Seismic data compression using deep neural network predictors

Conference Paper · August 2019

DOI: 10.1190/segam2019-3205392.1

5 authors, including:

Hilal Nuha
Telkom University
9 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Bo Liu
King Fahd University of Petroleum and Minerals
66 PUBLICATIONS **525** CITATIONS
SEE PROFILE

Mohamed Ahmed Mohandes
King Fahd University of Petroleum and Minerals
99 PUBLICATIONS **3,438** CITATIONS
SEE PROFILE

Faramarz Fekri
Georgia Institute of Technology
244 PUBLICATIONS **3,613** CITATIONS
SEE PROFILE

Seismic Data Compression Using Deep Neural Network Predictors

H. Nuha, A. Balghonaim, M. Mohandes, and Bo Liu, King Fahd University of Petroleum and Minerals;
Faramarz Fekri, Georgia Institute of Technology.

Summary :

Seismic data compression is highly demanded to reduce the cost for transmission and storage due to an enormous volume of collected data. This paper presents a prediction based compression for seismic data using deep neural networks (DNN) and entropy encoding. First, a DNN with multiple hidden layers is pre-trained using restricted Boltzmann machines (RBMs) to obtain good initial weights. The DNN is then fine-tuned in a supervised fashion to achieve a better prediction precision. The residual between actual and predicted samples are quantized to achieve smaller data representation. The quantized residuals are further encoded using the Huffman coding. Our experiments with a real data set show that the DNN significantly outperforms the Linear Predictive Compression (LPC) in term of reconstruction quality.

Introduction

Seismic data is collected for oil and gas explorations by a series of sensors located at different locations. Seismic exploration of the Earth's crust is used extensively not only on land but also offshore to identify oil and gas deposits in subterranean Earth formations. Seismic data takes enormous storage space and demands considerable transmission bandwidth to deliver the data from a seismic survey location to a processing center. Therefore, compressing the seismic data is desirable in order to reduce the cost for transmission and storage.

In recent years, many algorithms for seismic data compression have been proposed. The compression techniques can be divided into two, namely, lossless and lossy methods. In most seismic data compression applications, the lossless methods do not provide sufficient compression ratio, hence are inefficient. The near lossless methods, however, offer higher compression ratios and are expected to be practical in seismic data compression (Nave & Cohen, 1993; Rosten, et al., 2004; Lindstrom, et al., 2016).

Previous studies on seismic data compression mainly employed transformation and prediction based approaches for lossy and lossless compressions. Lossless prediction based approaches combined with entropy coding resulted in a low compression ratio with less than 5:1 for integer data (Stearns, et al., 1993) and less than 2:1 for floating point data with transformation technique (Lindstrom, et al., 2016), hence inefficient. Alternatively, the lossy compression scheme has been extensively studied in the past few years since it can achieve higher compression ratios (Spanias, et al., 1991; Averbuch, et al., 2001; Bernasconi & Vassallo, 2003; Rosten, et al., 2004; Rubin, et al., 2016; Liu, et al., 2018). The majority of the compression algorithms usually apply three common stages, namely de-correlation, quantization, and encoding. Transformation is a powerful tool for de-correlation that leads to a better representation of the data in the transform domain (Nuha, et al., 2017). Wavelets, local trigonometric functions, and a variety of combinations of the above have been widely used as the transformation basis (Villasenor, et al., 1996; Al-Moohimeed, 2004; Wu, et al., 2006). Prediction based approach is also extensively used for waveform data like seismic and electrocardiograph data (Stearns, et al., 1993; Nave & Cohen, 1993; Kannan & Eswaran, 2007). In the quantization and encoding stages, the residuals and coefficients from the previous stage are then quantized and encoded using fixed length encoding (Fajardo, et al., 2015) or variable length encoding for a lower bitrate (Averbuch, et al., 2001).

The compression algorithm used in this work is a prediction based method using deep learning architecture. In the conventional prediction based method, the $(n+1)^{th}$ sample of a signal is predicted by its n past samples. The discrepancies between the predicted and actual samples are sent or stored as residuals instead of the original sample itself. The compression is achieved because the variance of the residuals is lower than that of the original sample, and hence may be represented by fewer bits per sample. We use the DNN architectures to achieve a more precise prediction as indicated by a lower residual variance. The residuals are then quantized and encoded using the Huffman coding. The proposed scheme is empirically compared with the LPC using a real seismic dataset.

