

Development of a Multi-attribute Convolutional Neural Network to Seismic Facies Classification

Igor B. Oliveira, Igor Braga, João Puga, Anderson Franco, Luciano Pereira and Geovane Ouverney, Invision Geophysics

Copyright 2019, SBGf - Sociedade Brasileira de Geofísica

This paper was prepared for presentation during the 16th International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 19-22 August 2019.

Contents of this paper were reviewed by the Technical Committee of the 16th International Congress of the Brazilian Geophysical Society and do not necessarily represent any position of the SBGf, its officers or members. Electronic reproduction or storage of any part of this paper for commercial purposes without the written consent of the Brazilian Geophysical Society is prohibited.

Abstract

Seismic facies classification contributes to the interpretation and geological modeling of subsurface. The complexity of geological environments still challenges any standardization of patterns recognition methods for estimating seismic facies and the search for an ideal method, since this step can be complex depending on the purpose and the available dataset. The convolutional neural networks (CNN) were developed to serve as the computer vision, seeking to imitate the functioning of human vision. Despite the potential of this technology, the convolutional networks have not yet been successfully adapted to work with seismic data. The main objective is to develop a workflow based on multi-attribute convolutional neural network for seismic facies characterization. The workflow is based on four main steps: facies classification; data extraction; CNN train and prediction. Aiming to verify the efficiency of this new method, it was elaborated a synthetic image and it was also applied to a real seismic data. In addition, seismic attributes have been calculated to enhance characteristics of the original dataset. The method demonstrates exciting results achieving accuracies above 90% over the expected facies model.

Introduction

Artificial neural networks are computational models inspired by the nervous system of living beings. They can acquire and maintain knowledge; and can be defined as a set of processing units, represented by artificial neurons, interlinked by many interconnections, implemented by vectors and matrices of synaptic weights (Da Silva *et al.*, 2017). McCulloch and Pitts (1943) proposed the most used neuron model in different artificial neural network architectures, as can be seen in Figure 1. The topology of a given neural network, within a particular architecture, can be defined as the different structural compositions it can assume (Da Silva *et al.*, 2017).

Analyzing the Figure 1 it is possible to see that the artificial neuron is composed basically of seven elements, as described below (Da Silva *et al.*, 2017):

- i. Input signals: x_1, x_2, \dots, x_n , that are the signals or samples coming from the external environment;

- ii. Synaptic weights: w_1, w_2, \dots, w_n , the values used to weight each one of the input signal, which enables the quantification of their relevance to the neuron;
- iii. Linear aggregator: Σ , that gathers all input signals weighted by the synaptic weights to produce an activation voltage;
- iv. Activation threshold or bias: θ , is a variable used to specify the proper threshold that the result produced by the linear aggregator should have to generate a trigger value;
- v. Activation potential: u , the result produced by the difference between Σ and θ ;
- vi. Activation function: g , whose goal is limiting the neuron output within a reasonable range of values;
- vii. Output signal: y , consists on the final value produced by the neuron.

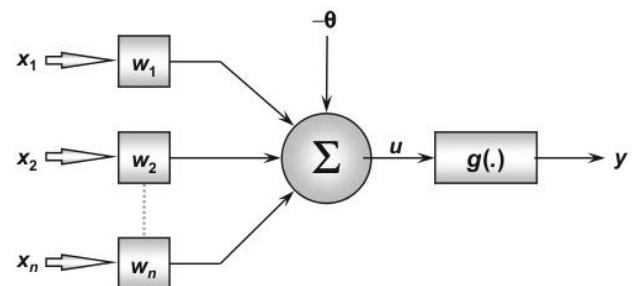


Figure 1 – Artificial neural. Source: Da Silva *et al.*, 2017

The Perceptron (Rosenblatt, 1958), is the simplest configuration of an artificial neural network ever created and it belongs to the class of single-layer feedforward architecture. In 1960, the Adaline, created by Widrow and Hoff, promoted some advancements to the artificial neural network area, like the learning algorithm named Delta rule, which is considered the precursor of the generalized Delta rule used on the training of the Multiple-Layer Perceptron (MLP) network (Da Silva *et al.*, 2017).

On the other hand, Convolutional Neural Networks – CNNs (LeCun, 1989) are a specialized kind of neural network that use convolution in place of general matrix multiplication for processing data that has a known grid-like topology. CNNs have been tremendously successful in practical applications (Goodfellow *et al.*, 2016).

