Estimation of Future Thickness of Carbon Steel Pipe and Curing Time of Adhesive of GRE Pipe by Using Neural Network Models

Mustafa M. Mansour
Mustafa_alansary@yahoo.com
Department of Mechanical Engineering
University of Baghdad \ Iraq

Prof. Dr. Qasim Mohammed Doos
Kasim_daws@yahoo.com

Abstract:

The main objective of this research is to estimate both of (future thickness of carbon steel pipe and curing time of Adhesive of GRE pipe) by using neural network model. Alyuda NeuroIntelligence software has been used to obtain these two models. These models will be based on multi – layer feed forward neural network and by applying two experiments for each case, the best networks have been concluded to estimate these cases. The results shows that the network with a number of hidden neurons 5 and that has been trained by conjugate gradient descent algorithm and with using logistic activation function for hidden and output layer gave good performance indication for estimating the future thickness which gave results of network output that are nearly closer to the targets, with correlation (0.9999) and R-Squared (0.9967), while the network with a number of hidden neurons 6 and that has been trained by Quasi – Newton algorithm and with using Hyperbolic Tangent activation function for hidden and output layer gave good performance indication for estimating the curing time which gave results of network output that are nearly closer to the targets, with correlation (0.9999) and R-Squared (0.9958).

Keywords: Neural network, carbon steel pipe, GRE pipe, Alyuda neuroIntelligence, curing time.

1. Introduction

Gas and oil provide more than Sixty percent of the world's primary fuel. Therefore, it isn't strange to observe that there is more than one million tons of oil and 250 million m3 of gas Consuming around the world every hour [6]. pipelines transported most of this gas and oil which have been employed as one of the most practical and low price method for large oil and gas transport since 1950 [4]. An Artificial Neural Network (ANN) is a mathematical model that tries to simulate the structure and functionalities of biological neural networks. Basic building block of every artificial neural network is artificial neuron, that is, a simple mathematical model (function). Such a model has three simple sets of rules: multiplication, summation and activation. At the entrance of artificial neuron the inputs are weighted what means that every input value is multiplied with individual weight. In the middle section of artificial neuron is sum function that sums all weighted inputs and bias. At the exit of artificial neuron the sum of previously weighted inputs and bias is passing through activation function that is also called transfer function [2].

Fig. 1 shows the working principle of an artificial neuron, while **Fig. 2** shows a simple artificial neural network. The difference between artificial neural network and other algorithms is artificial neural network can realizes the data and understands how the system works and it can predict new data than didn't presented through training [7]. There are three main fundamentally different classes of network Architectures:

(1) Feed – Forward ANN: The flowing of the information in this type is unidirectional. A unit sending information to other unit from that it does not receiving any information. Feedback loop is does not exist. They are utilized in pattern (generation, recognition, classificatin). They have fixed input and output [8].

(a) Single Layer Feed - forward Network: Here, neurons are arranged in a layers shape. The unpretentious shape of a layered networks, contain inputs layer of nodes of the source which projects directly onto the outputs layer of neurons (computation node), but not vice versa. In another word, these networks are precisely of the feed - forward types [5].

(b) Multilayer Feed-forward Networks: Multilayer NN consisting of neurons which are arranged into layer form that is arranged in this manner (input layer, hidden layer, output layer) [3].

(2) Feedback ANN: This network allows to the feedback loop.

Recurrent Network: The recurrent neural networks recognize themselves from the feed - forward neural networks in which it have at least one feedback loop. For example, the recurrent networks may consist of single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons [5].

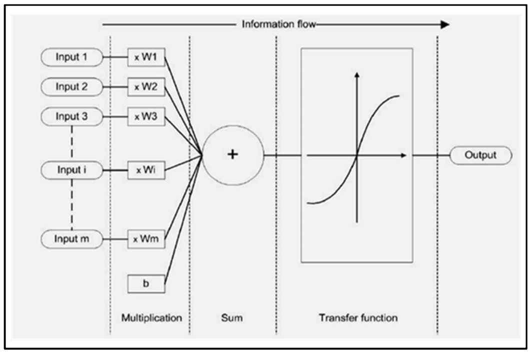

Fig. 1 The working principle of an artificial neuron

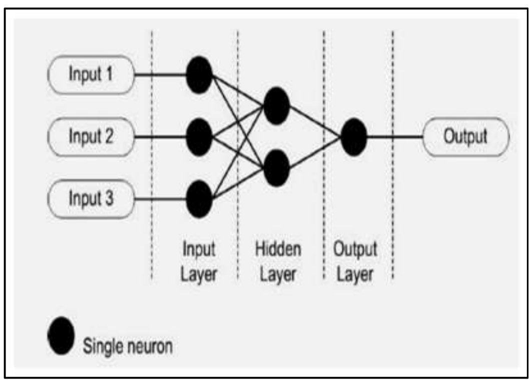

Fig. 2 A simple artificial neural network

2. Research methodology

A neural network model is proposed to estimate future thickness of carbon steel pipe and Curing time of Adhesive of GRE pipe, these models will be based on multi – layer feed forward neural network.

