Applying Artificial Intelligence in Retrieving Design Solution

Y. Moubachir, B. Hamri, S. Taibi
Q.S.M. Lab, Mohammed V University in Rabat, EMI, Morocco

***Abstract*—Design is a very important step in the product life cycle, because it is generally the key for the success or the failure of the product. The field of design theories and methodologies is fill with theories and methods that have been taught and developed throughout the years. Most of them relay on subdividing the design process into phases, where the transition between each two phases relay on using some design tools. One of the main challenges of nowadays is to find a way for the integration of artificial intelligence (AI) in the design process. This integration could be very benefic, due to the fact that AI can learn quickly the relationship between input and output of any phenomena, and it can also give us a prediction of the behavior, if the inputs parameters vary. In our previous work we shaded the light, on how we can improve the transition between design phase by storing and retrieving design solutions using morphological analysis and design tools like DFX. In this work, we present a deferent methodology to perform this transition which relay on using an artificial intelligence tool called Artificial Neural Networks (ANN) instead of morphological analysis to retrieve the right design solution. To illustrate this method, we will take the same example from our previous work and will show how we can use ANN to learn and predict the right design solution.**

***Keywords*—Artificial intelligence; ANN; design methodologies; DFX; morphological analysis**

I. Introduction

The design phase is a very important step in the product life cycle [1], even if it is generally costing for approximately 5% of the global cost of the project. But the decisions made in this step influence 70% of the global cost of the project [2]. Nowadays, product design is becoming a very challenging task due to the fact that a design team need to have a deep understanding of a variety of fields of knowledge (technological, social, cultural...) to ensure the success of their products once it’s putted in the market [3]. In this scope the field of design theory and methodologies is rich of research and result that have been used and taught in industry and education [4], which are used as guidelines to help identifying right steps to take, for the purpose of identify the shape of the product that will succeed. Those facts lead to the growing complexity of products [5] which make the design process a very difficult step. This complexity is related to the fact that the number of parameters that we need to take in consideration in our product are increasing exponentially.

The application of Artificial neural network (ANN) in the design phase of the product life cycle is still in the earlier stage of development [6] [7]. And it’s integration in the design process is systematically growing in many field of design [8]. This is due to the fact that ANN has a big potential to help reducing complexity in the design process, and leading the designers to converge quickly to the wright design solution [9].

The goal of this work is to present a new methodology that will help in the integration of artificial neural network (ANN) in the design process. To achieve this goal, we will start from our previous work which had as purpose, to store and retrieve design solution based on morphological analyses and design tools [10], in that work the idea was to improve the transition between design phases, by relaying on storing technical solution based on some criterions that are already predefined, and then retrieving the right one based on the selected criterions, using morphological analysis method. In this work, the main contribution relay on using artificial neural network for retrieving design solution instead of morphological analysis. This can be achieved based on a deferent modelization of the problem. To illustrate this new method, two type of modelization of the problem will be proposed and compered.

In this scope our work is built up as bellows. In Section II, a short presentation of the main idea for the previous work, and the illustrative example that will be used. Section III, depicts the functioning and the sizing of the artificial neural network are presented. Section IV, is devoted to show the two types of possible modelization of the ANN that can be used to retrieve design solution works. Section V sums up our contributions and outlines some possible upcoming work.

II. Previous Work

The main idea of the previous work was to show how we can identify a suitable design solution based on the desired values of set of inputs criterions (Fig .1).

The illustrative example was the “mounting of spur gear in a shaft”, as a design problem where we have different possible solutions regarding the inputs criteria. For that example, eight design solutions were proposed as shown in Fig. 2 and each design solution has different characteristics:

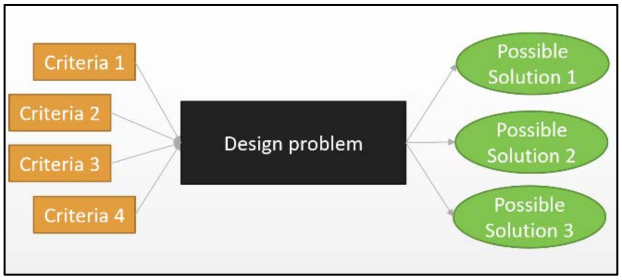


Fig. 1. Configuration of the Problem.

