Application Proposal of Neural Networks to Control Diameter of Rod Journal and Main Journal of Crankshafts in Production Line

L. C. M. F. Diogenes

Abstract – Manufacturing has been the target of the application of artificial intelligence. Neural networks can imitate human thought processes with few human interventions and, therefore, are able to learn and adapt to changes. This article presents a robust neural network model containing two neurons in the input layer, four in the hidden layer and one in the output layer to separate diameters of rod journal and main journal from a crankshaft, within a production line, which do not present diameter values within the specification provided by the customer. For values outside the expected, the crankshafts can be discarded or reworked.

Keywords — neural network, crankshaft, class separation, production line.

I. INTRODUCTION

The application of neural networks on crankshafts has already been reported by some authors. According to [1], artificial neural networks have been used to estimate pressure parameters inside an engine, as in a engine crankshaft speed.

In [2] a work is presented that uses neural networks to determine structural failures on steel bars and crankshafts.

According [3] to the crankshaft it is a piece of caston iron or forged steel that supports a lot of force when the piston pushes the connecting rod down and converts the movement piston in rotating motion.

The Fig. 1 shows the design of a crankshaft model, where the main bearing journals are located on a central axis while that connecting rod bearings journals are outside that line.

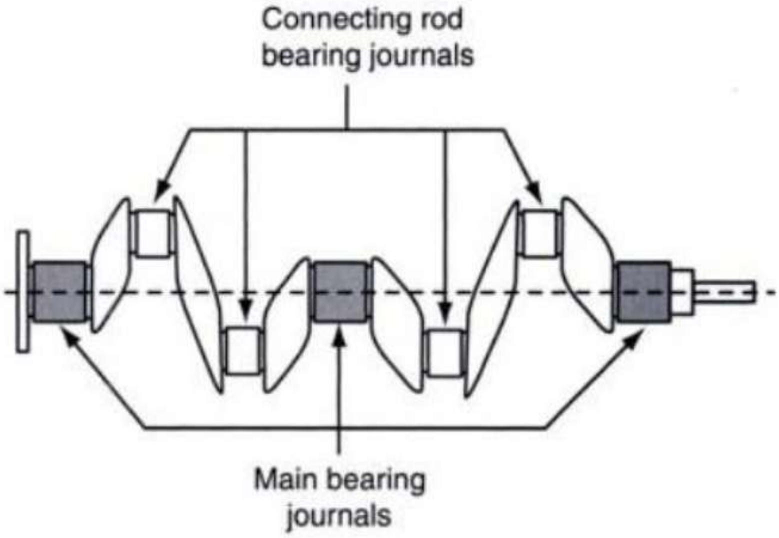

Fig. 1 One crankshaft piece [3]

L. C. M. F. Diogenes, independente reseacher, Brazil (e-mail: lucianafem@yahoo.com.br).

According [4] the crankshafts journals must have their diameters measured with a micrometer to know if they are within specification, as showed in Fig. 2.

Fig. 2 Measuring a bearing with micrometer [4]

Keeping the diameters of the crankshafts within tolerance is essential for fitting the other parts and also working them inside the engine. For this work, the diameters and tolerances of each bearing are indicated in the Fig. 3.