In this work two neural network models will be build for two cases: one to estimate future thickness of carbon steel pipe and the other to estimate the curing time of Adhesive of GRE pipe. Alyuda NeuroIntelligence software was used to obtain these two models. Alyuda NeuroIntelligence software is neural network software designed to assist experts in solving real - world problems. And it is giving a network which the outputs of its (results of this network) are nearly closer to the targets (the real results). ANN was developed and trained using the experimental data.

3. Case study

Carbon steel pipe with diameter 3 inch, transfer oil from furnace to shut drum (From propane de-asphalting unit (P.D.A) of DAURH refinery) as shown in **Fig. 3**.

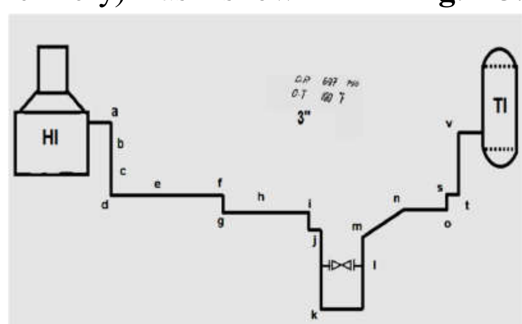

Fig. 3 Carbon steel pipe link furnaces with shut drum

Where the specified points (a, b, c etc.) are represented Periodic inspection points to find out corrosion rate. **Table .1** showed data that relate to current thickness measured every year by non - destructive inspection (ultrasonic testing).

Table .1 Current thicknesses (mm) measured for five years

Year / Check point	$t_{previous\,1}$	$t_{previous\,2}$	$t_{previous\,3}$	$t_{previous\,4}$	t_{last}
a	6.3	6.1	5.5	5.1	5
b	6.8	5.8	5.6	5.2	5.1
c	6	5.8	5.4	5.4	5
d	7.8	7.8	7.5	6.9	6.8
e	6	5.9	5.8	5..7	5.6
f	6.8	6.4	6.3	6.2	6.2
g	6.4	6.1	5.7	5.7	5.6
h	5.8	5.4	5.2	5	4.9
i	8.2	7.9	7.9	6.9	6.3
j	5.3	5.2	5.2	5.2	5
k	5.9	5.7	5.6	5.3	5.2
l	7.9	7.3	7.2	7	6.1
m	8.5	8.2	8	7.8	7.4
n	7.9	7.7	7.5	7.3	6.2
o	7.8	7.5	7.3	7.1	7
s	7.3	7.2	7.1	6.2	5.9
t	7.6	7.2	6.5	6	5.6
v	7.2	7.2	7	6.7	6.3

In this paper, two neural network models will be build to estimate two cases (future thickness of carbon steel pipe (t_{last}) by depending on the data in **Table .1**, and curing time of adhesive GRE pipe by depending on the data in **Table .2**.

Table .2 The variation of curing time with temperature

Temperature (°C)	Curing Time (hrs)
13	24
16	16
18	11
21	9
24	4.5
27	4
29	3.5
32	3
38	2.5

Alyuda neuroIntelligence software [1] will be used to estimate the two cases. Aluyda neural network software is successfully used by thousands of experts to solve tough data mining problems, empower pattern recognition and predictive modeling, build classifiers and neural net simulators, design trading systems and forecasting solutions. It supports all stages of neural net design and application. In future thickness estimation case, the model will be proposed when the input variables (previous thickness 1, previous thickness 2, previous thickness 3, and previous thickness 4) are known, these variables are the thickness of pipe for last four year. While in curing time estimation case, the model will be proposed when the input variable (temperature) is known.

4. Experimental work

In this paper two experiments (No. 1 & No.2) will be apply to reach to the best neural network model which estimate the future thickness of carbon steel pipe and another two experiments (No.3 & No.4) to reach to the best neural network model which estimate the curing time of adhesive GRE pipe as the following:

Experiment No.1

In experiment No.1 logistic activation function will be use for the hidden layer and the output layer to test the networks. **Table .3** shows the parameters that will be used for this experiment by applying the architecture search method to obtain the best design of network that will give us the best neural network model after training it, while **Table .4** shows the results of the first experiment.

Table .3 Network properties and architecture search options of Experiment No.1

Number of hidden layers	1
Hidden layer activation function	Logistic
Output error function	Sum-of-square
Output activation function	Logistic
Range of hidden neuron	2 -8
Fitness criteria	Inverse test error
Number of Iteration	2000
Architecture search	Exhaustive search

Table .4 Architecture search results of Experiment No.1

ID	Architecture	Test Error	Train Error
1	4-2-1	0.0185	0.0240
2	4-3-1	0.0199	0.0263
3	4-4-1	0.0157	0.0149
4	4-5-1	0.0121	0.0169
5	4-6-1	0.0412	0.0089
6	4-7-1	0.0325	0.0189
7	8-8-1	0.0177	0.0161

The results of the test error of the networks from **Table .4** showed that the fourth network [4 Inputs - 5 Hidden neurons – 1 Output] is the best network design for experiment no.1. The variation of test error with hidden neurons in experiment no.1 is shown in **Fig. 4**, while **Fig. 5** shows the top five tested networks.