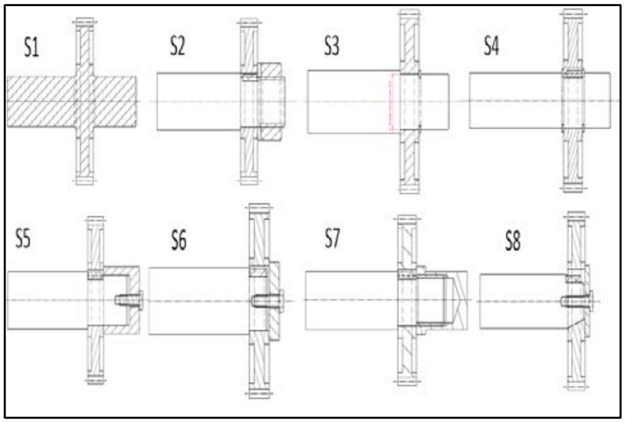


Fig. 2. Design Solutions for Mounting Spur Gear in a Shaft.

- S1: the spur gear is directly mounted in the shaft.
- S2, S4, S5, S6 and S7: A key is mounted between the shaft and the gear. The shaft and the spur gear hole have a cylindrical shape, with different axial blocking solutions.
- S3: An interference fit is used between the gear inner hole and the shaft.
- S8: A key is mounted between the shaft and the gear, and the shaft and the spur gear hole have a tapered shape.

For this illustrative example the developed criteria are Manufacturability index [11] (MI), Number of commercial part, Total number of part, manual assembly time [12] and Reparation cost index. For each solution we calculate the possible value regarding each preselected criterion. The results of calculations are given in Table I.

TABLE I. RESULT FOR EACH SOLUTION

Solutions	MI	Commercial parts	N° of parts	(α+β)/720	Reparation index
S1	0.31	0	1	0	1.00
S2	0.30	3	5	0.5	0.36
S3	0.49	1	3	0.25	0.41
S4	0.55	3	5	0.5	0.34
S5	0.28	2	5	0.5	0.35
S6	0.32	2	5	0.5	0.34
S7	0.30	1	4	0.5	0.41
S8	0.14	3	5	1	0.48

III. Artificial Neural Network

The artificial neural network is a mathematical tool that was developed based on the functioning model of the human brain. This mathematical tool is used in many fields of artificial intelligence, like image recognition, machine learning, prediction and classification. In this work ANN will be used as machine learning tool, where the aim is to learn the obtained values of each solution, in order to build a mathematical model which will give us for each set of criterions, the most suitable possible solutions.

A. Functioning of the Neural Network

An artificial neural network is a model that generally consists of three types of layers. An input layer, several hidden layers and an output layer (Fig. 3). For each neuron in the network (Fig. 4), the output value is calculated as follow (1):

$$x_j = \sigma\left(\sum_{i=1}^{n} w_{ij} \cdot O_i\right) \qquad (1)$$

Where :

x_j : The neuron output value.

w_{ij} : Value of the weight which links two neurons.

O_i: Neuron output value of the preceding layer with (O_n=1 bias).

σ : Transfer function.