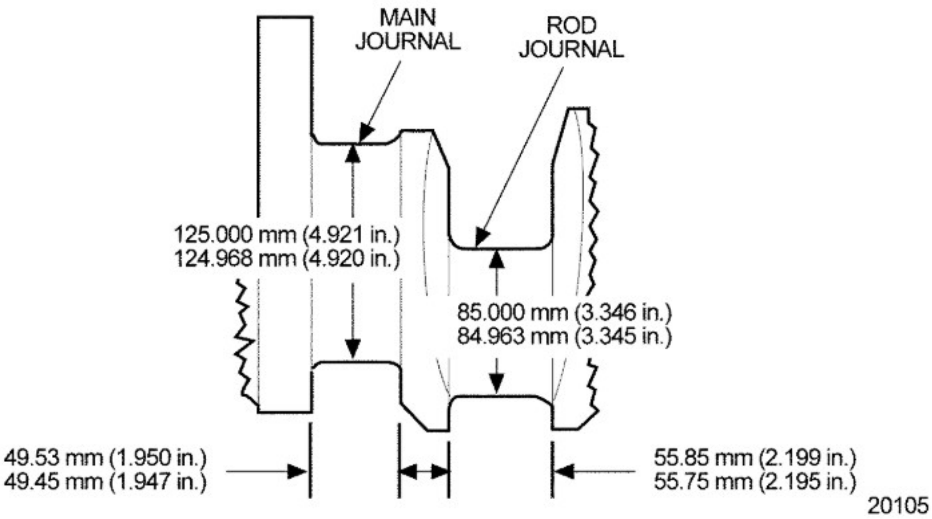

Fig. 3 Diametesr and tolerances considered in this work [5]

II. Artificial Neural Networks

To understand what an artificial neural network is analyze the neurobiology of a neural cell. In the human body there are around 100 billion neurons which communicate through electrical signals that are transmitted very quickly, ie by duration spikes in cell membrane voltage [6]. The Fig. 4 shows a very simple image representing a neuron and its parts.

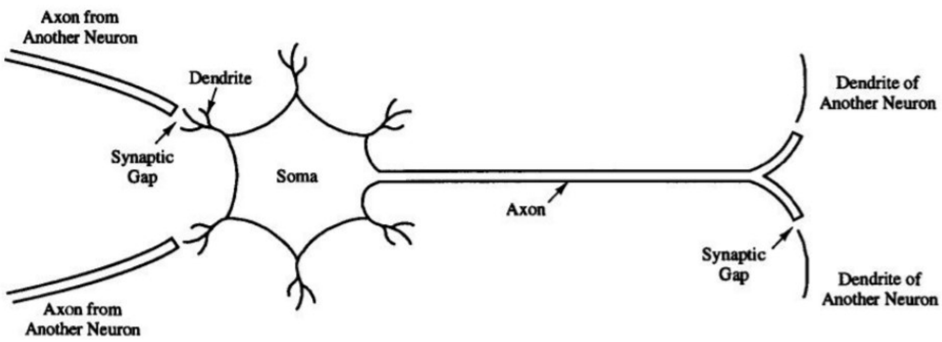

Fig. 4 A neuron and its constituent parts [Fausett,1992]

The branches of neurons are called dentites and they are responsible for conducting signals into the cell body. The single branch is called an axon and transmits a signal from the cell body to its extremities. The axon of one neuron is attached to the dentites of another neuron through synapses. In cases specific, the axon can be attached to another axon or directly to a cell body [8].

Artificial neural networks (ANNs) are software implementations of the neuronal structure of our brains [9]. Synapses in the artificial neural networks are modelled by a single number, called of weight, so that each input x_i is multiplied by a weight w_i before being sent to cell body artificial. The weighted signals are summed to supply a node activation. The neuron's output has value 0 or 1and it determined by whether the weighted sum [6], [10], [11]:

$$\sum w_i x_i \tag{1}$$

is less than or greater than some threshold value.

In the Fig. 5 is showed a simple artificial neuron [6]:

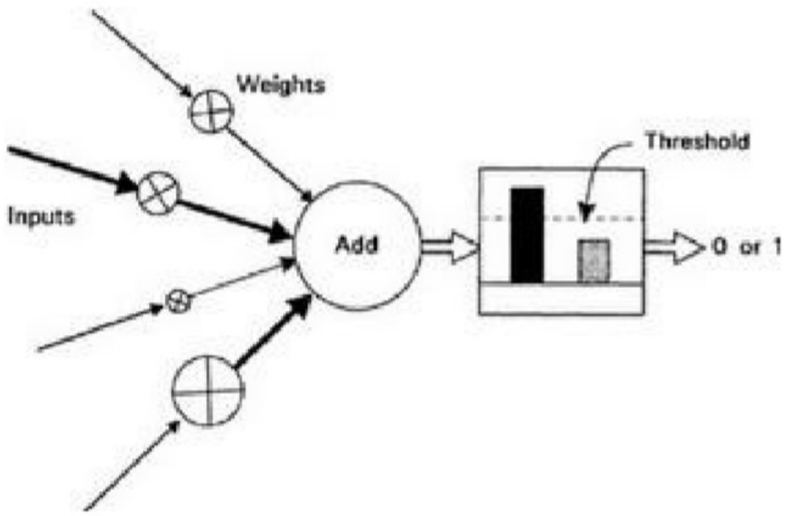

Fig. 5 Artificial neuron [6]

An architecture of at least one intermediate (hidden) layer of neurons, located between an input layer and an output neural layer, is called a multilayer perceptron (MLP). The process of training MLP networks using the backpropagation algorithm is widely used. In this method, the weights and bias are updated in a reverse order, resulting in a gradual decrease in the sum of errors [12].

The neural model , or the MLP model, proposed to separate crankshafts with diameters within the specifications sent by the customer, and which must be presented in the drawing, consists of two neurons in the first layer, four neurons in the hidden layer and two neurons in the hidden layer, as shown in Fig. 6. In inputs x_1 are the measurements of the diameters of the rod jornal and in inputs x_2 the values of the diameters of the main journal. The values of the inputs are sent to the hidden layer and there in that layer, they still receive weight values ranging from w_1 to w_8. After the math is done, the new values are sent to the output layer, where they also receive the values from w_9 to w_{12}.

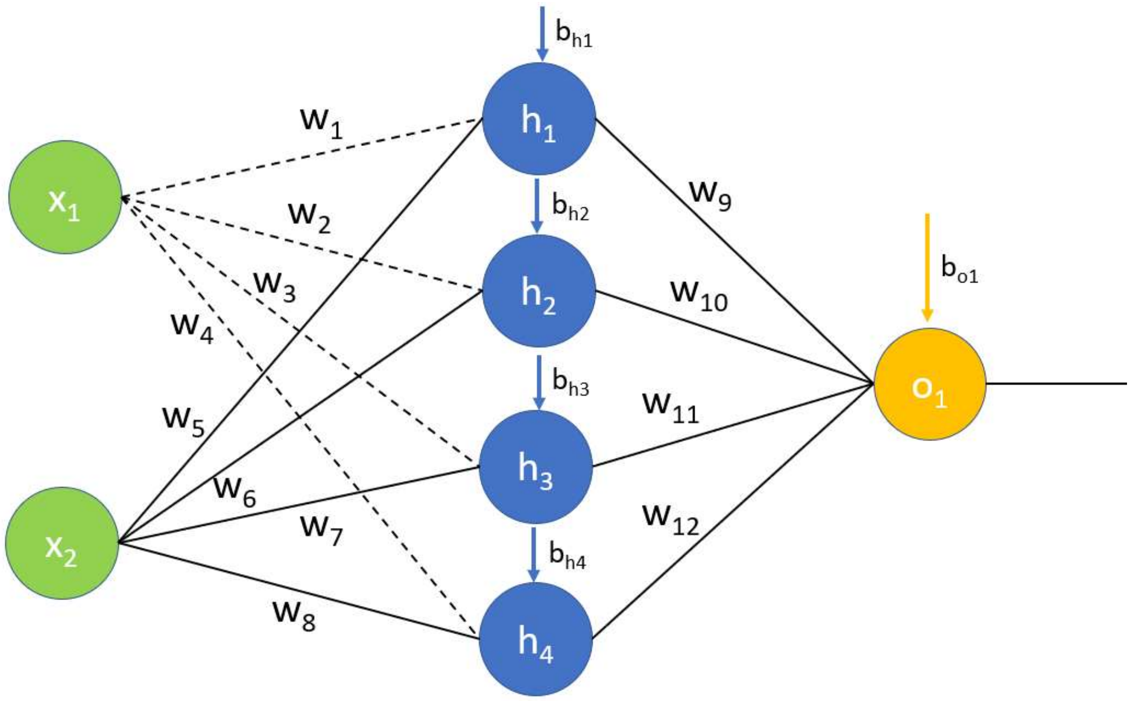

Fig. 6 Neural neural model proposed for this work where h_i is the hidden layer and o_1 the output layer

Inputs can need be scaled in the range (0,1). Then, consider x_{min} and x_{max} as being the minumum and maximum values of each input [13]:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

where: $x = x_1$ or x_2.