Experiment No.2

In experiment No.2 hyperbolic tangent activation function will be use to test the networks, all the previous steps that were performed in experiment no.1 will be repeated in the same sequence in experiment no.2. **Table.5** shows the parameters that will be used in experiment No.2, and **Table .6** shows the results of this second experiment.

Table .5 Network properties and architecture search options of Experiment No.2

Number of hidden layers	1
Hidden layer activation function	Hyperbolic Tangent
Output error function	Sum-of-square
Output activation function	Hyperbolic Tangent
Range of hidden neuron	2 -8
Fitness criteria	Inverse test error
Number of Iteration	2000
Architecture search	Exhaustive search

Table .6 Architecture search results of Experiment No.2

ID	Architecture	Test Error	Train Error
1	4-2-1	0.0832	0.0355
2	4-3-1	0.0762	0.0344
3	4-4-1	0.0976	0.0439
4	4-5-1	0.0969	0.0076
5	4-6-1	0.0868	0.0399
6	4-7-1	0.0909	0.0054
7	4-8-1	0.0604	0.0277

The results of the test error of the networks from **Table .6** also shows that the last network [4 Inputs - 8 Hidden neurons – 1 Output] give smallest test error so that it is consider the best network design for experiment no.2. The variation of test error with hidden neurons in experiment no.2 is shown in **Fig. 6**, while **Fig. 7** shows the top five tested networks in experiment No.2.