Methodologies

A typical prediction based compression scheme is illustrated in Figure 1. The scheme is implemented in two-stages. In the first stage, a portion of the seismic data samples is used for training the predictor until the goal is reached. The weights or coefficients of the trained predictor are sent to the receiver side for reconstruction. The first n samples of each trace are also sent to the receiver as the initial data segments for prediction, where n is the order of prediction (Kannan & Eswaran, 2007).

Prediction task is a supervised learning problem. Given input vector $X = [x_{m+1}\ x_{m+2} \dots\ x_{m+n}]$ and output $Y = [x_{m+n+1}]$ pairs obtained from the training data set $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_k, Y_k)\}$, it learns the best parameters for predicting model $\hat{Y} = h(X)$ that minimizes the loss function, i.e. $e(Y, \hat{Y}) = \frac{1}{2}(Y - \hat{Y})^2$.

In the second stage, the generated residuals are quantized using K-means algorithm. The quantized residuals are further encoded with Huffman coding. The encoded residual

10.1190/segam2019-3205392.1

sequences are then transmitted to the receiver side and decoded to reconstruct the data.

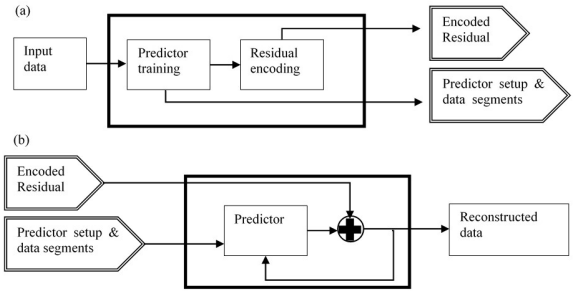

Figure 1: Prediction based method (a) sender side and (b) receiver side

Classical prediction based compression using the LPC

The LPC uses the auto-regressive (AR) model to represent discrete time series as $\hat{x}(k) = \sum_{i=1}^{p} a_i x(k-i)$, where p is the order of the AR model and a_is are the LPC coefficients, which are obtained by minimizing mean squared fitting errors. The corresponding modeling residual signal $e_L(k) = x(k) - \hat{x}(k)$ representing the predictive error has a lower entropy than $x[k]$. The lossless LPC uses p coefficients a_is, the first p time samples $x(1)$, $x(2)$, $\cdots$, $x(p)$, along with losslessly decoded residual samples $e_L(p+1)$, $e_L(p+2)$, $\cdots$, $x(N)$, to reconstruct the original signal $x[k]$. To achieve a higher compression ratio, the residual samples have to be compressed in a lossy way. However, lossy compression of residual samples results in accumulation of the reconstruction errors. Therefore, to avoid error accumulation in practice, the compressor (or, sending node) emulates the reconstruction procedure of the decompressor (or, receiving node) to calculate the residual from the estimates of the reconstructed samples.

The quantization error during residual quantization is also expected to be larger. The predictive and quantization errors are accumulated in the reconstructed trace and the decompressed signal may largely diverge. Therefore, we normalize each trace to reduce the quantization errors and to prevent the diverging reconstruction.

Prediction using DNN

In this section, we described the DNN as a time series predictor. Figure 2 shows the DNN for prediction which comprises an input layer with n units, where n is the order of prediction, an output layer with a single unit, and L hidden layers $H_1, H_2, \ldots,$ and H_L. We used the same prediction based strategy as the LPC, where the residual is quantized. The predicted value enhanced with the quantized residual becomes the input to predict the next value. The DNN is expected to achieve more precise predictions such that the variance of the residual becomes smaller.