For each neuron in hidden layer (h_1 up to h_4), the output neuron is given by [Frauke Günther and Stefan Fritsch, 2010]:

$$h_i(\boldsymbol{x}) = f(b_{0i} + \sum_{i=1}^{n} w_i x_i) = f(b_{0i} + \boldsymbol{w}^T \boldsymbol{x}) \tag{3}$$

where: i= 1, 2,3 or 4, b_0i: bias, weigths **w**: **w**= $(w_1,...,w_n)$ and inputs **x** : **x**= $(x_1,...,x_n)$.

The activation function f is applied to hidden layer h_i or output layer o_1 and the fuction chosen for this work is defined by the formula [14]:

$$f(u) = \frac{1}{1+e^{-u}} \tag{4}$$

where: u= $h_i(\boldsymbol{x})$ or $o_1(\boldsymbol{x})$ and the output this function is 0 or 1.

For the last layer o_1, or output layer, the output neuron is given by [11]:

$$o_1(\boldsymbol{x}) = f\left(w_0 + \sum_{j=1}^{J} w_j . f\left(w_{0j} + \sum_{i=1}^{n} w_{ij} x_i\right)\right) \tag{5}$$

$$o_1(\boldsymbol{x}) = f\left(w_0 + \sum_{j=1}^{J} w_j . f\left(w_{0j} + \boldsymbol{w}_j^T \boldsymbol{x}\right)\right) \tag{6}$$

Consider that output result is f (o1 (x)) = g_{o1}. For a classification problem g_{o1} gives output in the range of 0 to 1. If output is between 0.5 and 1 should indicate that the crankshaft is approved, that is, that its diameters are within the tolerance values. If the value is between 0 and 0.5, it shows that the crankshaft is failing and must be discarded or reworked.

An error function E defines the error between the desired output y and the calculated output g_{o1}:

$$E = \frac{1}{2}\left(y - g_{o_1}\right)^2 \tag{7}$$

The updated values of the bias and weights are determined by the gradient of the error function, that is [6], [14]:

$$b_k^{(t+1)} = b_k^{(t)} - \eta_k^{(t)} . \left(\frac{\partial E^{(t)}}{\partial b_k^{(t)}}\right) \tag{8}$$

$$w_k^{(t+1)} = w_k^{(t)} - \eta_k^{(t)} . \left(\frac{\partial E^{(t)}}{\partial w_k^{(t)}}\right) \tag{9}$$

To minimize the value of the error function, the gradient values are calculated for each weight of the net. Then, for w_9, w_{10}, w_{11} e w_{12}, the equation is written as:

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial g_{o_1}} * \frac{\partial g_{o_1}}{\partial u_{o_1}} * \frac{\partial u_{o_1}}{\partial w_i} \tag{10}$$

where i= 9,10, 11 and 12.

For b_{h1}, b_{h2}, b_{h3} and b_{h4}:

$$\frac{\partial E_{total}}{\partial b_{h_i}} = \frac{\partial E_{total}}{\partial g_{h_i}} * \frac{\partial g_{h_i}}{\partial u_{h_i}} * \frac{\partial u_{h_i}}{\partial b_{h_i}} \tag{11}$$

where i= 1, 2, 3 and 4.