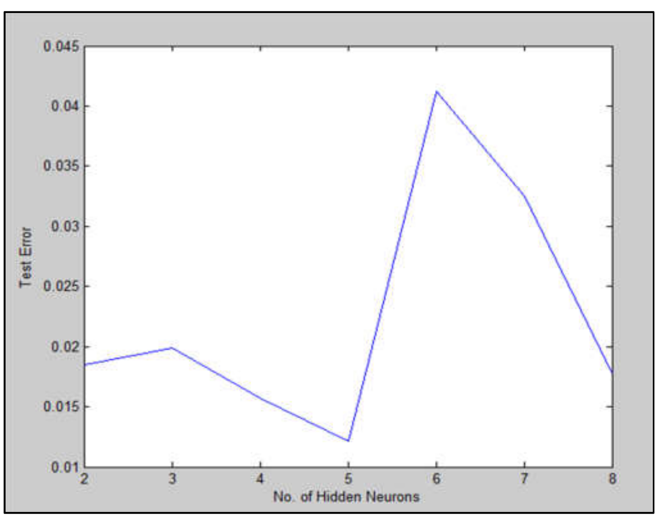

Fig. 4. The variation of test error with hidden neurons for Experiment No.1

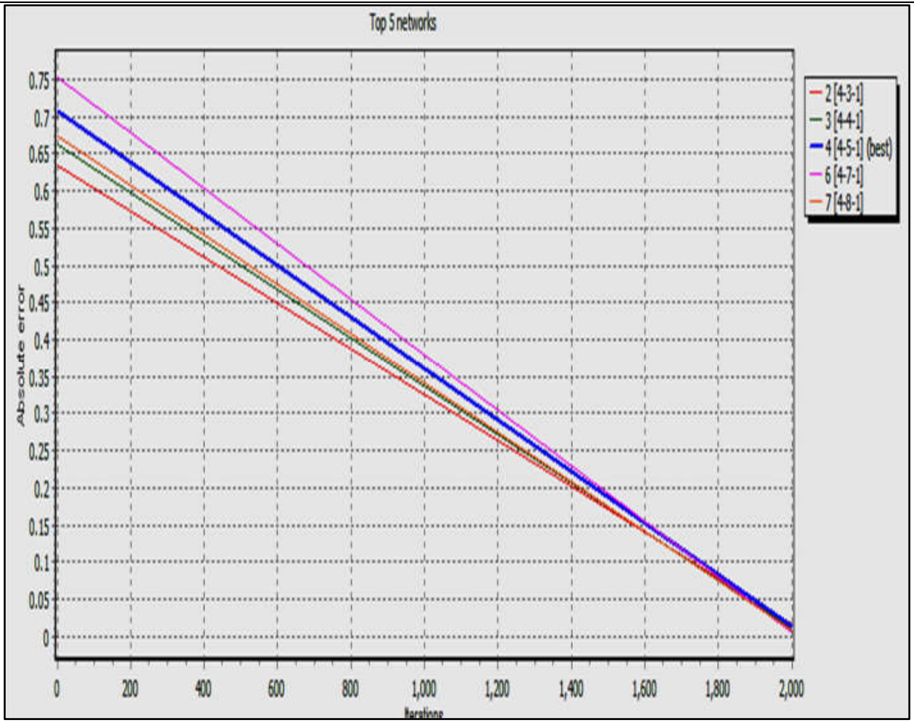

Fig. 5. The top five tested network in Experiment No.1

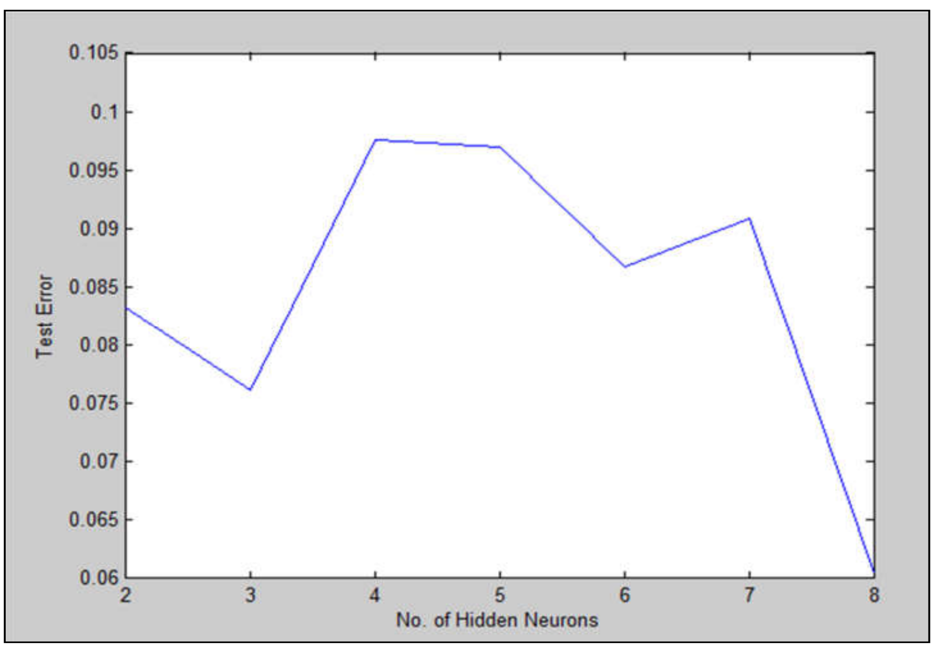

Fig. 6. The variation of test error with hidden for Experiment No.2

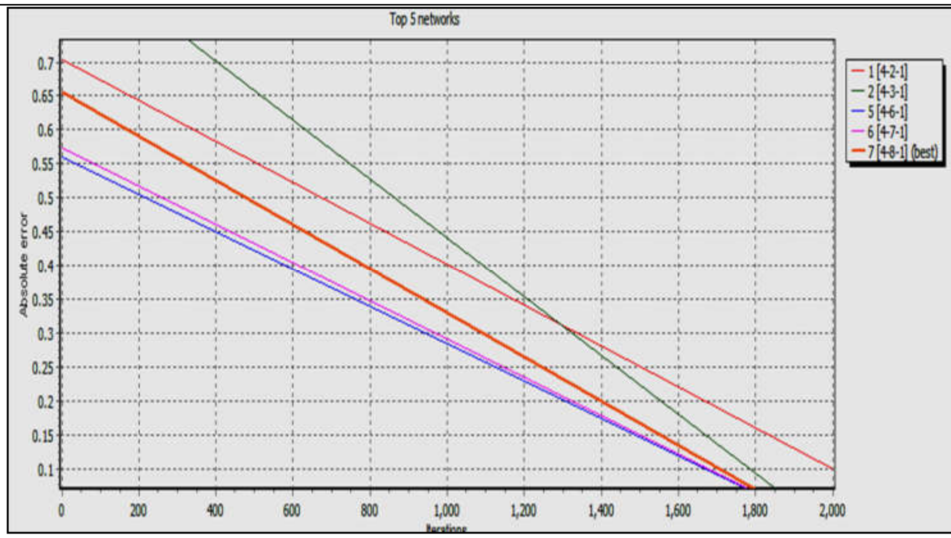

Fig. 7. The top five tested networks in Experiment No.2

Experiment No.3

In this experiment logistic activation function of hidden and output layer will be used to test the networks. To obtaining the best design of network by applying the architecture search method, same **Table .3** will be used for experiment no.3; then the results of this experiment have been obtained as shown in **Table .7**.

Table .7 Architecture search results of Experiment No.3

ID	Architecture	Test Error	Train Error
1	1-2-1	1.3729	0.1945
2	1-3-1	0.7161	0.0673
3	1-4-1	1.2078	0.1778
4	1-5-1	0.7982	0.0761
5	1-6-1	0.6404	0.0656
6	1-7-1	0.8456	0.0684
7	1-8-1	0.7444	0.1066
8	1-9-1	0.6282	0.0630
9	1-10-1	0.8526	0.2572

From the results it is obvious that the eighth network [1 Inputs - 9 Hidden neurons – 1 Output] is the best network design for experiment no.3. **Fig.8** shows the variation of test error with hidden neurons in experiment no.3, while **Fig. 9** shows the top five tested networks in experiment No.3.

