Neural Network model for prediction of ground vibration produced by blasting

Manuel Maria Arez de Vilhena Alves Rafael

Instituto Superior Técnico, Lisboa, Portugal

May 2019

Abstract

Ground vibrations induced by blasting in rock mass excavation can cause damage to the rock mass itself, to structures located in the vicinity and, as well as, disturb humans. Motivated by these reasons, the present work aims to implement a method to predict the peak particle velocities produced, using a consistently organized dataset (including multiple parameters characteristic of several rock blasts). Since the estimation error has been proving to be smaller using artificial neural network methods (ANN) when comparing to results obtained with empirical predictors, ANN were chosen.

The best resulting artificial neural networks obtained, through "K-Fold" cross-validation and "Earlystopping" methods, have $15-12-1$ and $15-15-1$ arquitectures , coefficients of determination $R^2 = 84.0\%$ and $R^2 = 86.6\%$, and Mean Absolute Errors $MAE = 5.59$ and $MAE = 5.02$.

Keywords: Rock blasting, Ground Vibration, Mathematical Modeling, Artificial Neural Networks, Rock Dynamics

1. Introduction

For technical and economic reasons, the use of explosives to rock mass excavation is a widely used approach, and for this reason, as well as by society's growing requirement to comply with environmental impacts, there is the need to create tools to control the impacts associated with excavation techniques, in particular with regard to ground vibrations.

Ground vibrations can be measured on site by engineering seismographs to certify that safety limits are met in accordance with the standards. Nevertheless, it is imperative to have a reliable model that predicts the amplitude of the wave and frequency as a function of distance and other parameters.

Several empirical ground vibration predictors, using linear regression, were developed. These predictors, frequently, only consider the distance from the blast to monitoring point, and the maximum charge per delay to explain this phenomenon. However it is known the ground vibrations produced depend on many other factors, such as: physico-mechanical characteristics of the rock and adopted blast design.

To overcome these problems, a computational model, named Artificial Neural Networks, is proposed as an attempt to identify the relationship, even if non-linear, between several variables of influence in one of the parameters of interest, the vibration amplitude in terms of peak particle velocity.

2. Ground vibration

Immediately after the disturbance caused by the detonation of an explosive charge in a blasthole, the generated energy is transmitted to the rock mass, producing fragmentation. Beyound the fragmentation zone, the energy carried by the seismic waves is attenuated below the limit of occurrence of permanent deformation. In this context, the waves propagate through the medium as elastic waves, known as ground vibrations, oscillating the particles of the medium through which they travel, with reversible deformations (Singh, 2004).

According to Srbulov (2010), ground vibrations are described by their amplitude spectra, maximum amplitude, frequency and duration of the vibration, which vary in space and time, as a consequence of the paths travelled and attenuations suffered.

More systematically, Dowding et al., (2011) states that the frequency coupled with the propagation velocity of the waves determine the response of the rock mass and embedded structures, in which the cracks in the structures result from the induced extensions, where particle velocity is used as an index of strain. Ground vibration is generally described as a time-varying displacement, velocity or acceleration of a specific particle. Since they are vector quantities, they must be described by three mutually perpendicular components in the form of vibration velocities measured in mm/s (Dowding et al., 2011).

Typically, by considering values recorded in a given measurement period, the maximum value of the vibration velocity amplitude of the three directional components is used (v_l, v_v e v_t, respectively corresponding to the longitudinal, vertical and transverse velocities), in order to determine the peak particle velocity (PPV); presented in equation (1).

$$PPV = max(PPV_t, PPV_l, PPV_v) \tag{1}$$

Alternatively to the PPV, for example, the Portuguese standard 2074 (2015) sets the limit for the measured vibration amplitudes in the form of the peak vector sum (PVS); eqation (2).

$$PVS = max(\sqrt{v_t^2(t) + v_l^2(t) + v_v^2(t)}) \tag{2}$$

3. Literature Review

In order to properly deal with the ground vibration problematic, i.e., to comply with the applicable standards, it is convenient to have means of predicting it for a given blasting operation or, ideally, to design a reliable propagation law. The approaches commonly used for the prediction of ground vibrations are the use of an empirical model or, more recently but less used, the implementation of an artificial neural network.

3.1. Empirical models

In 1959, a model was suggested by Duvall & Petkof shown in equation (3), which express the vibration amplitude (v) as a function of the maximum charge detonated per delay (Q), distance between the detonation and the monitoring point (D) and constants (k and b), that simulate the parameters of influence of the particular excavation, such as the implemented blast design, the local geology, etc.

$$v = k \times (\frac{D}{Q^{1/2}})^{-b} \tag{3}$$

Subsquently, other authors proposed alternative models. Some of them, assign different weights to the relation between charge per delay and distance, others use the introduction of more constants in their equations, others do not consider the maximum charge per delay (since, in those blasts, the charge per delay used is approximately equal).