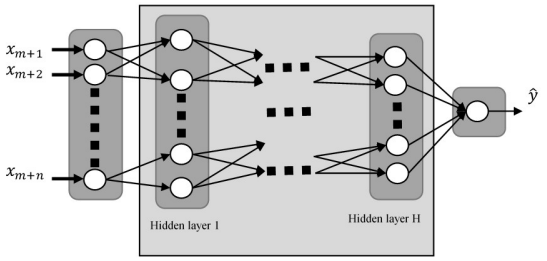

Figure 2: A DNN with multiple hidden layers

The DNN can model extremely complex and non-linear relationships between inputs and outputs from training data. A DNN is generally difficult to train since the error gradient would vanish as the number of hidden layers increases. A recent work on deep learning has reported that the DNN can be trained layer-wise to achieve good results for many machine learning tasks such as recognition and classification (Hinton & Salakhutdinov, 2006; Hinton, et al., 2012). The idea of deep learning is to learn the data structure by pre-training a neural networks with multiple hidden layers in an unsupervised way using large amounts of data. The neural networks is further fine-tuned using target data for supervised training to slightly adjust the learned features for more precise prediction (Huang, et al., 2014).

Pre-training is performed by decomposing each layer as an input-output pair. Each pair is then treated as an Restricted Boltzmann Machine (RBM). Each RBM consists of two layers namely visible $\boldsymbol{v}$ and hidden $\boldsymbol{h}$ (Figure 3). The biases $\boldsymbol{b}$ and $\boldsymbol{c}$ denote the visible and hidden layer biases. The visible and hidden units are connected with weights $\boldsymbol{W}$. The initial values of the weights and biases are assumed to be random with Gaussian distributions $N(0, 0.1)$.

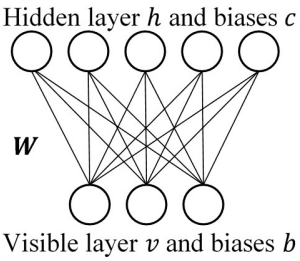

Figure 3: An RBM architecture

In the positive phase, we set the training vector $\boldsymbol{x}$ as the values of the visible units ($\mathbf{v}^+$). The value of each hidden unit ($h_j{}^+$) is set to one with probability

$$\hat{h} = p(h = 1|v) = \sigma\left(c + \sum v\ w_{ij}\right) \quad (1)$$

where $\sigma(x)$ is the logistic sigmoid function $1/(1+\exp(-x))$. The values from the positive phase ${h_j}^+$ are used to calculate the values in the negative phase as follows:

$$\hat{v}_i^- = \sigma\left(b_i + \sum_j h_j^+ w_{ij}\right) \quad (2)$$

$$\hat{h}_j^- = \sigma\left(c_j + \sum_i \hat{v}_i^- w_{ij}\right) \quad (3)$$

Finally, the updates for the weights and biases are given by

$$\Delta w_{ij} = \epsilon\left(\langle v_i \hat{h}_j\rangle^+ - \langle \hat{v}_i \hat{h}_j\rangle^-\right) \quad (4)$$

$$\Delta b_i = \epsilon\left(\langle v_i\rangle^+ - \langle \hat{v}_i\rangle^-\right) \quad (5)$$

$$\Delta c_j = \epsilon\left(\langle \hat{h}_j\rangle^+ - \langle \hat{h}_j\rangle^-\right) \quad (6)$$

where $\langle\cdot\rangle$ denotes expectation operator over all data and ϵ is the learning rate. Superscript + and – indicate the positive and negative phase, respectively. This process is repeated for a pre-determined maximum number of iterations. The same procedure is applied for the next RBM using the outputs of the previous RBM as inputs to the current one. This process continues until the last RBM. The weights and biases obtained from this pre-training process are used as initial values for a feed-forward neural networks obtained by stacking the RBM pairs in sequence. This resultant neural networks is trained using gradient based back-propagation for fine tuning. The training process continues until the error between the predicted output and the actual value reaches a pre-determined value or a maximum number of iterations is reached.

Residual encoding and the Huffman encoding

The residuals of the prediction are then quantized and encoded into fixed-length code-words. Precise predictions yield a low residual variance energy. Therefore, the residual can be quantized to a lower quantization level. This step can be improved further since the strings with different number of occurrences can be efficiently encoded using entropy based coding. The Huffman encoding (Huffman, 1952) is used in our experiment as an entropy coding that exploits non-uniformity of symbols to achieve a shorter average code-word length (Harnik, et al., 2014). This algorithm allows the symbols with higher occurrences to have shorter code-words and vice versa (Salomon, 2007).