There are several type of transfer function that can be used in the neural network, like the Gaussian function (2), tangent hyperbolic (3), linear (4) or the sigmoid function (5). This last one is generally the mostly used transfer function [13].

$$\sigma(t) = exp(-t^2) \qquad (2)$$

$$\sigma(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \qquad (3)$$

$$\sigma(t) = t \qquad (4)$$

$$\sigma(t) = \frac{1}{1 - e^{(-t)}} \qquad (5)$$

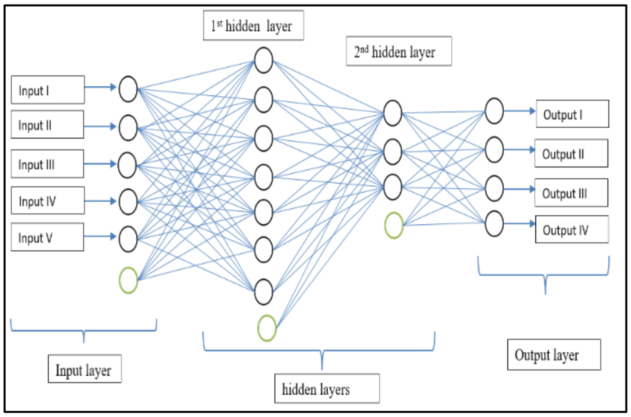


Fig. 3. Neural Network Model.

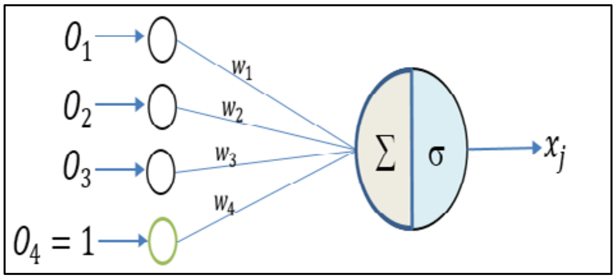


Fig. 4. The Model of the Activation of each Neuron.

B. Sizing the Neural Network

The purpose of sizing the neural network is to identify the most suitable neural network for our case of study. This operation begins by identifying the number of neurons in each layer and the number of hidden layers. in other words, the goal is to find the smallest size of the network that has an equation that generalizes the best the relationship between inputs and outputs (Fig.5 (a)) to do so two things must be avoided:

- The overfitting: which means that the network used is too large, so that the equation obtained has integrated the errors in the training set of the model (Fig. 5(b)).
- The underfitting: which means that the network used is too small, so that the equation obtained is not adequate with the actual model (Fig. 5(c)).

Thus, to properly size the network, we will start with the smallest possible network. We will then calculate the error obtained. If this error is too large, we will continue to increase the size of the network until the error becomes too small.

C. The used Neural Network Model

The use of a neural network begins with the learning process. The learning process begins by a set of input and output data are given to the network for learning purpose. In our case we will give the network, for each set of output values S_k (solution S) a set of input values corresponding to it I_m (criterion) where « m » is the number of inputs and « k » is the number output. The goal of this model is to identify from the training data the values of the w_{ij} of the mathematical equation which links the input values to output values. The learning of the network is done according to the following steps :

1) The identification of the input values and the corresponding output values (training set).

2) Initialization of the values of w_{ij}.

3) Calculating the error E : which is the square of the difference between the results of the neuron values and real the output (6).

4) Using the Gradient Descent Method (Several variants of the gradient decent are possible, in our example we will use the delta-bar-delta method [5]) to modify the values of w_{ij}.

5) Return to step 3 until the error becomes small.

$$E = 1/2\left(\sum_i^k (S_i - x_i)^2\right) \qquad (6)$$

With :

k : is the number of outputs.

Si : the value of the output of the training set.

xi : the value of the output obtained by the neural network.

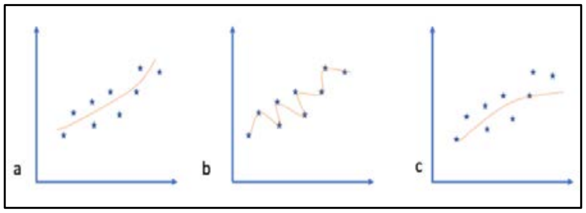


Fig. 5. (a) Proper Result, (b) Overfitting, (c) Underfitting.