Convolution is a mathematical operation on two functions of a real-valued argument that results in a third that measures the sum of the product of these functions along the region implied by the superposition of them. The

convolution operation is typically denoted with an asterisk, as shown in Equation (1).

$$s(t) = (x*w) (t) = \int x(a)w(t-a) da \quad (1)$$

In convolutional network terminology, the first argument (function “x”) is referred to as the input, and the function “w” as the kernel. The output “s” is the feature map. In machine learning, the input is usually a multidimensional array of data and the kernel of parameters. These arrays can be referred as tensors. See Figure 2 for an example of convolution (Goodfellow et al., 2016).

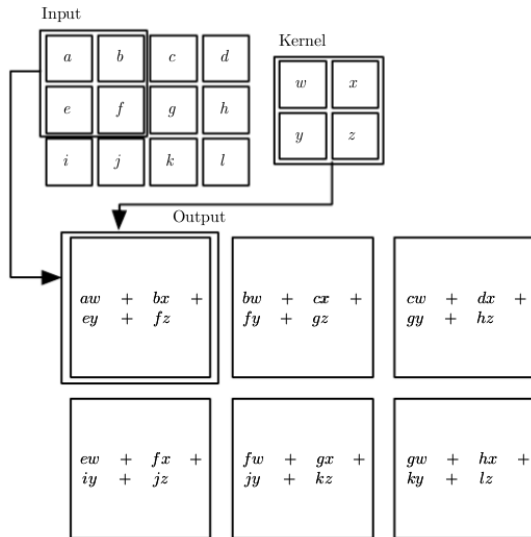


Figure 2 – An example of 2D convolution. Source: Goodfellow et al., 2016

After each convolutional layer, it is convention to apply a nonlinear layer or activation immediately afterward. The purpose of this layer is to introduce nonlinearity to a system that has just been computing linear operations. The ReLU layer applies the function $f(x) = \max(0, x)$ to all of the values in the input volume (Deshpande, accessed on November 2, 2017).

The pooling layer, also referred to as a downsampling layer, basically takes a filter and a stride of the same length and reduces the input size resulting in improved statistical efficiency and reduced memory requirements for storing the parameters, as indicated in Figure 3. Stride controls how the filter convolves around the input, in other words, the amount by which the filter shifts is the stride.

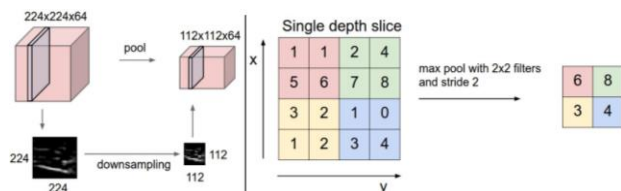


Figure 3 – Pooling layer downsamples the volume spatially. Source: Stanford course, accessed on November 6, 2017

Dropout layers have a very specific function in neural networks. This layer “drops out” a random set of

activations by setting them to zero, as shown in Figure 4. It forces the network to be redundant, by that the network should be able to provide the right classification or output for a specific example even if some of the activations are dropped out. It makes sure that the network is not being too “fitted” to the training data and thus helps alleviate the overfitting problem. This layer is only used during training, and not during test time (Deshpande, accessed on November 2, 2017).

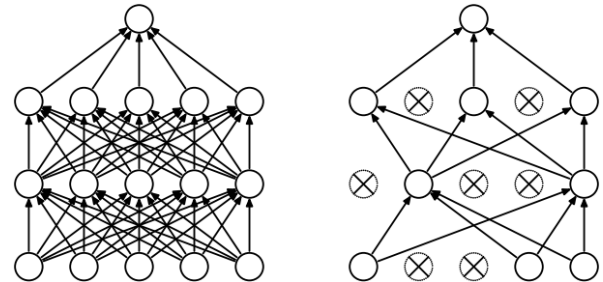


Figure 4 – Dropout neural net model. Left: A standard neural net with two hidden layers. Right: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped. Source: Srivastava et al., 2014

The fully connected layer is a traditional multi-layer perceptron that uses a softmax activation function in the output layer. The softmax function, or normalized exponential function, is a generalization of the logistic function (sigmoid) that squashes a vector of arbitrary real values to a vector of real values in the range [0,1] that add up to 1. The purpose of the fully connected layer is to use the features for classifying the input image into various classes based on the training dataset (Karn, accessed on November 15, 2017). Figure 5 presents a simple model of convolutional neural network.

