Well-Log Estimation from Seismic Data Using Encoder-Decoder

Giovanny Meneses Arboleda[1], Luiz Trindade Santos[1,2], Carlos Rodriguez Suarez[1], Roberto Quevedo Quispe[1], Marcelo Gattass[1,2], [1] Tecgraf/Puc-Rio, [2] Pontifical Catholic University of Rio de Janeiro

Abstract

The seismic inversion problem may allow us to obtain the petrophysical properties of underground layers from seismic reflection data. The knowledge of these characteristics is of vital importance for the identification of regions with hydrocarbons. Different approaches using Machine Learning algorithms have been proposed to address this issue, such as feed-forward neural networks, Support Vector Machine (SVM), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Temporal Convolutional Networks (TCN). This work proposes an encoder-decoder network architecture to overcome this problem, with a convolutional neural network as encoder and an LSTM network as decoder. The proposed method was tested to estimate the density, P-impedance, and sonic well-logs using the seismic data from the offshore Netherlands F3 block.

Introduction

The process of obtaining petrophysical properties of the subsurface structures and their spatial variability is called Reservoir Characterization (RC). Usually, the process begins after a seismic and geological survey, using data from well-logs, core analysis, and seismic reflection. This process allows us to predict the subsurface fluid flow, and reservoir performance to optimize the scavenging efficiency Yu et al. (2011). Several works have been proposed to solve this problem, but its difficulty and intrinsic uncertainties do not allow the generation of a systematic method that can be used outside of certain study areas. Machine learning algorithms (ML) are well known for their large capacity to generalize, and they show promise in the RC field, as has been shown in recent works allowing us to use seismic data to infer several types of well-logs. Alfarraj et al. (2018) use a Recurrent Neural network to estimate density and p-impedance. Biswas et al. (2019) use a Convolutional Neural Network (CNN), and a physical guide to estimate p-wave, s-wave, and density. Das et al. (2018) use a CNN to estimate seismic impedance.

Recurrent Neural Networks

Unlike traditional feedforward networks, recurrent neural networks (RNN) have recurrent connections to save information from recent events and contextualize the information. Commonly used in language models, natural language processing, speaking, audio, and video Shinde et al.). However, these kinds of networks have a larger number of parameters, which makes them harder to train. Additionally, with very large context intervals, the RNN suffers from Exploding Gradient e Vanish Gradient Bengio et al. (1994). To overcome this issue, different architectures have been proposed, such as Gate Recurrent Unit (GRU) Cho et al. (2014) and the Long-Short Term Memory (LSTM) Hochreiter et al. (1997). An LSTM module (Figure 1) has three gates that manipulate the internal state c_t and the output y_t

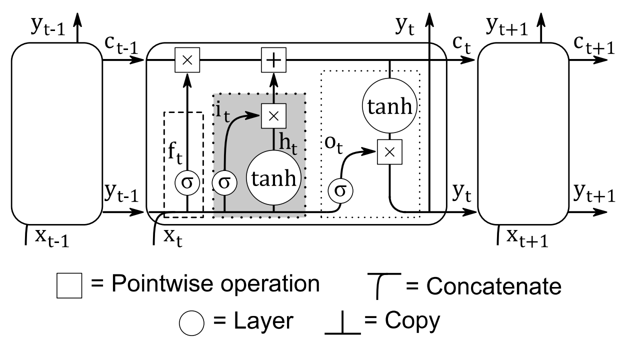

Figure 1 *– LSTM repetition model and its internal gates.*

The first gate (traced-square), called Forget Gate (f_t), chooses the information to be discarded. The output of the previous memory block (y_{t-1}) is taken and concatenated with the input of the cell (x_t) and then passed through a sigmoid function $(\sigma())$. The second gate (dotted dark-squared) is called Learning/Input Gated (i_t). Whose role is to decide what new information will be stored in the cell state, taking the output of the previous cell and the input and passing it through by a sigmoid function and, at the same time, through a hyperbolic tangent function. The third gate (dotted-square), called Output Gate (O_t), also uses the input of the cell (x_t), the internal state of the previous cell (y_{t-1}), and a sigmoid function. Finally, the new output (y_t) is updated using the output cell and its internal state (c_t). The equations in their compacted form SANTOS (2019) can be described, as