IV. Learning Model

In this section we present the two learning models for the studied problem “mounting a pinion in a shaft”. The first model consist of using a network of five inputs and three outputs, and the second one consist of using a network of five inputs and eight outputs. Then we will compare the obtained results of the two models.

In our case we will refer to the output values of the network by S_k, which refer to one of the eight design solutions presented in Fig. 2. For the set of input values related to the S_k solution, we will refer to them by I_k, this set is composed of 5 values which are corresponding to the values obtained by each of the S_k solution in regard to the five criterions I_k. For example, I_1 related to the S_1 solution is {0.3 ;1 ;0 ;1 ;0 ;1}.

A. First Model

For this first learning model, a neural network composed of 5 inputs and 3 outputs will be used. In this case the five inputs are corresponding to the 5 values of the criterions. The input values for each criterion should be standardized (there values should be between -1 and 1) [5], this step is very important so the network can give us reliable results. For this, we will divide the values of the number of commercial parts and the total number of parts by 10, and for the other criterions they are already standardized.

The output values are three neurons so each solution will be encoded as follow:

S1 corresponds to the value (0,0,0).

S2 = (0.0.1)

S3 = (0.1.0)

S4 = (0.1.1)

S5 = (1.0.0)

S6 = (1.0.1)

S7 = (1.1.0)

S8 = (1.1.1)

The python programing language and Tcl as graphical user interface (Fig. 6) were used to do the learning with a developed backpropagation algorithm. The weights values of our network were calculated using the Delta-Bar-Delta method [5].

To size the network, two possible networks were evaluated. the first network “5.2.3” consists of two neuron in the hidden layer and the second one “5.3.3” is composed of three neuron in the Hidden layer (Fig. 7).

The details of the obtained results are for each network are:

1) For neural network “5.2.3”:

- Learning parameter: Learning rate: 0.1; Momentum: 0.7; Delta-bar-delta parameter: ; κ=0.05; θ=0.3; ϕ=0.2
- Global error = 0.073.
- Weights: [array([[-6.9558979, -10.7009645, 17.15070722, 9.81960633, -12.31982889, -

1.65349976],[1.7778629, -8.79354065, 17.27106976, -3.2645242 , 16.7249306 , -13.02356945]]), array([[35.5968495 , 34.0311664 , -37.00298035],[-5.08638478, -8.1145575 , 5.80673535],[5.01774992, -49.84129666, 4.49115069]])].

- Results

  [S1] : [4.48e-02 9.21e-02 2.28e-20]≈ [0,0,0]

  [S2] : [1.69e-01 5.97e-01 9.71e-01] ≈ [0.0.1]

  [S3] : [2.67e-14 9.89e-01 8.41e-02] ≈ [0.1.0]

  [S4] : [5.08e-05 8.23e-01 8.54e-01] ≈ [0.1.1]

  [S5] : [9.83e-01 3.33e-01 4.27e-01] ≈ [1.0.0]

  [S6] : [9.67e-01 3.66e-01 6.00e-01] ≈ [1.0.1]

  [S7] : [8.39e-01 3.89e-01 8.63e-02] ≈ [1.1.0]

  [S8] : [9.27e-01 4.43e-01 9.74e-01] ≈ [1.1.1]

2) For Neural network"5.3.3":

- Learning parameter: Learning rate: 0.1 ; Momentum: 0.7 ; Delta-bar-delta parameter: κ=0.05; θ=0.3; ϕ=0.2.
- Global error = 5.96 e-10.
- Weights :[array([[21.28907954, 7.92106131, -16.21637106, 9.40439058, 0.42501641, -5.98126962],[-23.6018077, -14.94841144, -9.47059949, -11.06965724, 29.07147555, 9.45002567], [-7.3583812 , 22.28383593, -1.5479036 , -16.2858285, 6.4895823, 3.67820648]]), array([[-22.88075349, 4.32776792, -33.44627925, 25.59618602], [37.39235073, 15.50577841, -22.68797391, -17.65045261],[36.05731457, -54.60337139, -8.63428378, 11.54696872]])].
- Results:

  [S1] : [8.05 e-06 1.13 e-05 1.51 e-17]

  [S2] : [3.18 e-05 1.87 e-09 9.99 e-01]

  [S3] : [4.68 e-06 9.99 e-01 2.34 e-05]

  [S4] : [2.55 e-05 9.99 e-01 1.00 e+00]

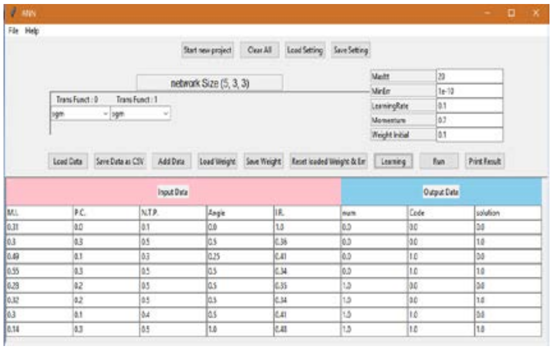
  [S5] : [9.99 e-01 1.81 e-06 4.69 e-05]

  [S6] : [9.99 e-01 4.71 e-05 9.99 e-01]

  [S7] : [9.99 e-01 9.99 e-01 8.74 e-13]

  [S8] : [9.99 e-01 9.99 e-01 9.99 e-01]

From this study it is noted that the network "5.3.3" provides a global error around 5.96 E-10 (Fig. 8) while the network "5. 2 .3" gives a global error (Fig. 9) of 0.073. So we can say that the network " 5.3.3 " is the most suitable for our example, because it is the smallest network that provides good results.



network Size (5, 3, 3)

Trans Funct : 0: sigm
Trans Funct : 1: sigm

MaxIt	20
MinErr	1e-10
LearningRate	0.1
Momentum	0.7
Weight Initial	0.1

Input Data					Output Data		
M.I.	P.C.	N.T.P.	Angle	I.R.	num	Code	solution
0.31	0.0	0.1	0.0	1.0	0.0	0.0	0.0
0.3	0.3	0.5	0.5	0.36	0.0	0.0	1.0
0.49	0.1	0.3	0.25	0.41	0.0	1.0	0.0
0.55	0.3	0.5	0.5	0.34	0.0	1.0	1.0
0.29	0.2	0.5	0.5	0.35	1.0	0.0	0.0
0.32	0.2	0.5	0.5	0.34	1.0	0.0	1.0
0.3	0.1	0.4	0.5	0.41	1.0	1.0	0.0
0.14	0.3	0.5	1.0	0.48	1.0	1.0	1.0

Fig. 6. The Developed Program for the Learning.

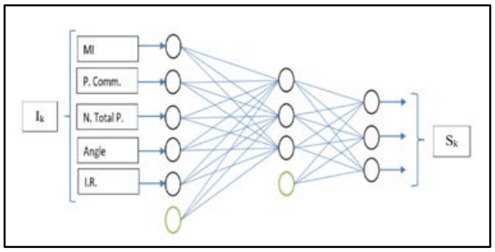


Fig. 7. The (5.3.3) Network used for the Learning of the Technical Solution.

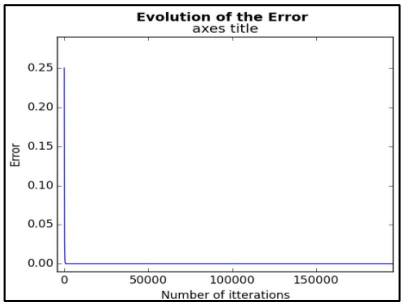


Fig. 8. The Evolution of the Error after each Iteration using a "5.3.3" Network.