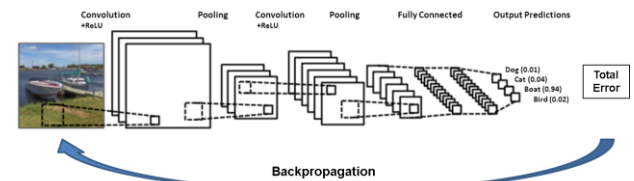


Figure 5 – A simple CNN model. Source: Modified of Clarifai, accessed on November 27, 2017

Through this technology, the main goal of this work is to develop a reliable methodology based on multi-attribute convolutional neural networks for seismic facies characterization at any geological background helping the interpretation and geological modeling even in complex structures.

Method

The methodology of this work is based on the development of a multiattribute convolutional neural network to seismic facies classification. The workflow of this new elaborated method is shown below, in Figure 6. The workflow can be divided according to the following steps: i) Facies classification; ii) Data extraction; iii) CNN train and iv) Prediction.

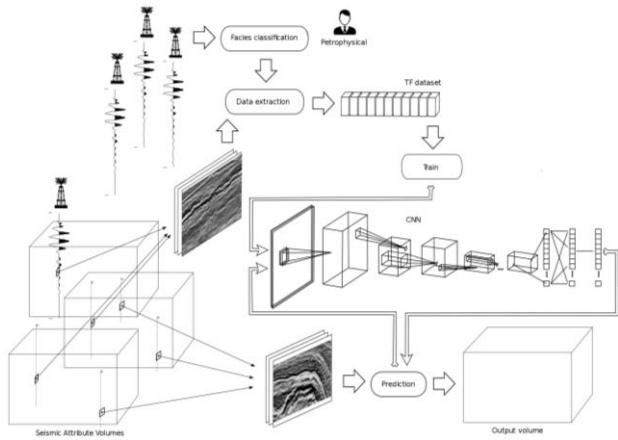


Figure 6 – Workflow of the multiattribute convolutional neural network to seismic facies classification

Facies classification is based on well attribute logs located in the area of interest. This process is performed by the petrophysicist and it has as product an attribute log indicating the occurrence of each facies through predetermined labels.

Data extraction is performed by an extractor that runs the seismic volumes marked by the track of each well collecting small seismic sections, assign the corresponding label found in the facies classification log and organizes them into different datasets. The data generated by the extractor may not be enough for the training of CNN, so it is applied data augmentation techniques.

CNN train is carried out from the dataset generated in the data extraction step. A determined number of iterations is parametrized so that satisfactory accuracy is obtained without overfitting the network.

Prediction is performed using the trained CNN. In this process, the classifier scans the entire seismic volumes collecting and grouping the small seismic sections according to the criteria adopted in the data extraction step. These data are passed through the trained CNN and the outputs generated are used in the assembly of the output seismic volume.

Aiming to develop the convolutional network, it was created a synthetic data from a layered grayscale image in regular steps, with 416 lateral samples and 615 depth samples. Each new layer of the image has a slightly darker shade of gray. Between each layer of the image (Figure 7) there are triangular shapes, which may occupy the upper layer, or lower, depending on the type of simulated patterns.

Four types of labels (synthetic patterns) were used, in the first the triangular shapes are formed in the bottom layer, overlapping the top layer in the tone of the bottom layer. In the second, the forms use the color of the top layer, marking the layer below with its color. The third and fourth labels do not have the geometric like others; they are composed of images of straight parts of the layers, as shown in Figure 8. Each of the labels composing the dataset were taken from five lines, generated from synthetic tracking and log data, as presented in Figure 9.