For b_{o1}:

$$\frac{\partial E_{total}}{\partial b_{o_1}} = \frac{\partial E_{total}}{\partial g_{o_1}} * \frac{\partial g_{o_1}}{\partial u_{o_1}} * \frac{\partial u_{o_1}}{\partial b_{o_1}} \tag{12}$$

For w_1 and w_2:

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial g_{h_1}} * \frac{\partial g_{h_1}}{\partial u_{h_1}} * \frac{\partial u_{h_1}}{\partial w_i} \tag{13}$$

where i= 1 and 2.
For w_3 and w_4:

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial g_{h_2}} * \frac{\partial g_{h_2}}{\partial u_{h_2}} * \frac{\partial u_{h_2}}{\partial w_i} \tag{14}$$

where i= 3 and 4.
For w_5 and w_6:

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial g_{h_3}} * \frac{\partial g_{h_3}}{\partial u_{h_3}} * \frac{\partial u_{h_3}}{\partial w_i} \tag{15}$$

where i= 5 and 6.
For w_7 and w_8:

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial g_{h_4}} * \frac{\partial g_{h_4}}{\partial u_{h_4}} * \frac{\partial u_{h_4}}{\partial w_i} \tag{16}$$

where i= 7 and 8.

Each bias and weight is updated iteratively using the value of the gradients. Then, the new bias and weights are given by [14]:

$$b_{h_1}(\mathrm{t}+1) = b_{h_1}(\mathrm{t}) - \eta(\delta_{o1} w_9)\, g_{h_1}(1 - g_{h_1}) \tag{17}$$

$$b_{h_2}(\mathrm{t}+1) = b_{h_2}(\mathrm{t}) - \eta(\delta_{o_1} w_{10})\, g_{h_2}(1 - g_{h_2}) \tag{18}$$

$$b_{h_3}(\mathrm{t}+1) = b_{h_3}(\mathrm{t}) - \eta(\delta_{o_1} w_{11})\, g_{h_3}(1 - g_{h_3}) \tag{19}$$

$$b_{h_4}(\mathrm{t}+1) = b_{h_4}(\mathrm{t}) - \eta(\delta_{o_1} w_{12})\, g_{h_4}(1 - g_{h_4}) \tag{20}$$

$$b_{0_1}(\mathrm{t}+1) = b_{0_1}(\mathrm{t}) - \eta\delta_{01} \tag{21}$$

$$w_1(t+1) = w_1(t) - \eta(\delta_{o1} w_9) g_{h_1}(1 - g_{h_1})\, x_1 \tag{22}$$

$$w_2(t+1) = w_2(t) - \eta(\delta_{o1} w_9) g_{h_1}(1 - g_{h_1})\, x_2 \tag{23}$$

$$w_3(t+1) = w_3(t) - \eta(\delta_{o1} w_{10}) g_{h_2}(1 - g_{h_2})\, x_1 \tag{24}$$

$$w_4(t+1) = w_4(t) - \eta(\delta_{o1} w_{10}) g_{h_2}(1 - g_{h_2})\, x_2 \tag{25}$$

$$w_5(t+1) = w_5(t) - \eta(\delta_{o1} w_{11}) g_{h_3}(1 - g_{h_3})\, x_1 \tag{26}$$

$$w_6(t+1) = w_6(t) - \eta(\delta_{o1} w_{11}) g_{h_3}(1 - g_{h_3})\, x_2 \tag{27}$$

$$w_7(t+1) = w_7(t) - \eta(\delta_{o1} w_{12}) g_{h_4}(1 - g_{h_4})\, x_1 \tag{28}$$

$$w_8(t+1) = w_8(t) - \eta(\delta_{o1} w_{12}) g_{h_4}(1 - g_{h_4})\, x_2 \tag{29}$$

$$w_9(t+1) = w_9(t) - \eta\delta_{o1} g_{h_1} \tag{30}$$

$$w_{10}(t+1) = w_{10}(t) - \eta\delta_{o1} g_{h_2} \tag{31}$$

$$w_{11}(t+1) = w_{11}(t) - \eta\delta_{o1}g_{h_3} \quad (32)$$

$$w_{12}(t+1) = w_{12}(t) - \eta\delta_{o1}g_{h_4} \quad (33)$$

where:

$$\delta_{o1} = [-(y - g_{o_1})][g_{o_1}(1 - g_{o_1})] \quad (34)$$

III. Results

In order to check from which epoch the program starts to present the desired result, that is, approve the inputs if the output values are greater than 0.5 and fail, otherwise, the program was run countless times until it found this threshold value. The table I shows the results for the threshold value found, which was 14,578. Then, the value of the previous epoch 14,577, shows one of the inputs (the third) with a different result than the desired one.