Recent studies (Kumar et al., 2016) summarize empirical models presented by several authors; those can be seen in the table 1.

In addition to the models shown in table 1, other empirical models have been developed, with a growing tendency to contemplate other parameters of influence on ground vibrations, such as geomechanical variables. However, both models tend to be not robust for implementation in zones with properties not identical to those considered in the individual development of each model; besides, they are limited on the number of variables they use.

Table 1: Ground vibration empirical models proposed by different authors (adapted from Kumar et al., 2016)

Authors	**Empirical models**
Duvall and Petkof (1959)	$v = k(D/Q^{1/2})^{-b}$
Langefors and Kihlstrom (1963)	$v = k(Q/D^{2/3})^{b/2}$
Ambraseys and Hedron (1968)	$v = k(D/Q^{1/3})^{-b}$
IS 6922 (1973)	$v = k(Q^{2/3}/D)^{b}$
Ghosh and Daemen (1983)	$v = k(D/Q^{1/2})^{-b}e^{-\alpha D}$
Ghosh and Daemen (1983)	$v = k(D/Q^{1/3})^{-b}e^{-\alpha D}$
Pal Roy (1991)	$v = n + k(D/Q^{1/2})^{-1}$

3.2. Artificial Neural Network models

The limitations mentioned above with respect to these models justify the search for new models through the adoption of new practices, such as Artificial Intelligence, to which the Artificial Neural Networks, used in the present work, belong.

These computational mathematical modeling techniques, inspired by the neuronal networks of animals, have been widely used due to the success verified in the resolution of real problems in several areas of knowledge (Basheer & Hajmeer, 2000). According to Tiile (2016) its application in mining engineering is also growing consequently many researchers have applied the ANN system to predict blast-induced rock fragmentation, airblast and ground vibration (Khandelwal e Singh, 2005; Monjezi et al., 2010; Bahrami et al., 2011; Enayatollahi et al., 2014; hajihassani et al., 2014).

Concerning the implemantation of the ANN, Simpson (1990) states the prior definition required of three intrinsic mathematical components: the network architecture; the activation functions; and the learning algorithm to apply to the network and its training dataset.

3.2.1. Artificial Neural Network arquitecture

The architecture of a neural network imposes in its conception the existence of at least two layers, namely, the input and output layer. The input layer will have a number of neurons equal to the number of explanatory variables to enter in the model. The output will have a number equal to the number of dependent variables to be predicted. In order for a network to be able to converge on a more complex model, offering the possibility of weighted nonlinear combinations of values resulting from functions of the weighted linear combination of the input data, it will be necessary to recall to another type of layer, the hidden layer .

The figure 1 (note[1]) presents a neural network with a number of layers L equal to three (input - layer 1; hidden - layer 2; and output - layer 3), with 3, 4 and 2 neurons (accordingly). Possible representations of the architecture of this network are "$3-4-2$" ou $S_1 = 3$; $S_2 = 4$; $S_3 = 2$, where S_j is the number of neurons in layer number j.

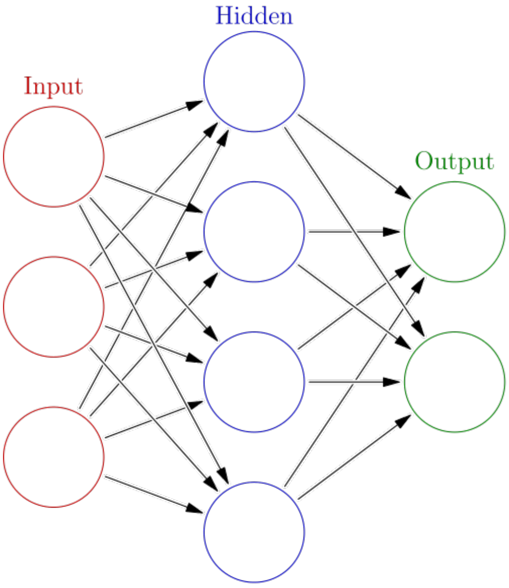


Figure 1: Example of an artificial neural network with a $3-4-2$ arquitecture

The computational mechanism of a neural network is based on producing weighted linear combinations between neurons of consecutive layers. The matrices θ^j of the weights of each connection can be ajusted by a learning algorithm acting on a random initialization of θ^j (Ng, 2012).

The expression (4) presents the computation of the neuron $a_i^{(j+1)}$ through the activation function $g(x)$, weights $\theta^{(j)}$ and the neuron $a_i^{(j)}$.