Experiment No.4

The same **Table .5** will be use to test the networks. **Table .8** shows the results of this experiment.The best network design for experiment No.4 is the fifth network [1 Input - 6 Hidden neurons – 1 Output]. **Fig.**

10 and **Fig. 11** are the corresponding figures for experiment No.4.

Table .8 Architecture search results of Experiment No.4

ID	Architecture	Test Error	Train Error
1	1-2-1	0.60517	0.05075
2	1-3-1	0.55909	0.04806
3	1-4-1	0.61752	0.05297
4	1-5-1	0.60716	0.05163
5	1-6-1	0.52554	0.04682
6	1-7-1	0.54627	0.05173
7	1-8-1	0.54102	0.04882
8	1-9-1	0.52967	0.04751
9	1-10-1	0.57530	0.05740

a. Training Network

1. The best two networks [4-5-1] & [4-8-1] that were obtained in experiment no.1 and experiment no. 2 have been applied with the three training algorithms (Conjugate gradient descent, Quasi – Newton, and Levenberg – Marquartdt) and the results are given in **Table .9**.

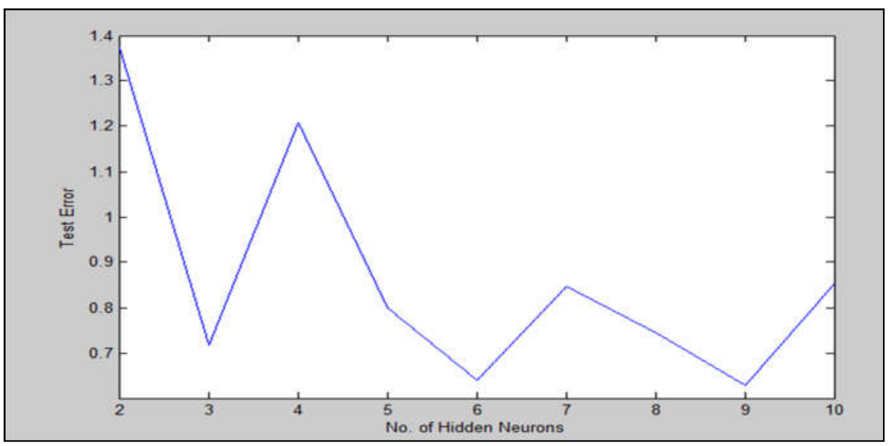

Fig. 8. The variation of test error with hidden neurons for Experiment No.3

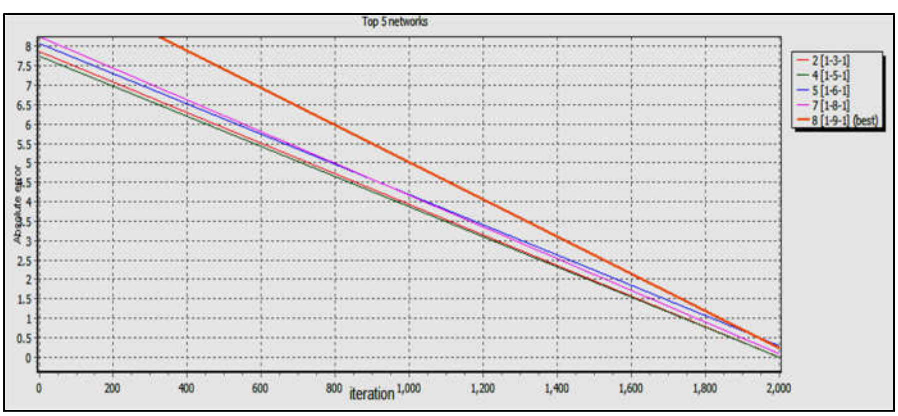

Fig. 9. The top five tested networks in Experiment No.3

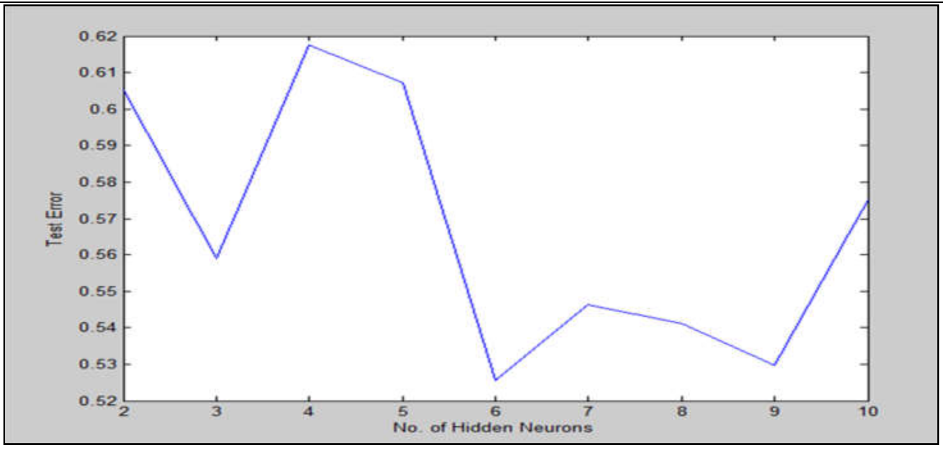

Fig. 10. The variation of test error with hidden neurons for experiment No.4

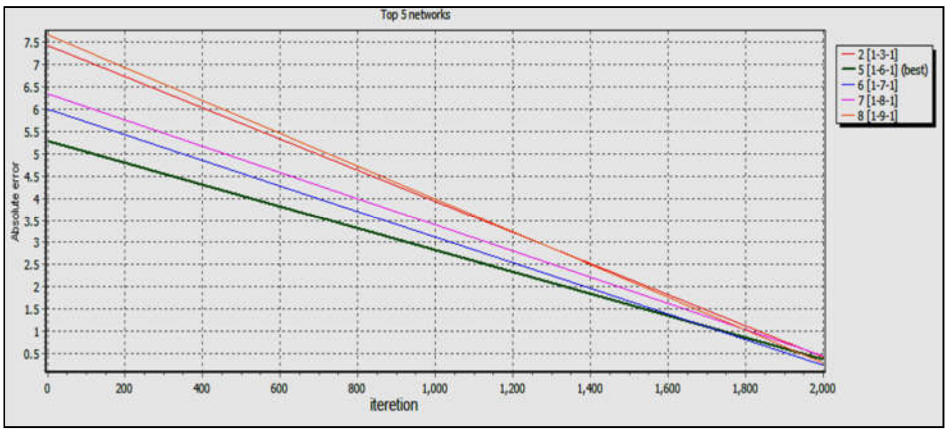

Fig. 11. The top five tested networks in experiment No.4

Table .9 Results of training network [4-5-1]and network [4-8-1] with the three training algorithms

Net No.	Architecture	Training algorithm	Hidden activation function	Output activation function	Iterations	Avg. test error	Avg. training error	Correlation
Net #1	[4-5-1]	Conjugate Gradient Descent	Logistic	Logistic	2001	0.0919	0.000863	0.99999
Net #2	[4-5-1]	Quasi – Newton	Logistic	Logistic	2001	0.05089	0.01102	0.95610 3
Net #3	[4-5-1]	Levenberg - Marquartdt	Logistic	Logistic	2001	0.03115	0.03061	0.88578