Table I: Result of neural network training for 14,577 and 14,578 epochs.

Input (mm)	Desirable Result	Output 14,577 epochs	Result	Output 14,578 epochs	Result
[84.963, 124.968]	A	0.97	A	0.97	A
[84.963, 125]	A	0.89	A	0.89	A
[85, 124.968]	A	0.49	R	0.5	A
[85, 125]	A	0.91	A	0.91	A
[84.951, 124.96]	R	0	R	0	R
[84.951, 125.013]	R	0	R	0	R
[85.013, 124.96]	R	0	R	0	R
[85.013, 125.013]	R	0	R	0	R
[84.981, 124.96]	R	0.32	R	0.32	R
[84.981, 125.013]	R	0.06	R	0.06	R
[84.991, 124.96]	R	0.25	R	0.25	R
[84.991, 125.013]	R	0.01	R	0.01	R
[84.951, 124.98]	R	0.02	R	0.01	R
[85.013, 124.98]	R	0.25	R	0.25	R
[84.956, 124.968]	R	0.12	R	0.12	R
[85.013, 124.968]	R	0	R	0	R

To increase the precision in the selection of diameters, the program was run with 20,000 epochs. The new bias and weight values, calculated by the equations 17 up to 33 are shown in the table II.

Table II: Result of neural network training for 20,000 epochs.

c	w value	b	b value	Input (mm)	Desirable Result	Output	Result
w_1	-22.75	b_{h1}	2.16	[84.963, 124.968]	A	0.96	A
w_2	0.52	b_{h2}	-15.77	[84.963, 125]	A	0.99	A
w_3	17.05	b_{h3}	-1.37	[85, 124.968]	A	0.97	A
w_4	0.54	b_{h4}	-12.0	[85, 125]	A	0.98	A
w_5	0.06	b_{o1}	-6.45	[84.951, 124.96]	R	0	R
w_6	18.21			[84.951, 125.013]	R	0	R
w_7	0.7			[85.013, 124.96]	R	0	R
w_8	12.32			[85.013, 125.013]	R	0	R
w_9	-14.08			[84.981, 124.96]	R	0.03	R
w_{10}	-13.57			[84.981, 125.013]	R	0.01	R
w_{11}	13.99			[84.991, 124.96]	R	0.03	R
w_{12}	-18.01			[84.991, 125.013]	R	0.01	R
				[84.951, 124.98]	R	0	R
				[85.013, 124.98]	R	0.03	R
				[84.956, 124.968]	R	0.02	R
				[85.013, 124.968]	R	0	R

Since there are four neurons in the hidden layer, then each neuron provides a "line" (actually, a hyperplane) to separate the classes. These lines can be seen in the Fig .7. The values of approved inputs form the vertices of a rectangle. To better understand the proposed idea, consider that all input values that should be approved are contained within that rectangle. If they are gone, they should be failed. It can be seen that the program executed with 20,000 epochs provides a set of lines that are able to accurately separate the inputs that should be approved (blue dots), from the rejected ones (red dots).