The neuron value $a_i^{(j+1)}$, i.e., the neuron number i of the layer $j+1$ is the image of the activation function $g(z_i^{j+1})$, which computes the dot product (linear combination z_i^{j+1}) of the row i with the weights matrix $\theta^{(j)}$ with the column vector representative of the neuron $a^{(j)}$.

$$a_i^{(j+1)} = g(z_i^{(j+1)}) = g(\sum_{k=1}^{S_j} \theta_{i,k}^{(j)} a_k^{(j)}) \qquad (4)$$

3.2.2. Activation functions

Activation functions are mathematical functions $f(x)$ applied to the linear combination of the neuron values $a^{(j)}$ and the weights $\theta^{(j)}$ (both of the layer j). Their results represent the values of the neurons of the next layer, as shown in expression (4).

In general, the use of activation functions increases the efficiency of the neural network learning algorithm, by converting, usually, the results of the linear combination of the neurons and weights into values of a non-linear function, monotonous and limited or semi-limited, causing the algorithm to establish non-linear relations between the variables (Ng, 2012).

As an example it is presented the logistic activation function in the equation (5).

$$g(z) = \frac{1}{1+e^{-z}} \qquad (5)$$

3.2.3. Learning algorithm

In the development of the present study, the learning algorithm used was Backpropagation. This algorithm uses a cost function $J(\theta, x, y)$ to quantify the average error between the predictions made for a set of the data, whose value is previously known, and applies an approach of the gradient descent for its minimization. For this purpose MSE cost function was used (equation (6)).

$$J(\theta, x, y) = MSE = \frac{1}{m} \sum_{i=1}^{m} (y(\theta, x_i) - y_i)^2 \qquad (6)$$

Using a random initialization of the weights $\theta^{(j)}$, the algorithm starts each iteration of the training cycle by "feeding" the network with a sample x_i and performs the respective prediction of the result.

After this prediction attempt, $y(\theta, x_i) = a^{(L)}$, the quality of the prediction is evaluated by computing the error vector $\delta^{(L)}$ between the prediction $a^{(L)}$ and the known result y_i. The error obtained from $\delta^{(L)}$ is backpropagated, so that the individual influence of each neuron on the deviation between the prediction and the assumed value is quantified. This is traduced by the vector $\delta^{(j)}$, calculated for each precedent layer of neurons (using the first derivative of the activation function used $g'(z)$).

Each iteration of the training cycle generates $L-1$ vectors $\delta^{(j)}$, whose information "suggests" the deviations to apply to the weights $\theta^{(j)}$, according to the specific values of the neurons $a^{(j)}$ of a given iteration.

In order to update the matrices $\theta^{(j)}$ with weights that minimize the cost function $J(\theta, x, y)$, it is necessary to reconvert the information of the new suggested weights, in a matrix format, in $L-1$ gradient matrices $\Delta^{(j)}$ (updated at each iteration).

After one epoch, i.e., at the end of the training cycle, after running m samples x_i, the weights $\theta^{(j)}$ are updated with an associated MSE. To improve

[1] Credit to Glosser.ca↗, under creative commons license "Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)"↗

the performance of the model more epochs can be done.

For better framing of the backpropagation algorithm its pseudocode is presented (adapted from Ng, (2012)):

```
$x$ = $training\ set\ inputs$
$y$ = $training\ set\ expected\ outputs$
$\theta$ = $random\ initialization$
$While$ $(training\ criteria)$
    $a^{(1,\ldots,L)} = y(\theta, x)$
    $\delta^{(L)} = a^{(L)} - y$
    $For \quad j = (L-1), \ldots, \ 2$
        $\delta^{(j)} = (\theta^{(j)}\delta^{(j+1)}) \odot (g'(\theta^{(j)}a^{(j)}))$
    $For \quad j = (L-1), \ldots, \ 1$
        $\Delta^{(j)} := \Delta^{(j)} + (\delta^{(j+1)})(a^{(j)})^T$
        $\theta^{(j)} := \theta^{(j)} + \frac{1}{m}\Delta^{(j)}$
$MSE$ = $J(\theta, x, y)$
$Return$ $\theta,\ MSE$
```

4. Procedure

The procedure adopted consisted in the creation of a ground vibration dataset to test models, the implementation of the empirical models in the dataset created and the development of artificial neural networks. This was performed with the software Anaconda[2], using the Python language [3] 3.7 with the help of the following libraries [4]:

- Pandas - to handle the dataset;
- Matplotlib - to produce plots;
- Scikit-learn - to use functions for data analysis;
- Keras, on top of TensorFlow - to produce artificial neural networks

4.1. Dataset

For the elaboration of the dataset were taken into account 1114 ground vibration monitoring records, the respective blast designs, the distance from the monitoring point to the blasting face and the geology of 21 quarries. The following variables were collected: number of blastholes ($blastholes$) blasthole diameter (ϕ), hole depth (F), face height (H), stemming (T), burden (A), spacing (S), explosive per hole (Q/F), maximum charge per delay (Q), scaled distance (Ds), distance between face and monitoring point (D) and peak particle velocity (PPV).

All this information comes from the technical report done by Nicholls et al. (1971) at the service of USBM (United States Bureau Mines).