Net #4	[4-8-1]	Conjugate Gradient Descent	Hyperbolic Tangent	Hyperbolic Tangent	2001	0.10562	0.00873	0.9705
Net #5	[4-8-1]	Quasi – Newton	Hyperbolic Tangent	Hyperbolic Tangent	2001	0.09728	0.01082	0.9662
Net #6	[4-8-1]	Levenberg - Marquartdt	Hyperbolic Tangent	Hyperbolic Tangent	2001	0.04044	0.01957	0.9334

2. The two obtain networks [1-9-1] & [1-6-1] have been trained with the same training algorithms that were used with future thickness case, **Table .10** shows the results of these training.

Table .10 Results of training network [1-9-1] and network [1-6-1] with the three training algorithms

Net No.	Architecture	Training algorithm	Hidden activation function	Output activation function	Iterations	Avg. test error	Avg. training error	Correlation
Net #1	[1-9-1]	Conjugate Gradient Descent	Logistic	Logistic	2001	0.6141	0.0697	0.9839
Net #2	[1-9-1]	Quasi – Newton	Logistic	Logistic	2001	08405	0.0780	0.9786
Net #3	[1-9-1]	Levenberg - Marquartdt	Logistic	Logistic	2001	0.4433	0.0578	0.9914
Net #4	[1-6-1]	Conjugate Gradient Descent	Hyperbolic Tangent	Hyperbolic Tangent	2001	0.5660	0.0471	0.9898
Net #5	[1-6-1]	Quasi – Newton	Hyperbolic Tangent	Hyperbolic Tangent	2001	2.21E-12	0.0011	0.9999
Net #6	[1-6-1]	Levenberg - Marquartdt	Hyperbolic Tangent	Hyperbolic Tangent	2001	0.4502	0.0784	0.9951

5. Results and Discussion

In the estimation of future thickness of carbon steel pipe case:

When the performance of the two structures with the three different training algorithms has been examined, the network with a number of hidden neurons 5 and that has been trained by conjugate gradient descent algorithm and with using logistic activation function for hidden and output layer gave good performance indication. The final NN model for estimation future thickness of carbon steel pipe is shown in **Table .11**. **Fig. 12** shows neural network architecture of this model.