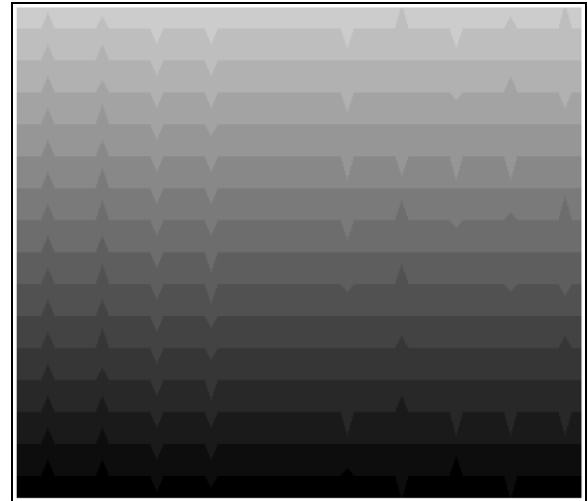


Figure 7 – Synthetic data from a layered grayscale image

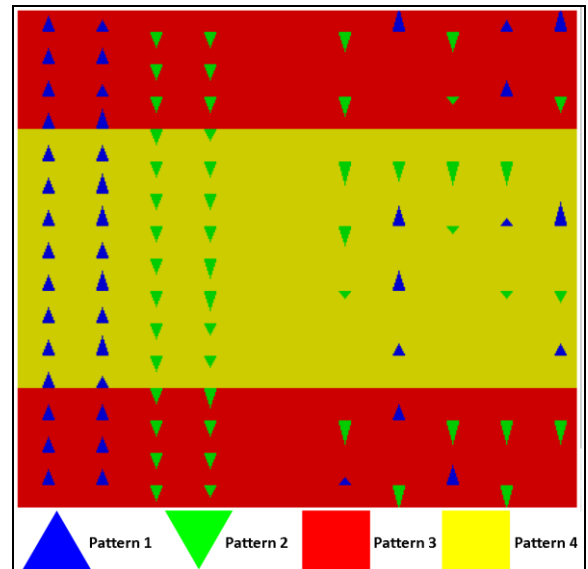


Figure 8 – Synthetic pattern classification model

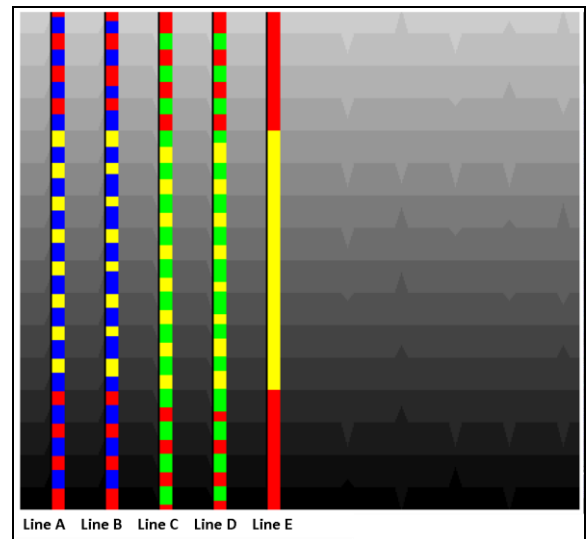


Figure 9 – Synthetic data including lines with the labels

To improve data distinction, image attributes have been calculated to enhance characteristics of the original data. The attributes generated were energy, similarity, instantaneous frequency, mode and Hilbert transform, resulting a six-dimensional image, considering also the synthetic data, as shown in Figure 10.

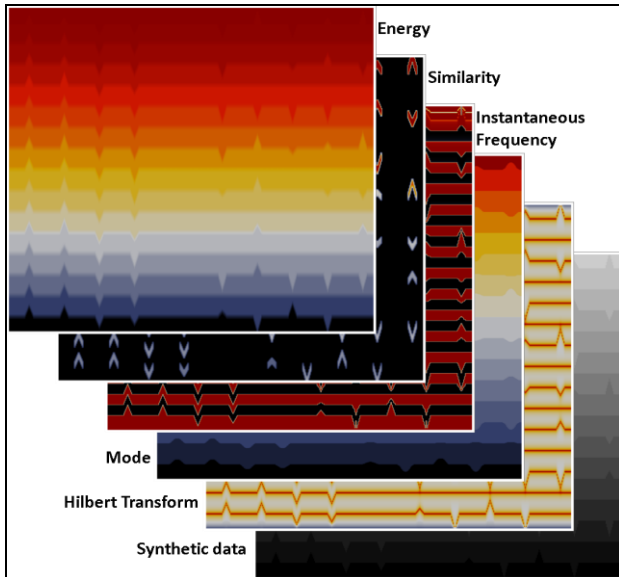


Figure 10 – Attributes generated based on synthetic data, resulting a six-dimensional image

As mentioned in data extraction explanations, the data generated by the extractor may not be enough for CNN training, so it is applied data augmentation techniques, as demonstrated in Figure 11. The increase of artificially data multiplies the original dataset and facilitates the search of the neural network in the synthetic data.