In order to access the behavior of each variable of the dataset, the respective descriptive statistics were calculated and summarized in the table 2.

Table 2: Descriptive statistics

$Variables$	Min	$P_{25\%}$	$Mean$	$P_{50\%}$	$P_{75\%}$	Max	σ^2	σ
$Blastholes$	3.0	15.0	47.4	31.0	51.0	490.0	4550.0	67.5
ϕ	63.5	76.2	116.6	88.9	152.4	228.6	2068.7	45.5
F	3.1	6.1	10.5	9.5	13.7	29.1	28.0	5.3
H	2.7	5.5	9.7	9.1	13.1	28.5	26.9	5.2
T	0.6	1.7	3.0	2.7	4.6	7.8	2.8	1.7
A	1.1	2.1	3.0	2.6	3.6	7.0	1.8	1.4
S	1.2	2.4	3.4	3.1	4.6	8.8	2.0	1.4
Q/F	4.5	18.1	94.1	45.6	90.7	734.8	18000.4	134.2
Q	11.3	160.9	391.2	292.6	539.8	2095.6	117596.9	342.9
Ds	1.8	7.4	17.9	12.9	22.0	155.7	288.1	17.0
D	43.0	128.2	259.3	211.7	336.8	1109.5	32739.7	180.9
PPV	0.2	4.2	20.4	12.0	27.6	129.5	517.6	22.8

4.2. Implementation of empirical ground vibration models

From the empirical models presented in the table 1, the Duvall and Petkof (1959) and Langefors and Kihlstrom (1963)'s models were selected.

4.2.1. Model of Duvall and Petkof (1959)

The model of Duvall and Petkof applied to the dataset obtained the adjustment given by the equation: $PPV = \upsilon = 362 \times (\frac{D}{Q^{1/2}})^{-1.38}$.

The visualization of the observed versus the values predicted by this model can be seen in the diagram of the figure 2.

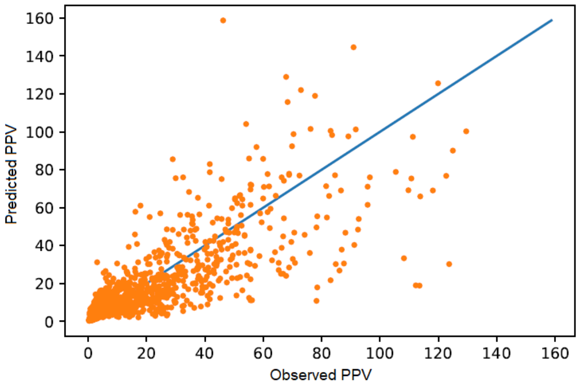


Figure 2: Observed PPV vs predicted PPV for the Duvall and Petkof (1959) model

Through the analysis of the above diagram it is verified that the model sometimes underestimates, other times overestimates the amplitudes of vibration. Thiss lack of robustness can be explained by

[2] Free distribution

[3] Free distribution and open source.

[4] All free distribution.

the variability associated with the data coming from different quarries.

Comparing with the observed data, the estimation of PPV obtained the coefficient of determination 57.7% and mean absolute error 8.74.

4.2.2. Model of Langefors and Kihlstrom (1963)

The model of Langefors and Kihlstrom applied to the dataset obtained the adjustment given by the equation: $PPV = \upsilon = 2.11 \times (\frac{Q}{D^{2/3}})^{1.61}$.

The visualization of the observed versus the values predicted by this model can be seen in the diagram of the figure 3.

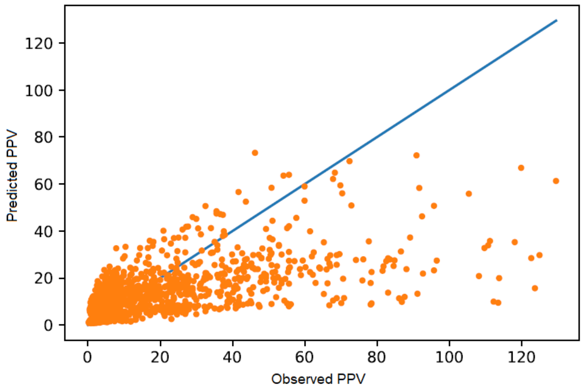


Figure 3: Observed PPV vs predicted PPV for Langefors and Kihlstrom (1963) model

Through the analysis of the scatter plot above, the resulting underestimation is evident. It is also worth mentioning that its performance against the model of Duvall and Petkof (1959), measured by the mean absolute error and coefficient of determination, is by far inferior, where the $R^2 = 27.9\%$ and $MAE = 11.9$.

4.3. Artificial neural network design for ground viration estimation

Among all the explanatory variables, the numerical variables selected to be introduced as inputs in the neural networks were:

- Blasthole diameter (ϕ);
- Hole depth (F);
- Maximum charge per delay (Q); and
- Distance between face and monitoring point (D).