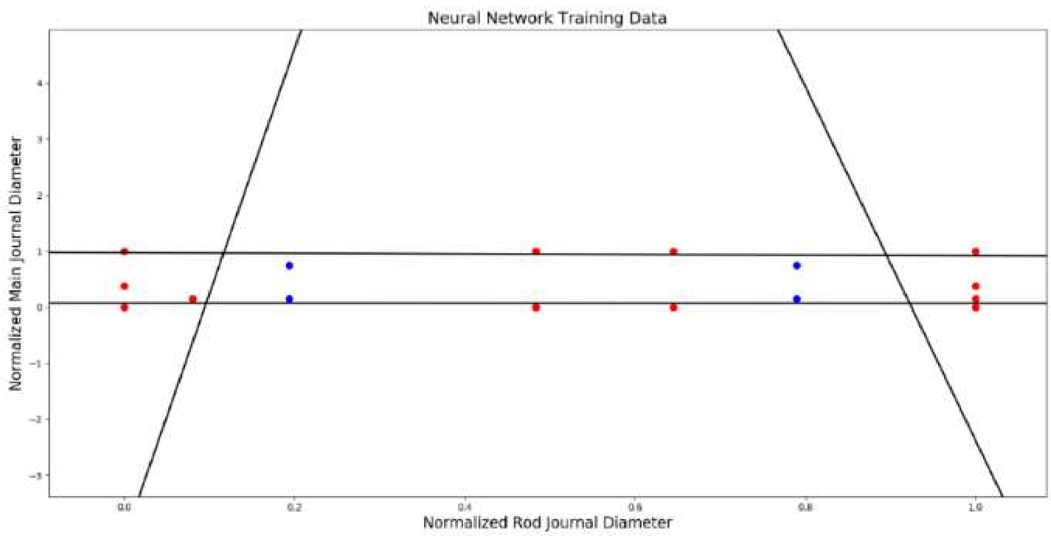

Fig.7 Separation lines show the values of diameters that are approved, in blue, and those that fail, in red

The graph showing the error value for each epoch, calculated by the equation 10, is shown in the Fig. 8. It can be seen that as the epoch increases, the error value decreases.

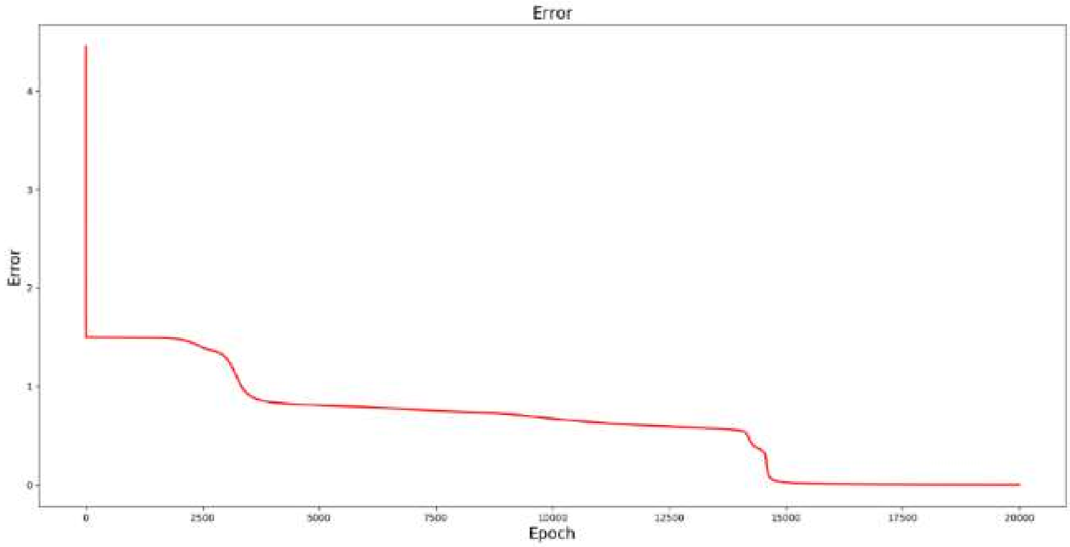

Fig. 8 Calculated error for each epoch

Table III: New samples are used in test stage to the trained classifier for 20,000 epochs.