After data augmentation, the dataset is stored as a single block, but there is a separation of training, test and validation data into portions that can be selected by the network trainer. There is also option of balancing the number of labels of each type.

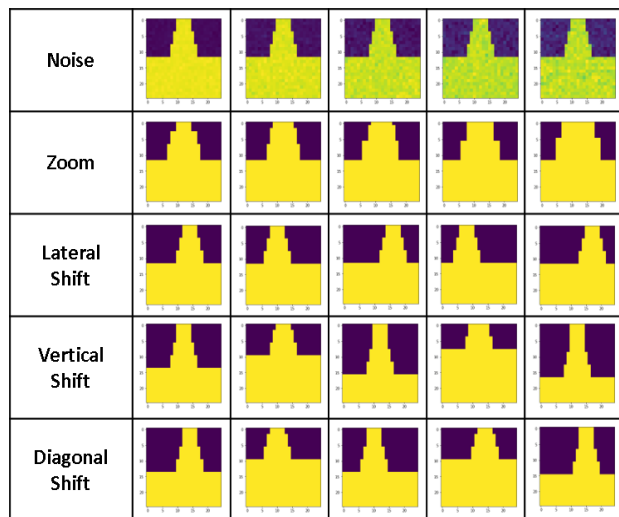


Figure 11 – Different type of data augmentation used in this work

The CNN architecture used to perform the training and prediction was calibrated through parametrization tests. The CNN graph consists of a convolutional layer with a 2x2 kernel and 64 features, followed by three fully connected layers with 64 neurons and ReLU activator and a dropout layer followed by a fully connected final layer with softmax activation function, as described below in Figure 12.

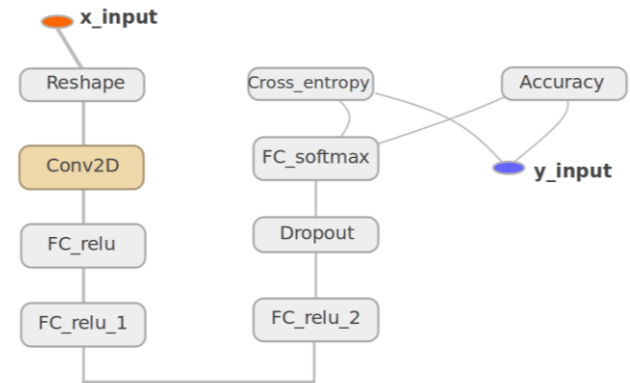


Figure 12 – CNN graph

Results

The results obtained during some tests of prediction are presented below. Figure 13 summarizes the main steps carried out and the respectively prediction image.

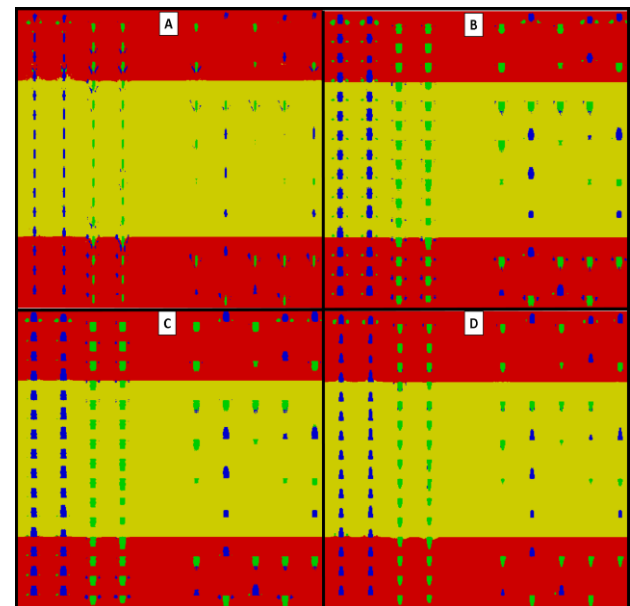


Figure 13 – Image prediction. A) After 10 epochs of training. Note the good accuracy for red and yellow patterns, however lacking horizontal definition and shape recognition; B) with addition of lateral shift as augmentation. Note the gain on horizontal definition; C) after including dropout during CNN training. Note the improvement of triangle shape recognition and D) final prediction after adding noise as augmentation and post-processing

Figure 14 presents a schematic comparing the synthetic pattern classification model and the final prediction image performed by the new multi-attribute convolutional neural network methodology. In this case, the accuracy of total image reaches 93%, an exciting result that shows the potential of this technology in the classification of patterns in images.