$$\begin{gathered} f_t = \sigma\left(W_f \cdot [y_{t-1}, x_t] + b_f\right) \\ i_t = \sigma(W_i \cdot [y_{t-1}, x_t] + b_i) \\ h_t = \tanh(W_c \cdot [y_{t-1}, x_t] + b_C) \\ c_t = f_t \odot c_{t-1} + i_t \odot h_t \\ o_t = \sigma(W_o \cdot [y_{t-1}, x_t] + b_o) \\ y_t = o_t \odot \tanh(c_t) \end{gathered} \tag{1}$$

Where W_f, W_i, W_c, W_o and b_f, b_i, b_c, b_o are the weights matrices and bias vector parameters, respectively, that are learned during training

Convolutional Neural Networks

Convolutional Neural Networks (CNN) are networks that use a spatial relationship. A general description of a CNN is given by ROSA (2018)

$$x^1 \to \psi_1(x^1; W^1) \to x^2 \to \psi_2(x^2; W^2) \to \cdots \to x^n \to \psi_n(x^n; W^n) \tag{2}$$

where the input x^m goes through the layers ψ_m until it reaches the final one ψ_n, the functions $\psi_m(.)$ are usually linear functions followed by nonlinear functions, and the W^m are the adjustable parameters of the layers. The output of each layer is the input of the next. Different layers are commonly used.

Convolutional layer: this layer applies different types of filters to the input. Filters are an array of weights W with size $(\omega \times \omega \times d)$ that interpolate a d-dimensional feature map to a k-dimensional feature map with an input size of $(n \times n \times d)$, centered in (i, j). The response of the f-th filter is given by

$$x_{i,j,r}^{l+1} = \sum_{c=1}^{d} \sum_{q=0}^{\omega-1} \sum_{v=0}^{\omega-1} \left(W_{v,q,c,r}^{l} \times x_{v,q,c}^{l}\right)_{i.j} + \beta \tag{3}$$

where β are the trainable weights, and l is the number of layers. The output for each convolutional layer for all the k filters in the entire spatial arrangement (i, j) is

$$\left(\frac{m - \omega + 2zp}{s} + 1\right) \times \left(\frac{n - \omega + 2zp}{s} + 1\right) \times k - \dim \tag{4}$$

Nonlinear layer: the vanish gradient problem appears where the magnitude of the layer`s learning gradient has been reduced until its loss. To address this problem, a nonlinear activation function is usually used. One of the most used functions is the Rectified Linear Unit ReLU Nair et al.). The function replaces the negative elements of the feature map with zero and is described as

$$ReLU(x) = \max(0, x) \tag{5}$$

Pooling layer: generate a low-resolution version of the input to obtain its main elements. In order to prevent the model from being susceptible to physical changes while preserving the discrimination. Goodfellow et al. (2016), this layer is similar to the convolutional layer but uses pooling functions as the average or the maximum value.

Fully connected layer. This layer implies that all neurons in the previous layer are connected to all neurons in the next layer. Finally the loss layer. This layer contains a loss function, which is minimized using back propagation while the algorithm is trained. The kind of function used depends on the type of problem. Here, because it is a regression problem, we use the Mean Squared Error (MSE), which computes the mean of square errors between the predicted and the actual values and is expressed as

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(Y_1 - \hat{Y}_i\right)^2 \tag{6}$$

Method

Pre-processing

Firstly, it is necessary to define the volume around the well. Use x as the number of inlines / cross-lines as the length of each side of a square, and T as the range depth/time. Next, knowing the depth is needed to synchronize the seismic data and the well-log. Finally, normalize the well`s volume between 1 and -1. On the other hand, for the well-log preprocess, subsampling, and filtering, the subsampling was made using Inverse Distance Weighting (IDW) Shepard) and a low-pass Butterworth filter.