Experimental results

We investigate the performance of our scheme using the East Texas dataset. The proposed scheme using the DNN predictor is applied on East Texas seismic dataset and its performance is evaluated and compared with the LPC. In our experiments, we normalized each trace between 0 and 1 as follows

$$x_0 = \frac{(x - x_{min})}{(x_{max} - x_{min})}, \quad (7)$$

where x and x_0 denote the original and normalized trace, respectively. Scalars x_{max} and x_{min} denote the maximum and minimum value in the trace. Both of these values must be sent to the receiver side to recover the trace to the actual scale.

In our experiments, we divided our data equally into training, cross validation, and testing. Based on performance on cross validation data, we used 20 iterations for pre-training for each RBM with a unified learning rate of 0.1 for all weights and biases. Three RBMs are pre-trained to generate initial weights and biases for three hidden layers with 30-20-10 units. All hidden units used sigmoid activation units. The fine-tuning is done using the standard back-propagation technique with a learning rate of 0.001 and a maximum number of iterations of 100.

The compression ratio is defined as the ratio of the original data volume and the compressed data volume. In our implementation, a single DNN is a universal predictor for all traces since it learned the features from all the training data. However, the LPC coefficients is only optimal for each trace since the AR-model is derived for each trace individually. Therefore, the compression ratio of the LPC is defined as

$$CR_{LPC} = \frac{Nl}{(2n+2)l + (N-n)l_q} \quad (8)$$

where N, n, and l_q denote the trace length, initial data segment length, and code-word size. The quantization levels L_Q determine the code-word length as follows $l_Q = \log_2 L_Q$. Typical seismic data format uses $l = 32\ bits$ for each sample. With the same notations, the compression ratio of the DNN based compression is simply

$$CR_{DNN} = \frac{Nl}{(n+2)l + (N-n)l_q} \quad (9)$$

The reconstruction quality is measured using the signal to noise ratio (SNR) as follows

$$\text{SNR} = 10\log_{10}\left[\frac{\sum_{m=1}^{N} x^2(m)}{\sum_{m=1}^{N}\left(x(m) - \hat{x}(m)\right)^2}\right] \quad (10)$$

where x and $\hat{x}$ denote the original and the reconstructed data.

Table 1: SNR of the LPC and the DNN

N	LPC (dB)			DNN (dB)		
	L_Q					
	16	32	64	16	32	64
5	*23.1*	28.96	30.27	43.02	52.15	58.31
10	*26.56*	27.64	29.55	42.07	51.45	57.41
15	25.86	*28.69*	29.88	42.81	52.81	58.82

We then pre-train the DNN with segments of seismic traces and fine-tune it, in a supervised manner, to predict next sample. The residuals distribution of the prediction using DNN and LPC are shown Figure 4. Note that the residuals are obtained using normalized original data, The normalized original data is more uniformly distributed than that of the residual predictors. It can be noticed that the DNN exhibits

10.1190/segam2019-3205392.1

the lowest residual variance as indicated by a compact distribution. A lower variance indicates that the residuals can be quantized with a better precision than that of lower variance with the same quantization levels. From the information theory point of view, a lower variance yields a lower entropy that allows a higher compression ratio.

Three codebooks of the residual encoding are constructed using the residuals of the training data where the first, second, and third codebooks implement quantization levels of $L_Q = 16, 32, 64$. Table 1 shows the SNR of the DNN and the LPC for data segment lengths $n = 5, 10, 15$. It can be noticed that the DNN outperforms the LPC significantly for all configurations. Increasing quantization levels obviously increase the reconstruction quality for both methods. The reconstruction quality of the DNN averagely increases by $7dB$ for each quantization levels. The LPC suffers from diverging when the quantization levels are not big enough as indicated by italic faces. Both techniques do not show significant improvements when the initial segment lengths are increased.