Prior to the training and the introduction of the input variables in the neural network, the samples (x_i) were scaled according to the formula $\frac{x-i-\mu_x}{\sigma_x}$. This formula turns values of different variables into the same order of magnitude, avoiding that weights of variables with high values overlap the weights of variables with low values, making it difficult or even preventing the learning of the network (Basheer & Hajmeer, 2000).

As for the type of geology, because it is a categorical variable, its information was extracted using indicator variables for its codification. The table 3 (adapted from Paneiro et al., (2019)) represents this codification, for a categorical variable with n categories, where the first category is considered as the reference category and for that all values of the indicator variables are 0, and for each of the other categories, the corresponding indicator variable are equal to 1. Knowing that there are 12 distinct types of formations in the dataset, 11 indicator variables of the types of geology were created.

Table 3: General representation of categories (C1, ..., Cn) and corresponding codification of a qualitative variable

Category	Indicator variables			
	K_1	K_2	...	K_n
C_0	0	0	...	0
C_1	1	0	...	0
C_2	0	1	...	0
...	...	...	...	...
C_n	0	0	...	1

Consequently, the number of neurons in the inputs layer is set to 15, which correspond to the variables: ϕ, D, Q, F plus 11 indicator variables. The remaining parameters needed to create the neural networks, including the activations and the optimizers functions, the number of hidden layers and its respective neurons, and the number of epochs, were defined as follows:

- Number of hidden layers - one hidden layer;
- Neurons in the hidden layer - combinations of neurons from 1 to the maximum of 15 were chosen (number equal to the number of input variables);
- Activation function - the three most common were tested (according to Ng, (2012)): ReLU, Tanh and Sigmoid;
- Optimizer - were tested four functions available in the Keras library[5], namely: Adam (*Adaptative moment estimation*), Adagrad (*Adaptative gradient algorithm*), SGD (*Stochastic gradient descent*) and RMSprop (*Root mean square propagation*);
- Number of epochs - considering that the optimization functions converge to local minimums

[5] https://keras.io/optimizers/ ↗

of the cost function with different speeds, the model was trained with six different epochs: 50, 100, 200, 400, 500 and 700.

These parameters were organized together in order to develop all possible neural networks resulting from its combinations. The scheme represented by the figure 4 intends to demonstrate this.

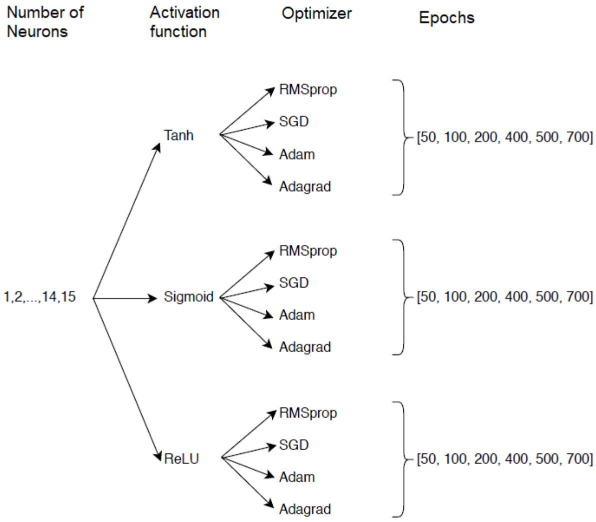
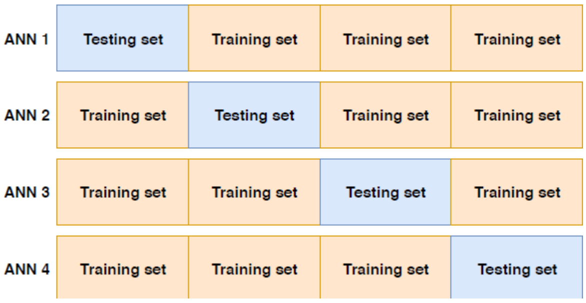


Figure 4: Possible combinations of arquitectures and hyperparameters used in the analysis

At the same time, the "K-Fold" cross-validation (figure 5) was employed in the 1114 observations contained in the dataset with CV=4 to divide them into four sets with approximate equal size.



ANN 1	Testing set	Training set	Training set	Training set
ANN 2	Training set	Testing set	Training set	Training set
ANN 3	Training set	Training set	Testing set	Training set
ANN 4	Training set	Training set	Training set	Testing set

Figure 5: "K-Fold" cross-validation with CV=4

In this approach, three sets (75% of observations) are used to train the neural network and the remaining set (25% of observations) is saved to test the performance of the model produced. This process is executed four times since there are four different combinations to divide the four sets into three sets to develop the model and another to do the corresponding test. The advantage of this approach relies on the possibility to evaluate the general predictive capacity of the ANN.