Table .11 Neural network model parameters for estimation future thickness of carbon steel

Parameters	Value
Number of hidden layer	1
Number of input neurons	4
Number of output neurons	1
Number of hidden layer neurons	5
Hidden layer activation function	logistic
Output layer activation function	logistic
Training algorithm	Conjugate Gradient Descent
Iterations	2000

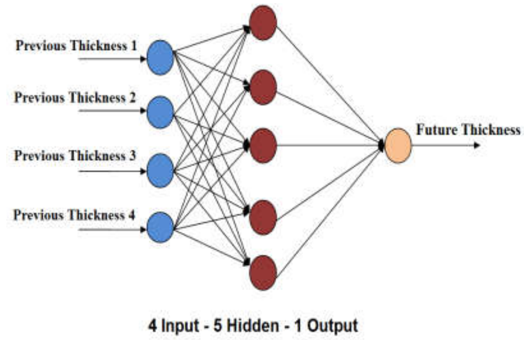

Fig. 12. Neural network architecture for estimation future thickness of carbon steel pipe

The model of neural network that was proposed to estimate future thickness which was illustrated in **Table.11** gave results of network output that are nearly closer to the targets, with correlation (0.9999) and R-Squared (0.9967), when the correlation is a statistical measure of strength of the relationship between the actual values and network outputs, the closer correlation to 1 is the stronger linear relationship, and the R-Squared Statistical ratio that compares model forecasting accuracy with accuracy of the simplest model that just used the mean of all target values is the forecast for all records. The closer this ratio to 1 the better is the model. **Table .12** shows the values of the target and the network output, absolute error (AE), as well as the absolute relative error (ARE) between the two for each pattern, (ARE is an error value that indicates the "quality" of the neural network training, the smaller the network error is, the better the network had been trained). **Table .13** illustrates the magnitude of mean, maximum, and minimum of the targets, output, absolute error, and absolute relative error, **Fig. 13** shows future thickness of carbon steel pipe using the proposed neural network model versus the target, while **Fig. 14** shows the scatter plot of target and network output. The two figures showed the amount of convergence between the two values.

Table .12 The target and network response of estimation future thickness

W.P No.	Target	Output	AE	ARE
1	5	5.004055	0.004055	0.081109
2	5.1	5.099624	0.000376	0.007378
3	5	4.996651	0.003349	0.066979
4	6.8	6.799849	0.000151	0.002221
5	5.6	5.626765	0.026765	0.47794
6	6.2	6.199852	0.000148	0.002391
7	5.6	5.600507	0.000507	0.009056
8	4.9	4.915768	0.015768	0.3218
9	6.3	5.909319	0.390681	2.207430
10	5	4.998414	0.001586	0.031724
11	5.2	5.19879	0.00121	0.023261
12	6.1	6.100645	0.000645	0.01057
13	7.4	7.39647	0.00353	0.047696
14	6.2	6.200227	0.000227	0.003667
15	7	7.000984	0.000984	0.014055
16	5.9	5.898164	0.001836	0.031119
17	5.6	6.399947	0.799947	3.214191
18	6.3	6.299328	0.000672	0.010664

Table .13 Summary of Table .12

	Target	Output	AE	ARE
Mean	5.844444	5.869187	0.180691	0.030754
Min.	4.9	4.915768	0.000148	0.002221
Max.	7.4	7.39647	0.799947	3.214191

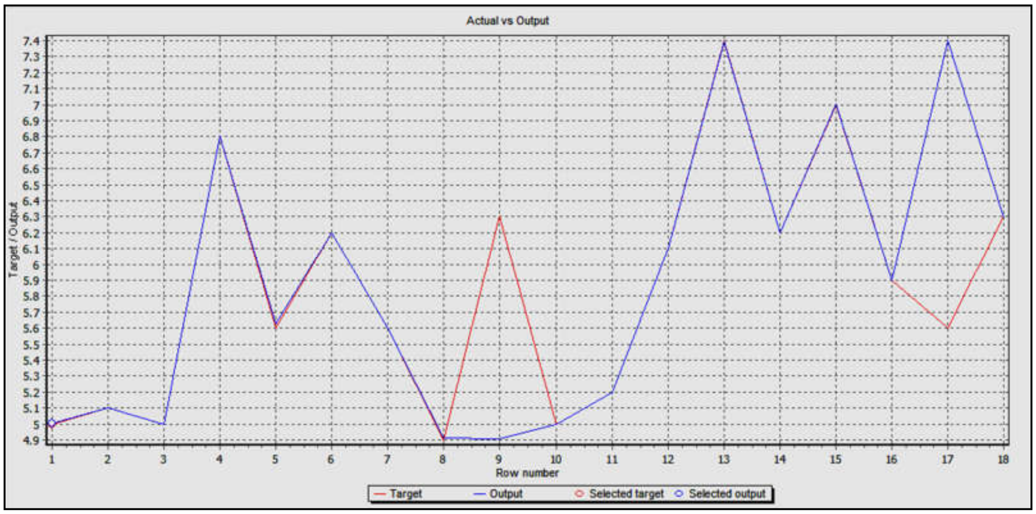

Fig. 13. The future thickness using the proposed neural network model vs. the target