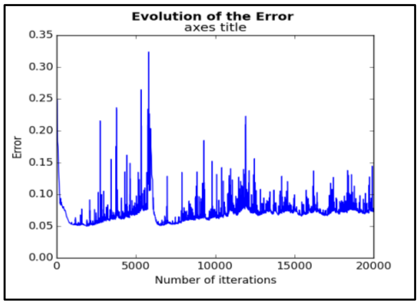


Fig. 9. The Evolution of the Error after each Iteration using a "5.2.3" Network.

B. Second Model

This second learning model is a neural network with 5 inputs and 8 outputs. The five inputs correspond to the values of the chosen criteria and the eight neurons were chosen for the output values. In this case, each output neuron corresponds to a given solution, if an output neuron obtains the value of 1, then this solution is the most suitable in regards to the given values of the inputs criteria.

To size this network, two networks were evaluated the first one is with one neuron in the hidden layer, and the second one with two neurons in the hidden layer (Fig. 10). Details of the obtained results are:

1) For Neural network“5.1.8”:

- Learning parameter: Learning rate: 0.1 ; Momentum: 0.7 ; Delta-bar-delta parameter: κ=0.05; θ=0.3; ϕ=0.2.
- Global error: 0.06801684134122636.
- Weights:[array([[3.66954535e-02, -1.02956629e+02, 7.11856022e+01, -2.84464136e+01, 3.84737574e-01, 2.04345583e+00]]), array([[3.05821270e+03, -3.05509115e+03], [-5.17925747e+00, -6.91778960e-01], [5.05589009e+01, -5.12220784e+01], [-5.17697254e+00, -6.91783745e-01], [1.76212801e+00, -3.19181346e+00], [1.76002122e+00, -3.18999900e+00], [5.05586156e+01, -5.12217948e+01], [-1.55159832e+04, 4.93991111e+00]])].
- Results:

  [S1] : [9.48e-01 2.81e-03 3.39e-01 2.81e-03 1.93e-01 1.93e-01 3.39e-01 0.0]

  [S2]: [0.00 3.32e-01 5.87e-23 3.32e-01 3.95e-02 3.95e-02 5.87e-23 5.05e-03]

  [S3] : [3.64e-02 2.84e-03 3.16e-01 2.84e-03 1.92e-01 1.92e-01 3.16e-01 0.00]

  [S4] : [0.00 3.32e-01 5.87e-23 3.32e-01 3.950e-02 3.95e-02 5.87e-23 4.98e-03]

  [S5] : [2.16e-64 3.62e-03 4.15e-02 3.62e-03 1.80e-01 1.80e-01 4.15e-02 0.00]

  [S6] : [1.54e-64 3.62e-03 4.13e-02 3.63e-03 1.82e-01 1.80e-01 4.13e-02 0.00]

  [S7]: [3.64e-02 2.84e-03 3.16e-01 2.84e-03 1.92e-01 1.92e-01 3.16e-01 0.00]

  [S8]: [0.00 3.33e-01 5.68e-23 3.33e-01 3.94e-02 3.95e-02 5.68e-23 9.92e-01]

2) For Neural network“5.2.8”:

- Learning parameter: Learning rate: 0.1 ; Momentum: 0.7 ; Delta-bar-delta parameter: κ=0.05; θ=0.3; ϕ=0.2
- Global error: 2.16e-08
- weights: [array([[-11.72500802, -13.56629037, 12.12204135, 20.69551574,7.94683641, -11.8472096],[-22.54860196, -49.77196842, 0.89553372, 5.18327433, 17.64766934, 5.37958571]]), array([[-23.52660555, 23.42517761, -13.33842946], [40.44144954, -435.47791596, -10.05392598], [-525.18616036, -19.43990113, 11.30272584], [-43.84693704, -145.95717923, 10.66856872], [244.01849652, -103.39034649, -169.71850489], [153.29525942, -326.16577446, -78.25786574], [23.84288311, 116.50855895, -125.77287833], [106.3662878 , -6.67651725, -92.49732366]])]
- Results:

  [S1]: [9.99e-1 3.46e-194 1.14e-004 1.62e-59 3.80e-119 3.02e-176 9.88e-5 1.02e-43]

  [S2] : [3.37e-11 9.99e-1 1.74e-100 6.81e-5 7.50e-26 2.64e-4 1.50e-50 1.05e-19]

  [S3] : [3.83e-5 7.49e-31 9.99e-1 9.42e-5 1.91e-80 5.95e-54 1.92e-48 3.02e-41]

  [S4] : [6.72e-7 1.92e-4 2.73e-4 9.99e-1 1.69e-70 3.05e-32 5.78e-55 3.50e-39]

  [S5] : [5.09e-13 4.66e-21 3.69e-178 3.15e-21 9.99e-1 2.81e-4 3.93e-39 1.07e-4]

  [S6] : [5.91e-13 2.45e-4 1.30e-153 4.96e-13 4.74e-4 9.99e-1 3.60e-45 3.23e-9]

  [S7] : [3.017e-5 3.01e-175 6.42e-198 2.61e-74 2.59e-28 9.33e-117 9.99e-1 1.77e-4]

  [S8] : [1.37e-8 4.90e-139 1.14e-230 6.47e-66 1.98e-4 1.29e-81 1.84e-4 9.99e-1]

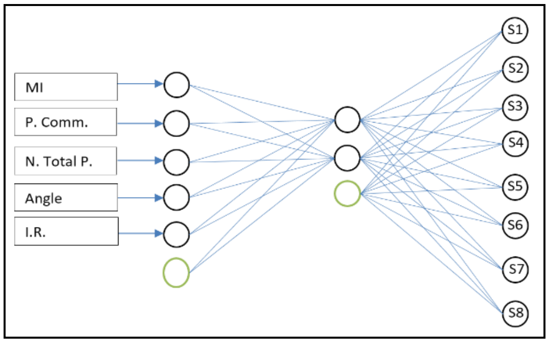


Fig. 10. A (5.2.8) Net used for the Learning of the Technical Solution.

According to this study, we notice that the network “5.2.8” gives a total error of about 2.16E-08 (Fig. 11) while the network “5. 1. 8” (Fig. 12) gives a total error of 0.068. So we can say that the network “5. 2. 8” is the most suitable for our example.

C. Synthesis

As a summary, we can see that the first network “5.3.3” is a network that was able to record the relationship between the elements of input and outputs, with a network that contains fewer parameters (13 neurons). But the downside of this model is that we are unable to identify the nearest solution if we deviate the values of the inputs criteria. On the other hand, the network “5.2.8” is a network which contains more parameters. But in the other hand, it has the advantage of giving the nearest solution if we deviate the values of the inputs criteria.

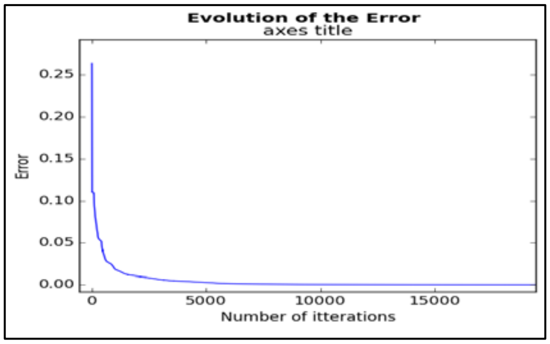


Fig. 11. The Evolution of the Error after each Iteration using a "5.2.8" Network.

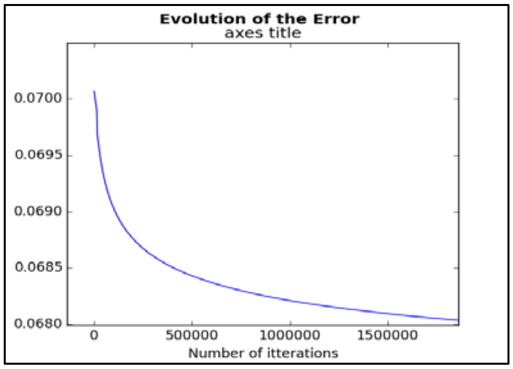


Fig. 12. The Evolution of the Error after each Iteration using a "5.1.8" Network.