In the implementation of the proccess, described above, 45 neural networks were produced, resulting from the combination of 1 to 15 neurons in the hidden layer for 3 activation functions. For those ANN, 4 optimizers and 6 epochs were used for training while training the model in combination with "K-Fold" cross-validation (CV=4).

For each hidden layer dimension (DCE), was recorded the set of parameters that promoted the lowest mean of the mean absolute error (MAE) between the observed and predicted PPV given by the four "K-Fold models.

In order to measure the generalization capacity of the models' predictions, depending on the data used to test, the standard deviation associated and the MAE_{min} were calculated. To check the performance of the models the coefficients of determination was also calculated. These results are summarized in the table 4.

Table 4: Best models for each hidden layer dimension size with "K-Fold" cross-validation

DCE	Optimizer	Activation	Epoch	MAE_{mean}	MAE_{min}	σ_{MAE}	R^2_{mean}	R^2_{max}	σ_{R2}
1	RMSprop	Tanh	700	9.07	8.32	0.55	50.4%	52.3%	2.6%
2	SGD	Sigmoid	400	8.25	7.11	1.42	70.2%	77.2%	10.0%
3	SGD	Sigmoid	400	7.70	7.12	0.68	71.0%	76.8%	7.7%
4	SGD	Sigmoid	700	6.81	6.62	0.18	77.4%	78.4%	1.1%
5	SGD	Sigmoid	500	6.68	6.18	0.42	77.8%	79.4%	1.6%
6	RMSprop	Tanh	700	6.67	6.32	0.22	78.5%	81.8%	1.9%
7	RMSprop	Tanh	400	7.20	6.60	0.40	73.9%	76.2%	1.5%
8	SGD	Sigmoid	700	6.23	5.79	0.35	81.3%	83.4%	1.7%
9	SGD	Sigmoid	700	6.15	5.71	0.43	80.8%	82.8%	1.8%
10	SGD	Sigmoid	500	6.16	5.60	0.42	80.9%	83.8%	2.3%
11	SGD	Sigmoid	400	6.10	5.42	0.45	81.4%	84.7%	2.6%
12	SGD	Sigmoid	400	6.09	5.59	0.29	81.6%	84.0%	1.6%
13	SGD	Sigmoid	500	6.11	5.46	0.43	81.0%	84.5%	2.2%
14	SGD	Sigmoid	500	6.27	5.43	0.91	80.0%	85.2%	6.2%
15	SGD	Sigmoid	700	6.03	5.41	0.56	81.3%	83.5%	2.4%

From the analysis of the best models obtained, it is notable that, for the optimizers considered, neither Adam nor Adagrad were selected. Considering the activation functions, the same happened for the ReLU. In general, the parameters that provided better performance neural networks were the SGD optimizer and Sigmoid activation function.

Excluding the model with one neuron in the hidden layer, all the others performed better than the best empirical model obtained (Duvall and Petkof, 1959). Being that, the best model obtained was the one with $DCE = 12$. In fact it has the highest coefficient of determination mean, R^2_{mean}, 81.6%; the second lowest mean of the mean absolute error, MAE_{mean}, 6.09; and a low standard deviation associated with the mean absolute error, σ_{MAE}, 0.29, which can be indicator of a greater capacity of generalization.

The figure 6 shows the training of the referred model with $DCE = 12$, where the evolution of the mean square error MSE decreases as a function of epochs.

It should be noted that, counterintuitively, in most epochs, there is a better performance of the model when applied to the test set than when applied to the training set. One possible justification is that the size of the training data set is three times

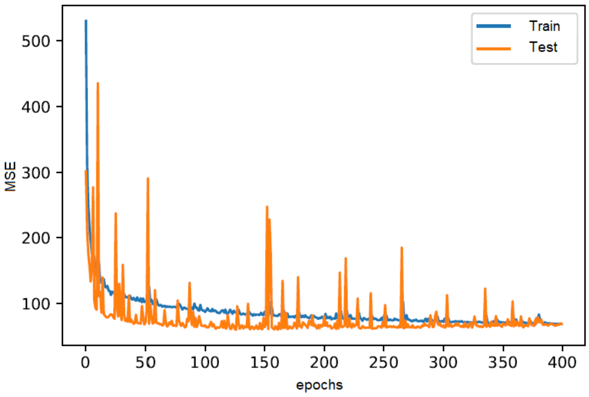


Figure 6: MSE as a funtion of the epochs for the best model obtained ($DCE = 12$)

larger than the test set, whereby there is a greater probability that it will contain a greater amount of anomalous values, whose presence will significantly increase its MSE. Sometimes, as might be expected, the cost function will tend to local minimums with much greater variance in the predicted values of the test set, that happens because the weights adjust to the training data ignoring the reality of certain anomalous values mostly found in the test set.

Through the visualization of the predicted versus the observed values (figure 7), concerning the best model with $DCE = 12$, a good approximation of the predictions done is observable.