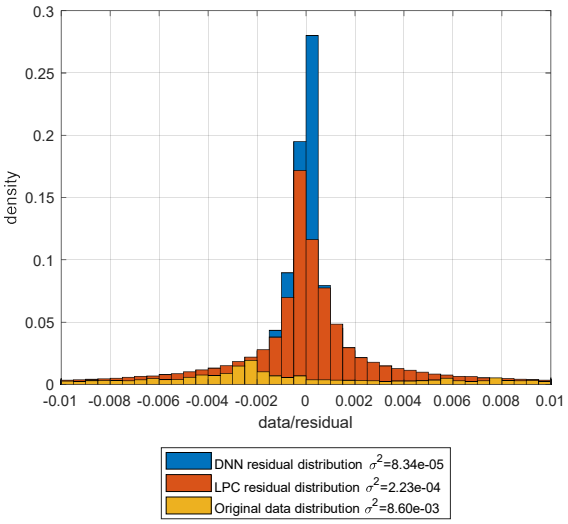

Figure 4: Original data and prediction residual distribution

Prediction based compressions heavily depend on the residuals encoding. The simplest way is to apply uniform encoding scheme or fixed length code-words to all quantized residuals. Such encoding however contributes to the major portion of the compressed data overhead. Therefore, we apply the Huffman algorithm to the quantized residuals to achieve higher compression ratios. Table 2 shows the compression ratios of uniform and the Huffman encodings. It can be noticed that the algorithm improves the compression ratios. A reconstruction sample is shown in Figure 5. The sample shows minor differences between original and reconstruction.

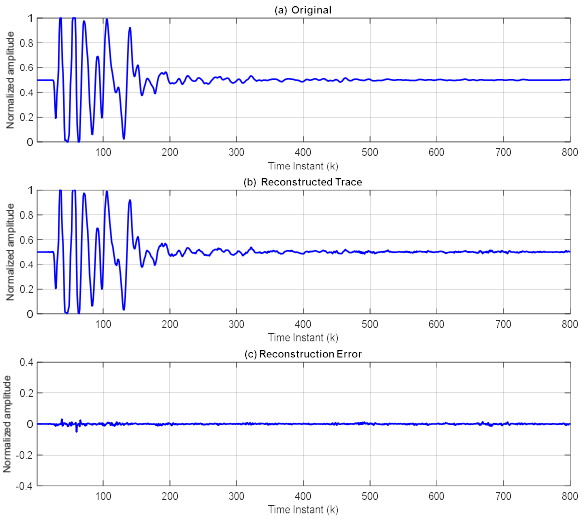

Figure 5: Reconstruction Sample (a) original (b) reconstructed using DNN (c) reconstruction error

Table 2 Compression ratio of the DNN

CR	$L_Q = 16$			$L_Q = 32$			$L_Q = 64$		
n	5	10	15	5	10	15	5	10	15
Uniform	7.8	7.6	7.4	6.3	6.2	6.07	5.3	5.2	5.1
Huffman	14.4	13.3	12.2	11.04	10.7	9.6	8.75	8.7	7.9

Conclusion

In this paper, we presented a prediction based seismic data compression scheme using deep neural networks (DNN). First, a DNN with multiple hidden layers was pre-trained using restricted Boltzmann machines (RBMs) to obtain near optimal initial weights. The DNN was then fine-tuned in a supervised fashion to achieve a higher prediction precision. The residual between actual and predicted samples were quantized to achieve a smaller data representation. Our experiments with a real data set showed that the DNN significantly outperformed the LPC in term of reconstruction quality. The quantized residuals were further encoded using the Huffman coding to reduce the average code-word length. The DNN with the encoded residuals achieved more than 40dB of SNR with a compression ratio of more than 10:1 without any significant distortion in the reconstructed data.

Acknowledgment

This work was supported by the Center for Energy and Geo Processing, King Fahd University of Petroleum and Minerals, under Project GTEC-1801.

10.1190/segam2019-3205392.1

REFERENCES

Al-Moohimeed, M. A., 2004, Towards an efficient compression algorithm for seismic data: Proceedings of the Asia-Pacific Radio Science Conference, 550–553, https://doi.org/10.1109/aprasc.2004.1422555.

Averbuch, A. Z., F. Meyer, J.-O. Stromberg, R. Coifman, and A. Vassiliou, 2001, Low bit-rate efficient compression for seismic data: IEEE Transactions on Image Processing, **10**, 1801–1814, https://doi.org/10.1109/83.974565.