Proposed Model

The proposed architecture is an encoder-decoder (Figure 2), and its input is a seismic trace matrix (sub-volume inlines) with x features columns and T z-lines rows. These inlines are used one at a time. The encoder is composed of two convolutional layers activated by the ReLU function; both have 64 filters and a kernel size of 3, followed by a MaxPooling layer. The first convolutional layer's input and the output have skipped connections and are concatenated with the encoder output. The bottleneck (in light blue) comprises a Flattening and a RepeatVector. The RepeatVector repeats the input T times. The decoder has one LSTM with 200 neurons with a dropout layer, deactivating 5% of the units. Followed by a fully connected layer with 100 neurons, and finally, the output has a regression layer composed of one neuron. The neuron is responsible for presenting the inferred Well-Log.

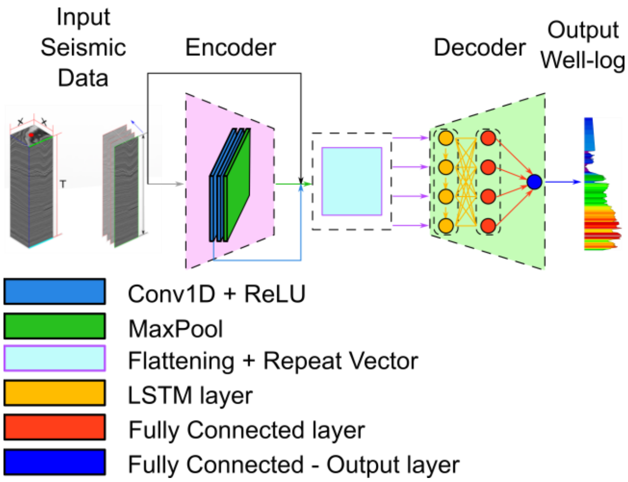

***Figure 2 –** Proposed encoder-decoder architecture.*

Experimental Evaluation

The seismic data used are from the offshore F3-block in the Netherlands TerraNubis (1987), with $386.93\ Km^2$, 651 inlines, and 951 cross-lines. The dataset is composed of 4 wells (F02-1, F03-2, F03-4, and F06-1), as shown in Figure 3. This dataset has 11 different well logs, including Density, Sonic, P-impedance.

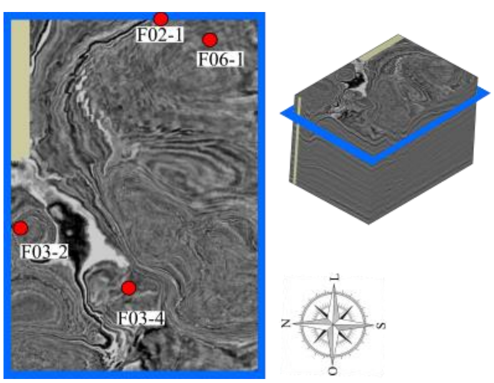

***Figure 3** – Offshore F3-block, Netherlands.*

The Dataset

The sub-volume used has 11×11 traces $(x = 11)$ centered at the well and the working depth (T) is between $600 - 1300\ ms$. Since the model is trained with the inlines. The database has 44 samples, of which 33 were used to perform the training process (3 wells) and 11 samples for testing. Validation uses 21% of the train samples. The sub-sampling of the well-log was performed by interpolating the midpoint every $4\ ms$. The filtering was performed using the Scipy community (2023) library, a low-pass Butterworth filter with order 3, power of 1, and a critical frequency of 4. To measure the performance of the model, the Pearson correlation is used Pearson (1895).

The operator defines the relation between the covariance and the standard deviation of two random variables (X, Y)

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad (7)$$

Experimental Evaluation

To test the performance of the proposed model. A 4-fold validation is performed. Every fold uses 3 wells for the training and one for the test (green), as shown in Table 1.