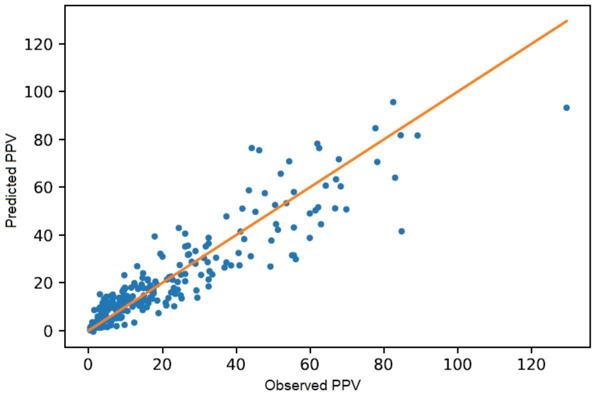


Figure 7: Observed PPV vs predicted PPV ($DCE = 12$)

Since the epochs can assume a wide range of values (in which only 6 were considered) and which, as visible in the figure 6, its value has a great influence on the performance of the neural network, it was chosen to determine the epochs using a method called ”Earlystopping”. As for the remaining parameters needed (to define the ANN), those were established equal to those presented in the first three columns of the table 4, once they correspond to previously dimensioned parameters.

The Earlystopping method consists on interrupting the training cycle (of the ANN) after a certain epoch when the MSE of the test set does not improve significantly in relation to recently obtained values. The implementation of this method tries to prevent the overfitting of the model to the data by the anticipated interruption of the training. For the present case, 0.01 was defined as the minimum value of variation of the test MSE over 15 epochs.

In this approach, the ”K-Fold” cross-validation was not implemented, because its simultaneous use with ”Earlystopping” is not convenient. Since the evolution of MSE is substantially different due to the set of data used to test the ANN, causing the ”Earlystopping” to interrupt the training at different epochs and due to that their results will not be comparable. Therefore, the set of observations was randomly divided into 75% of them to train the network and 25% for testing.

The results obtained by the use of ”Earlystopping” are presented in the table 5.

Table 5: Best models for each hidden layer dimension size with ”Earlystopping”

DCE	Optimizer	Activation	Epoch	MAE	R^2
1	RMSprop	Tanh	700	8.77	54.9%
2	SGD	Sigmoid	58	6.40	82%
3	SGD	Sigmoid	50	5.87	84%
4	SGD	Sigmoid	46	5.85	83.9%
5	SGD	Sigmoid	43	5.83	83.8%
6	RMSprop	Tanh	700	5.79	83.6%
7	RMSprop	Tanh	400	6.18	80.7%
8	SGD	Sigmoid	74	5.36	85.4%
9	SGD	Sigmoid	113	5.45	85.2%
10	SGD	Sigmoid	76	5.44	84.9%
11	SGD	Sigmoid	113	5.25	85.9%
12	SGD	Sigmoid	76	5.44	85%
13	SGD	Sigmoid	33	5.68	83.5%
14	SGD	Sigmoid	118	5.21	86.2%
15	SGD	Sigmoid	118	5.02	86.6%

It should be noted that for certain cases ($DCE = 1$, $DCE = 6$ and $DCE = 7$), where the optimizer is the RMSprop with Tanh activation function, the algorithm did not interrupt the training cycle in advance, since its learning did not obtain sufficiently low variance in the evolution of its MSE.

Doing the comparative analysis of the results (MAE and R^2) obtained by this method in addition to the results (MAE_min and R^2max) obtained in the table 5, it is observed that with the exception of the $DCE = 13$ model, all the remaining have the lowest MAE and the greatest R^2, with the best performing model being $DCE = 15$. By comparison with the observed data for the model with $DCE = 15$, the estimation of the PPV obtained the coefficient of determination $R^2 = 86.6\%$ and mean absolute error $MAE = 5.02$.

The training of the model can be observed in the figure 8 and the visualization of the observed versus those estimated in the figure 9.

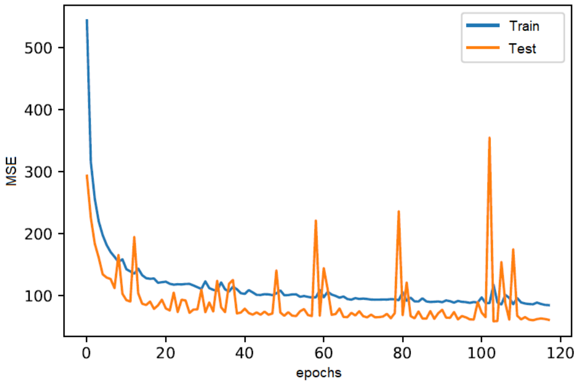


Figure 8: MSE as a funtion of the epochs for the best model obtained ($DCE = 15$)