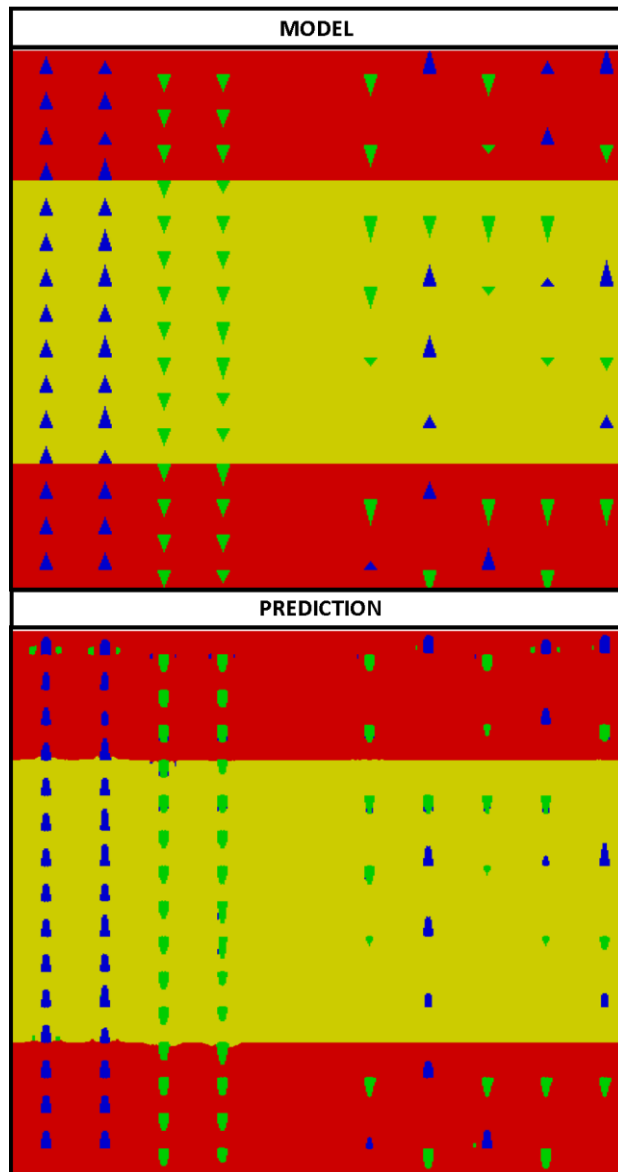


Figure 14 – Comparison showing the synthetic pattern classification model and the final prediction image performed by the new multi-attribute convolutional neural network methodology. In this case, the accuracy of total image reaches 93%

At last, the multi-attribute convolutional neural network methodology was applied at a real seismic data to demonstrate its applicability. The facies model for this approach used several well logs as input information aiming to classify three seismic facies: i) sandstone oil reservoir (in yellow); ii) sandstone water reservoir (in blue) and iii) shale (in green). Figure 15 shows the results obtained at one inline crossing one well.