	Input (mm)	Desirable Result	Output	Result
1	[84.968, 124.97]	A	0.63	A
2	[84.97, 124.99]	A	0.99	A
3	[84.997, 124.975]	A	0.98	A
4	[84.995, 124.98]	A	1	A
5	[84.963, 124.97]	R	0	R
6	[84.962, 124.968]	R	0	R
7	[84.963, 125.001]	R	0	R
8	[84.962, 125.001]	R	0	R
9	[85, 124.97]	R	0.05	R
10	[85.001, 124.968]	R	0	R
11	[85, 125.001]	R	0	R
12	[85.001, 125.001]	R	0	R
13	[84.981, 124.967]	R	0.33	R
14	[84.981, 125.001]	R	0.05	R
15	[85.002, 124.965]	R	0	R
16	[85.003, 125.002]	R	0	R

IV. CONCLUSION

The neural network model proposed in this work to be used to separate the diameters of a crankshaft outside the customer's specification proved to be very efficient.

The neural network for training began to deliver results as expected from 13,578 epochs, as shown in the table I. However, to further increase accuracy, all training was carried out for 20,000 epochs. The values of the new values of the weights and bias, as well as the output values, they were shown in the table II.

According to the table III, diameter values with only 0.001 mm below the specified, as in the inputs 6 and 8 where the diameter of the journal rod was 84,962 mm, the neural network was able to fail the crankshaft. Diameter values above just 0.001 mm (85.001) were also disapproved, as were the diameters of the journal rod of the inputs 10 and 12.

The same thing happened for the main journal below only 0.001 of the specification values (124.967mm) as in the input 13 or above (125.001 mm), as in the inputs 11 and 14.

Diameter values within the specification, as in inputs 1 to 4, were also considered approved.

All the desired diameters that were approved and failed had the same result in the proposed neural network, even when the precision was very small, almost on the threshold of being approved or failed. Thus, it is concluded that this neural network could be used in a production line to separate the crankshafts that must be separated from the production line for some possible rework or even disposal.

References

[1] F. Taglialatela-Scafati, M. Lavorgna, E. Mancaruso, B. M. Vaglieco. *Nonlinear Systems and Circuits in Internal Combustion Engines: Modeling and Control. Spring Nature*. 2018.

[2] R. M. Furtado. *Identificação de Falhas Estruturais Usando Sensores e Atuadores Piezelétricos e Redes Neurais Artificiais*. Tese de Mestrado. Unesp. Ilha Solteira, Brazil.2004.

[3] E. H. Dorries. *TechOne: Automotive Engine*. Thomas Delmar Learning. 2005.

[4] Des Hammill. *The Ford SOHC Pinto and Sierra Cosworth DOHC Engines High-peformance Manual*. January 10, 2003. ISBN-10: 1903706785. P.44.

[5] Detroit Diesel Engine Troubleshooting. Secção 17 – Crankshaft. http://www.detroitmanuals.info/series-60/040207.html . Acess: 13.11.19.

[6] K. Gurney. *An introduction to neural network*. Taylor & Francis e-Library, 1997.

[7] L. Fausett. *Fundamentals of Neural Networks - Architectures, Algorithms and Applications*. Pearson. 1992.

[8] J. M. Barreto. *Introdução as Redes Neurais Artificiais. Laboratório de Conexionismo e Ciências Cognitivas* UFSC -Departamento de Informática e de Estatística 88040-900 - Florianópolis – SC, 2002. pp.11.

[9] Thomas. *An Introduction to Neural Networks for Beginners*. 2017.

[10] D. Kriesel. *A Brief Introduction to Neural Networks*. 2005.

[11] F. Günther, S. Fritsch. *Neuralnet: Training of Neural Networks*. The R Journal, Volume 2/1, June 2010.

[12] N. da Silva, D. H. Spatti, R. A. Flauzino. Editora Artliber. *Redes Neurais Artificiais. Para Engenharia e Ciências Aplicadas*. Fundamentos Teóricos e Aspectos Práticos. 2ª edition. 2016.

[13] C. C. Aggarwal. *Neural Networks and Deep Learning*. Springer. 2018.

[14] S. Chakraverty, S. Mall. *Artificial Neural Networks for Engineers and Scientists*. Solving Ordinary Differential Equations. CRC Press. 2017.