	Poço 1	Poço 2	Poço 3	Poço 4
Test 1	F06-1	F03-4	F03-2	F02-1
Test 2	F06-1	F03-4	F03-2	F02-1
Test 3	F06-1	F03-4	F03-2	F02-1
Test 4	F06-1	F03-4	F03-2	F02-1

***Table 1 –** Test K-fold.*

The experiment was repeated 10 times, and the results are average for all experiments. Table 2 shows a summary of the results (in percentage) and a comparison with those presented by Alfarraj et al. (2018). There is shown the correlation of the Density, P-impedance, and sonic well-log. As can be seen, the proposed model has achieved a relatively high correlation overcoming their results without using augmentation data techniques. It should be noted that the estimation of petrophysical properties is not an easy task. However, better scores could be obtained using a larger database or more complex networks.

Property / Model	Density	P-impedance	Sonic
(Alfarraj and AlRegib, 2018)	70	72	-
Proposed Model	74 ± 6	76 ± 6	74 ± 6

***Table 2 –** Correlation coefficient (%) for Density, P-impedance, and Sonic logs.*

Figure 4 shows the predicted (blue) and measured (orange) density well-log for the best k-fold found using the proposed model. As can be seen, the model infers low frequencies better than high frequencies but has problems with difficulties with pronounced changes.

Conclusions

This paper proposes an encoder-decoder model to estimate well-log properties from seismic data using CNN and RNN, respectively. The model was validated using 4-fold validation and tested with three different properties: density, sonic, and P-impedance. The results show that the model has a good inference capacity standing out in low frequencies, beating results found in the literature. The response of the model could, perhaps, be improved by using a larger dataset.