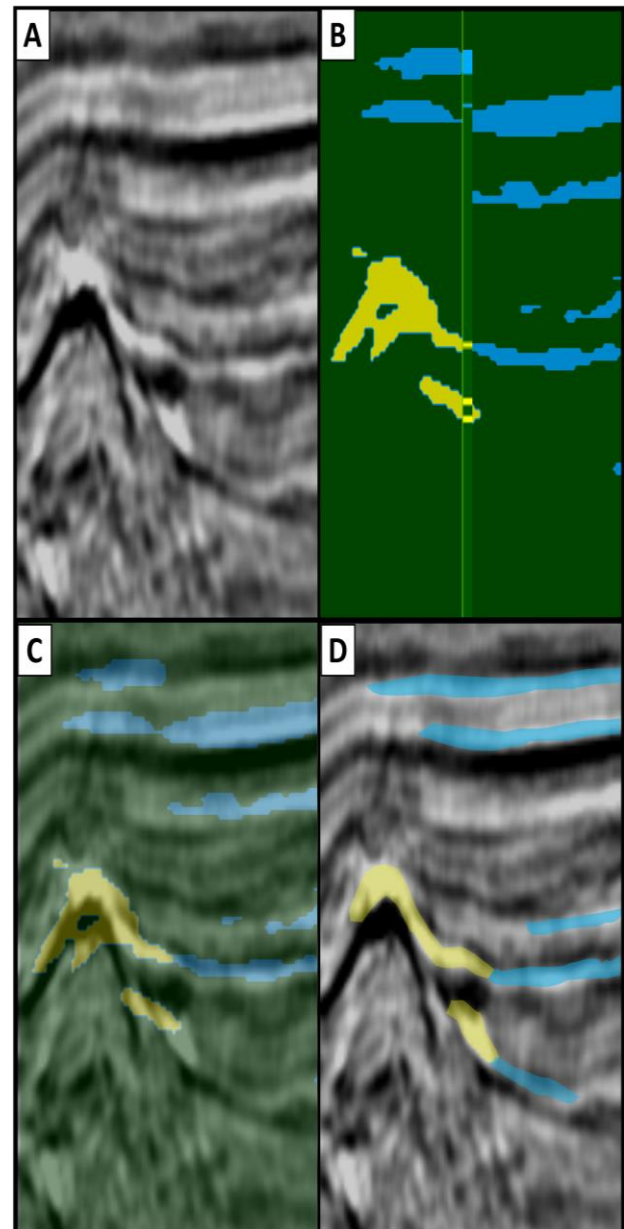


Figure 15 – Real seismic data A) acoustic P impedance; B) well labels showing the facies model fitted with the output image predicted by CNN; C) facies prediction with transparency over acoustic P impedance data and D) geological interpretation using the prediction as valuable information to identify the main bodies. The accuracy obtained between the well facies model and the prediction reached 92.7%

After performing the facies prediction in the seismic volume, a post-processing can be applied to reach the final result. The post-processing, in this specific case, was based on the application of a tool to eliminate “noises”, a simple refining of the result that consists of a mobile window filter that locates the most frequent facies in each window, assigning this value to the center point. Finally, this result guides the interpreter in identifying the main bodies and targets present in the area of interest, as shown in Figure 15D.

Conclusions

In this work, it was successfully developed a reliable methodology based on multi-attribute convolutional neural networks for seismic facies characterization. The synthetic data from a layered grayscale image was crucial to understanding the operation of the CNN as well as analyze the sensitivity of the parameters and hyperparameters of the network. The application of this new methodology for facies classification in a real seismic data was of fundamental importance to evaluate the technical feasibility of this system. The results obtained are very promising, the accuracy between the prediction and the models, both for the synthetic data and for the real seismic data were extremely high, exceeding 90%, representing exciting results that show the potential of this technology.

Acknowledgments

Invision Geophysics thanks Repsol-Sinopec Brasil research group, especially Leonildes Soares, Cassiane Nunes and Támara Garcia for all support to this work and Dr. Luiz Braga, Murilo Lacerda and Uilli Freitas for collaboration and valuable discussions.

This project is a joint research effort of Repsol-Sinopec and Invision Geophysics, developed with the ANP research and development incentive law nº 9.478, 06/08/1997.

References

Da Silva, I. N. *et al.* Artificial Neural Networks: A Practical Course. Springer International Publishing, Switzerland. 307 p., 2017.

Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning: Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts. 775 p., 2016.

LeCun, Y. *et al.* Handwritten Digit Recognition: Applications of Neural Network chips and Automatic Learning. IEEE Communications Magazine. p. 41-46, 1989.

McCulloch, W.; Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, Vol. 5, p. 115-133, 1943.

Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Cornell Aeronautical Laboratory, Vol. 65, No. 6, p. 386-408, 1958.

Srivastava, N. *et al.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, p. 1929-1958, 2014.

Widrow, B.; Hoff, M. Adaptive Switching Circuits. In Proceedings of the Ire Wescon Convention Record, p. 96-104, 1960.

Clarifai. How does Clarifai's visual recognition API work? Available in: <<https://www.clarifai.com/technology/>> Accessed on: November 27, 2017.

Deshpande, A. A Beginner's Guide to Understanding Convolutional Neural Networks Part 2. Available in: <<https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>> Accessed on: November 2, 2017.

Karn, U. An Intuitive Explanation of Convolutional Neural Networks. Available in: <<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>> Accessed on: November 15, 2017.

Stanford course. Convolutional Neural Networks for Visual Recognition. Available in: <<https://cs231n.github.io/convolutional-networks/>> Accessed on: November 6, 2017.