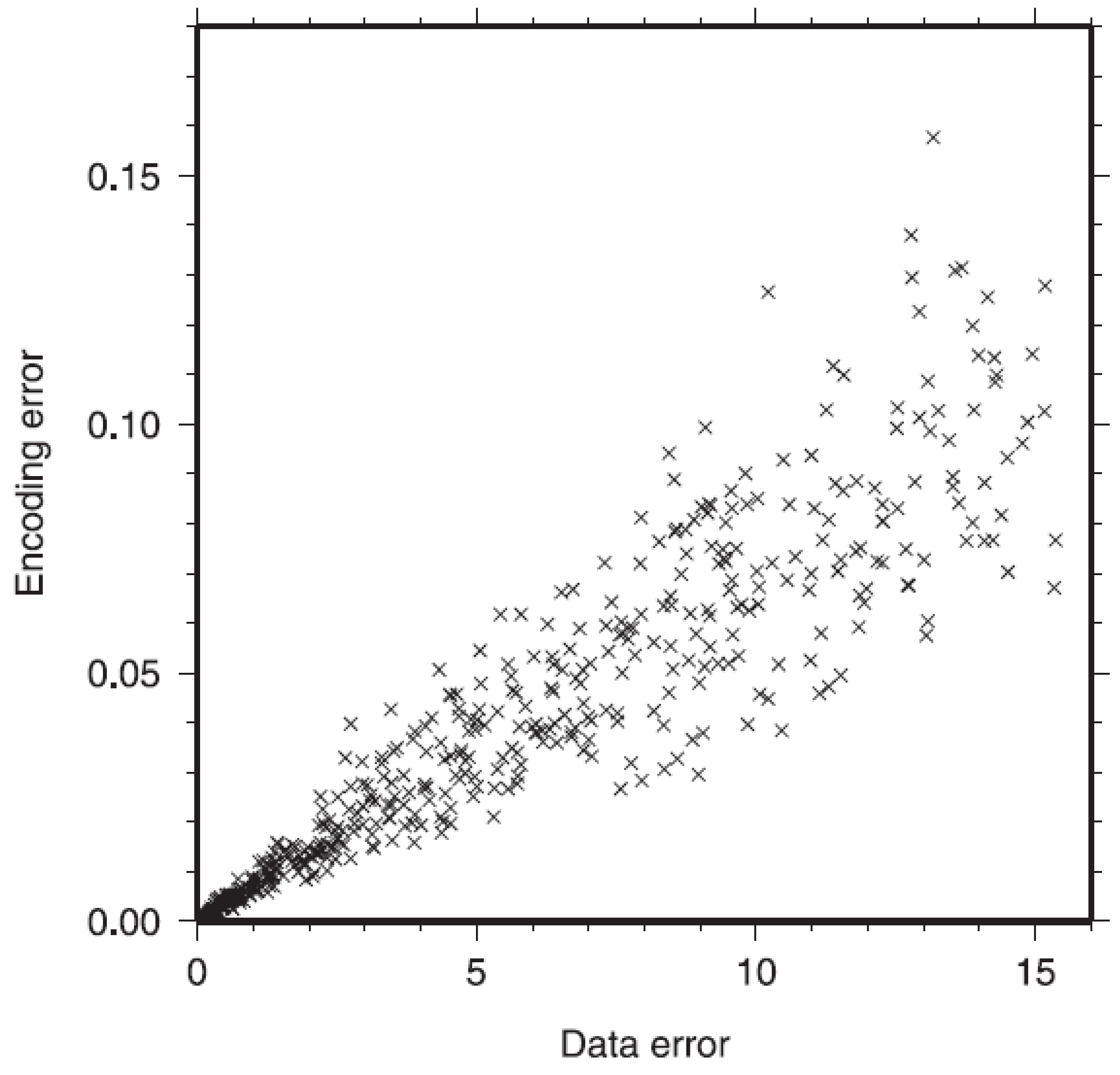


Figure 14. Waveforms that are 'close' have encodings that are 'close'. Plot of error between two waveforms against error between the corresponding encodings, defined as in eq. (5). For the interpretation of a given error in terms of waveform discrepancies, see Fig. 9.

Remark

내용쓰기

Effective

Orthogonality

Closeness

Applications

Further Study

Reference

Valentine, A.P. & Trampert, J., 2012a. Data-space reduction, quality assessment and searching of seismograms: autoencoder networks for waveform data, Geophys. J. Int., 189, 1183–1202

Hinton, G. & Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks, *Science,* **313,** 504–507.