V. Discussion and Conclusion

The use of neural network method in the retrieval of the wright design solution has a great practical value to industry, because it has the potential of saving time and cost in the design processes [14].

In comparison to our previous work [6], where we used morphological analysis to retreive design solution, we can see that the use of ANN has the advantage of predecting possibles solutions if we change the values of the inputs criterions. Where in our previouse work, the values of the criterions were fixed and cannot be changed. In this case, if our design problem had different values for at least one of the criterions the method cannot be used, but in the other hand the use of ANN help us to overcome this issue.

The purpose of this work is to present a method to improve the identification of technical solutions that will meet a set of design parameters that were pre-identified in the previous phase of the design. The proposed method is an idea that can be applied in several other problems like, selection of bearing type, Material selection or the choice of the machining process. The advantage of this method is that, it allows us to make a quick and visual choice of the suitable solution by using the power of ANN. But the drawback is that this method requires a huge work in advance to include all kinds of possible solutions in regards to the developed criteria.

References

[1] K. Ulrich, DESIGN: Creation of Artifacts in Society. the University of Pennsylvania., 2011.

[2] S. Munro, D. Foreman, D. McCarthy, I. Chambers, and Thomas, "Lean Design : Value Quality Profit," Munro & Associates, Inc., 2013.

[3] D. G. Ullman, The Mechanical Design Process, Fourth Edition. McGraw-Hill Higher Education, 2010.

[4] T. Tomiyama, P. Gu, Y. Jin, D. Lutters, C. Kind, and F. Kimura, "Design methodologies: Industrial and educational applications," CIRP Ann. - Manuf. Technol., vol. 58, no. 2, pp. 543–565, Jan. 2009.

[5] W. ElMaraghy, H. ElMaraghy, T. Tomiyama, and L. Monostori, "Complexity in engineering design and manufacturing," CIRP Ann. - Manuf. Technol., vol. 61, no. 2, pp. 793–814, Jan. 2012.

[6] H. A. Khayyat, "ANN based Intelligent Mechanical Engineering Design: A Review," Indian J. Sci. Technol., vol. 11, no. 27, pp. 1–7, 2018.

[7] Z. Zhang and K. Friedrich, "Artificial neural networks applied to polymer composites: A review," Compos. Sci. Technol., vol. 63, no. 14, pp. 2029–2044, 2003.

[8] W. Sitek and J. Trzaska, "Practical aspects of the design and use of the artificial neural networks in materials engineering," Metals (Basel)., vol. 11, no. 11, 2021.

[9] I. M. L. Ferreira and P. J. S. Gil, "Decision Support Tool for Conceptual Design using Neural Networks," 2010, no. 1.

[10] Y. Moubachir and D. Bouami, "Storing and Retrieving Design Solution in the Physical Domain Based on DFX Tools and Morphological Analysis," Procedia CIRP, vol. 34, pp. 64–68, 2015.

[11] S. K. Ong, M. J. Sun, and a. Y. C. Nee, "A fuzzy set AHP-based DFM tool for rotational parts," J. Mater. Process. Technol., vol. 138, no. 1–3, pp. 223–230, Jul. 2003.

[12] Z. Yoosufani, M. Ruddy, and G. Boothroid, "Effect of part symmetry on manual assembly times," J. Manuf. Syst., vol. 2, no. 2, pp. 189–195, 1983.

[13] S. Samarasinghe, "Neural Networks for Applied Sciences and Engineering," p. 581, 2007.

[14] I. Article, "NIRS : Large Scale ART-I Neural Architectures for Engineering Design Retrieval," neural Des. Retr., vol. 7, no. 9, pp. 1339–1350, 1994.