Bernasconi, G., and M. Vassallo, 2003, Efficient data compression for seismic-while-drilling applications: IEEE Transactions on Geoscience and Remote Sensing, **41**, 687–696, https://doi.org/10.1109/tgrs.2003.808896.

Fajardo, C., O. Reyes, and A. Ramirez, 2015, Seismic data compression using 2D lifting-wavelet algorithms: Ingenieria y Ciencia, **11**, 221–238, https://doi.org/10.17230/ingciencia.11.21.11.

Harnik, D., E. Khaitzin, D. Sotnikov, and S. Taharlev, 2014, A fast implementation of deflate: Data Compression Conference, 223–232, https://doi.org/10.1109/dcc.2014.66.

Hinton, G., L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, 2012, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups: IEEE Signal Processing Magazine, **29**, 82–97, https://doi.org/10.1109/msp.2012.2205597.

Hinton, G. E., and R. R. Salakhutdinov, 2006, Reducing the dimensionality of data with neural networks: Science, **313**, 504–507, https://doi.org/10.1126/science.1127647.

Huang, W., G. Song, H. Hong, and K. Xie, 2014, Deep architecture for traffic flow prediction: Deep belief networks with multitask learning: IEEE Transactions on Intelligent Transportation Systems, **15**, 2191–2201, https://doi.org/10.1109/tits.2014.2311123.

Huffman, D. A., 1952, A method for the construction of minimum-redundancy codes: Proceedings of the IRE, **40**, 1098–1101, https://doi.org/10.1109/jrproc.1952.273898.

Kannan, R., and C. Eswaran, 2007, Lossless compression schemes for ECG signals using neural network predictors: EURASIP Journal on Applied Signal Processing, **2007**, 102–102, https://doi.org/10.1155/2007/35641.

Lindstrom, P., P. Chen, and E.-J. Lee, 2016, Reducing disk storage of full-3D seismic waveform tomography (F3DT) through lossy online compression: Computers & Geosciences, **93**, 45–54, https://doi.org/10.1016/j.cageo.2016.04.009.

Liu, B., M. Mohandes, H. Nuha, M. Deriche, and F. Fekri, 2018, A distributed principal component analysis compression for smart seismic acquisition networks: IEEE Transactions on Geoscience and Remote Sensing, **56**, 3020–3029, https://doi.org/10.1109/tgrs.2018.2789354.

Nave, G., and A. Cohen, 1993, ECG compression using long-term prediction: IEEE Transactions on Biomedical Engineering, **40**, 877–885, https://doi.org/10.1109/10.245608.

Nuha, H., B. Liu, M. Mohandes, and M. Deriche, 2017, Seismic data compression using signal alignment and PCA: IEEE.

Rosten, T., T. A. Ramstad, and L. Amundsen, 2004, Optimization of sub-band coding method for seismic data compression: Geophysical Prospecting, **52**, 359–378, https://doi.org/10.1111/j.1365-2478.2004.00422.x.

Rubin, M. J., M. B. Wakin, and T. Camp, 2016, Lossy compression for wireless seismic data acquisition: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, **9**, 236–252, https://doi.org/10.1109/jstars.2015.2459675.

Salomon, D., 2007, A concise introduction to data compression: Springer Science & Business Media.

Spanias, A. S., S. B. Jonsson, and S. D. Stearns, 1991, Transform methods for seismic data compression*:* IEEE Transactions on Geoscience and Remote Sensing, **29**, 407–416, https://doi.org/10.1109/36.79431.

Stearns, D. S., L. Tan, and N. Magotra, 1993, Lossless compression of waveform data for efficient storage and transmission: IEEE Transactions on Geoscience and Remote Sensing, **31**, 645–654, https://doi.org/10.1109/36.225531.

Villasenor, J. D., R. A. Ergas, and P. L. Donoho, 1996, Seismic data compression using high-dimensional wavelet transforms: Proceedings of Data Compression Conference, 396–405, https://doi.org/10.1109/dcc.1996.488345.

Wu, W., Z. Yang, Q. Qin, and F. Hu, 2006, Adaptive seismic data compression using wavelet packets: IEEE International Symposium on Geoscience and Remote Sensing, 787–789, https://doi.org/10.1109/igarss.2006.202.