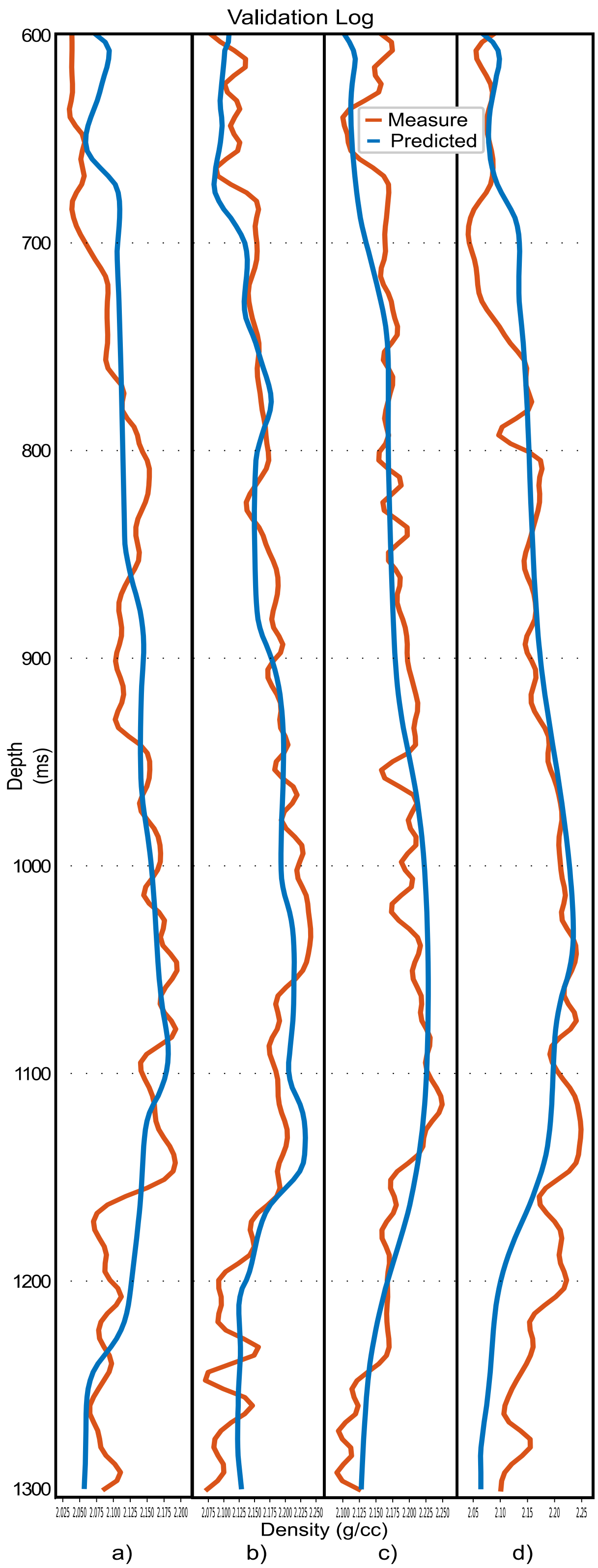

Figure 4 *– Density well-log for a) F02-1, b) F03-2, c) F03-4, and d) F06-1.*

Acknowledgments

This work has been supported by Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) and instituto de Desenvolvimento de Software Técnico-Científico da PUC-Rio (Tecgraf PUC-Rio).

References

ALFARRAJ, M. and ALREGIB, G., 2018. Petrophysical property estimation from seismic data using recurrent neural networks. *SEG Technical Program Expanded Abstracts*, pp. 2141–2146, Available at: DOI 10.1190/SEGAM2018-2995752.1.

BENGIO, Y., SIMARD, P., and FRASCONI, P., 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Available at: DOI 10.1109/72.279181.

BISWAS, R., SEN, M.K., DAS, V., and MUKERJI, T., 2019. Prestack and poststack inversion using a physics-guided convolutional neural network. *http://www.seg.org/interpretation*, vol. 7, no. 3, pp. SE161–SE174, Available at: DOI 10.1190/INT-2018-0236.1.

CHO, K., MERRIËNBOER, B. VAN, BAHDANAU, D., and BENGIO, Y., 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Available at: DOI 10.48550/arxiv.1409.1259.

DAS, V., POLLACK, A., WOLLNER, U., and MUKERJI, T., 2018. Convolutional Neural Network for Seismic Impedance Inversion. *SEG Technical Program Expanded Abstracts*, pp. 2071–2075, Available at: DOI 10.1190/SEGAM2018-2994378.1.

GOODFELLOW, I., BENGIO, Y., and COURVILLE, A., 2016. *Deep Learning*. Publisher: MIT Press book.

HOCHREITER, S. and SCHMIDHUBER, J., 1997. Long Short-term Memory. *Neural computation*, vol. 9, pp. 1735–1780, Available at: DOI 10.1162/neco.1997.9.8.1735.

NAIR, V. and HINTON, G.E., Rectified Linear Units Improve Restricted Boltzmann Machines. *ICML* Available from: httpsicml.ccConferences2010papers432.pdf.

PEARSON, K., 1895. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242.

ROSA, L.E.C.L., 2018. *Crop Recognition From Multitemporal Sar Image Sequences Using Deep Learning*

Techniques. Publisher: PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO.

SANTOS, L.F.T., 2019. *Detector De Assinaturas De Gás Em Levantamentos Sísmicos Utilizando LSTM*. Publisher: PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO.

SCIPY COMMUNITY, 2023. Getting started — SciPy v1.10.1 Manual. [online] Available from: https://docs.scipy.org/doc/scipy/getting_started.html [viewed 3 May 2023].

SHEPARD, D., A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 1968 23rd ACM national conference* Available from: https://doi.org/10.1145/800186.810616 [viewed 14 December 2022].

SHINDE, P.P. and SHAH, S., A Review of Machine Learning and Deep Learning Applications. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*.

TERRANUBIS, 1987. TerraNubis - Data Info of F3 Demo 2020. [online] Available from: https://terranubis.com/datainfo/F3-Demo-2020 [viewed 5 January 2023].

YU, X., MA, Y.Z., GOMEZ, E., PSAILA, D., POINTE, P.L., and LI, S., 2011. *Reservoir Characterization and Modeling: A Look Back to See the Way Forward*. In: Y.Z. Ma and P.R. La Pointe, eds. *Uncertainty Analysis and Reservoir Modeling: Developing and Managing Assets in an Uncertain World*. Publisher: American Association of Petroleum Geologists.