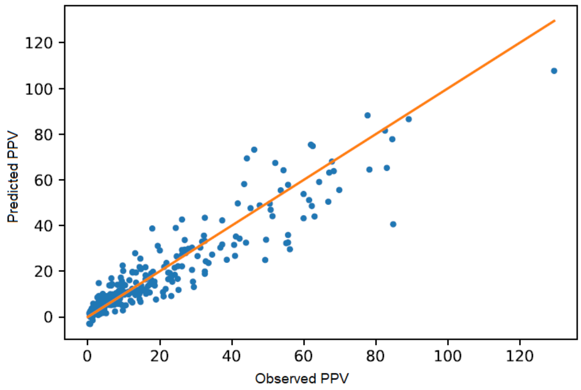


Figure 9: Observed PPV vs predicted PPV ($DCE = 15$ and 118 epochs)

The advantage of the model obtained by ”Earlystopping”, compared to the model obtained by the ”K-Fold” cross-validation, refers to the dimensioning of the epochs as a function of the prediction error of the model. However, the robustness of the model compared to the data used for its testing will tend to be smaller (test set size is four times lesser). As for the best performances verified for the different DCE, these may be associated with the quality of the test data set.

5. Conclusions

In the present work a dataset was developed with 1114 observations of blasts and the respective vibration amplitudes (PPV). The variables with the highest statistical significance for PPV were analyzed, concluding that these were the blasthole diameter, the drilling length, the maximum detonated charge per delay, the distance between the blast and the monitoring point, and the type of geological formation.

Empirical models for ground vibration propagation were fitted, such as Duvall and Petkof (1959), to estimate the relations between maximum charge per delay (Q) and the distance (D) with the PPV recorded in the dataset. The coefficient of determination obtained was $R^2 = 57.7\%$ and the mean absolute error $MAE = 8.74$.

As for the neural networks, the best ones obtained, with hidden layer dimensions (DCE) greater than one neuron, both with ”K-Fold” cross-validation and Earlystopping, performed better than the empirical models tested.

The best neural network obtained, with the ”K-Fold” cross-validation and $DCE = 12$, present an $R^2 = 84.0\%$ e $MAE = 5.59$. Whereas with Earlystopping and $DCE = 15$, present an $R^2 = 86.6\%$ and $MAE = 5.02$.

The observed evolution of the mean square error (MSE), referring to the training and test data (of the ANN) and the efforts made through the implementations of the algorithms used, suggest that there was no overfitting of the ANN. However, this property can not be guaranteed since it has often obtained better MSE results in the test data than in the training data. This may be due to the fact that the observations recorded in the dataset do not show much numerical diversity in the variables used, as they result from the collection of data from quarries with variables of little variability within each quarry. This way, it is possible that the anomalous values may have more expression in the training sets, given their larger dimension in relation to the test sets, usually increasing their MSE.

Although there is a good performance of the estimation capacity of the neural networks obtained, it does not reach those described in the bibliography referred to, since the records in the dataset correspond to a total of 21 different quarries, as opposed to a single excavation work , with lower variability of its variables. Thus, the obtained estimation capacity represents a greater generalized prediction capacity.

References

Basheer, I. A. & Hajmeer, M. (2000), ‘Artificial neural networks: fundamentals, computing, design, and application’, *Elsevier* **43**, 3–31.

Dowding, C., Peter & j (2011), *S. M. E. Mining Engineering Handbook*, 3^{nd} edn, Society for Mining, Metallurgy and Exploration, Inc.

Duvall, W. I. & Petkof, B. (1959), Spherical propagation of explosion-generated strain pulses in rock, Technical report, Bureau of Mines.

Kumar, R. C., Bhargava, D. & Kapilesh (2016), 'Determination of blast-induced ground vibration equations for rocks using mechanical and geological properties', *Journal of Rock Mechanics and Geotechnical Engineering* **8**, 341–349.

Ng, A. (2012), 'Machine learning course', *Coursera [online]*. .
URL: *https://www.coursera.org/learn/machinelearning*

Nicholls, H. R., Johnson, C. F. & Duvall, W. I. (1971), *Blasting vibrations and their effects on structures*, US Government Printers.

Paneiro, G., Durão, F. O., e Silva, M. C. & Bernardo, P. A. (2019), 'Neural network approach based on a bilevel optimization for the prediction of underground blast-induced ground vibration amplitudes', *Neural Computing and Applications* pp. 1–13.

Simpson, P. K. (1990), *Artificial neural systems: foundations, paradigms, applications, and implementations*, Pergamon.

Singh, T. (2004), 'Artificial neural network approach for prediction and control of ground vibrations in mines', *Mining Technology* **113**(4), 251–256.

Tiile, R. N. (2016), Artificial neural network approach to predict blast-induced ground vibration, airblast and rock fragmentation, Master's thesis, Missouri University of Science and Technology.