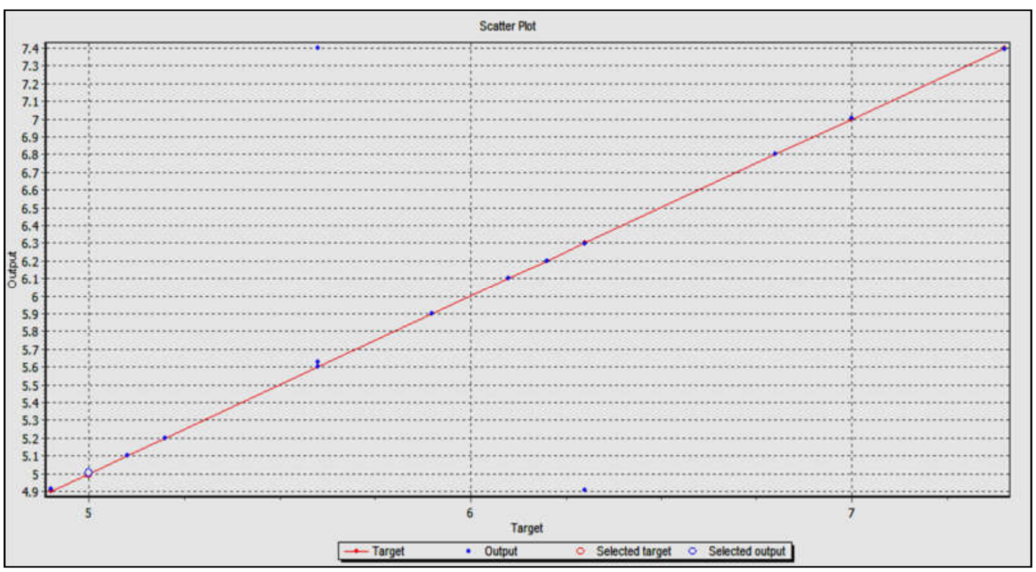

Fig. 14. The scatter plot of target and network output of future thickness

In the estimation of curing time of adhesive GRE pipe case:

The results from **Table .10** showed that the network with a number of hidden neurons 6 and that has been trained by Quasi – Newton algorithm and with using Hyperbolic Tangent activation function for hidden and output layer gave good performance indication. The final NN model for estimating curing time of adhesive GRE pipe is shown in **Table.14**. **Fig. 15** shows neural network architecture of this model.

Table .14 Neural network model parameters for estimating curing time of adhesive GRE pipe

Parameters	Value
Number of hidden layer	1
Number of input neurons	1
Number of output neurons	1
Number of hidden layer neurons	6
Hidden layer activation function	Hyperbolic Tangent
Output layer activation function	Hyperbolic Tangent
Training algorithm	Quasi – Newton
Iterations	2000

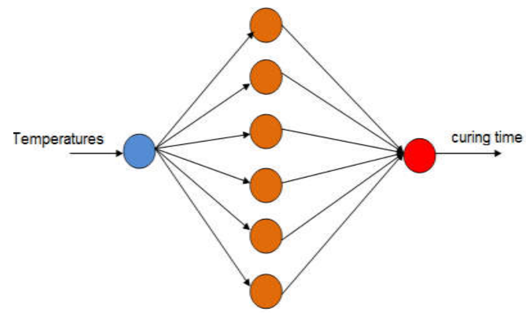

Fig. 15. Neural network architecture for estimating curing time of adhesive GRE pipe

The model of neural network that was proposed to estimate curing time of adhesive GRE pipe illustrated in **Table .14** gave results of network output that are nearly closer to the targets, with correlation (0.9999) and R-Squared (0.9958). **Table .15** shows the values of the target and the network output, absolute error (AE), as well as the absolute relative error (ARE) between the two for each pattern, while **Table .16** illustrates the magnitude of mean, maximum, and minimum of the targets, output, absolute error, and absolute relative error, **Fig. 16** shows curing time of adhesive GRE pipe using the proposed neural network model versus the target, while **Fig. 17** shows the scatter plot of target and network output.

Table .15 The target and network response of curing time of adhesive GRE pipe

W.P No.	Target	Output	AE	ARE
1	24	22.96222	1.03778	4.324081
2	16	17.34668	1.346681	8.416759
3	11	12.63562	1.635624	14.86931
4	9	7.811589	1.188411	13.20457
5	4.5	5.017939	0.517939	11.50976
6	4	3.410116	0.589884	14.7471
7	3.5	2.855909	0.644091	18.40261
8	3	2.550787	0.449213	14.97376
9	2.5	2.500177	0.000177	0.007097

Table .16 Summery of Table .15

	Target	Output	AE	ARE
Mean	8.611111	8.565672	0.823311	0.111617
Min.	2.5	2.500177	0.000177	0.000071
Max.	24	22.96222	1.635624	18.40261

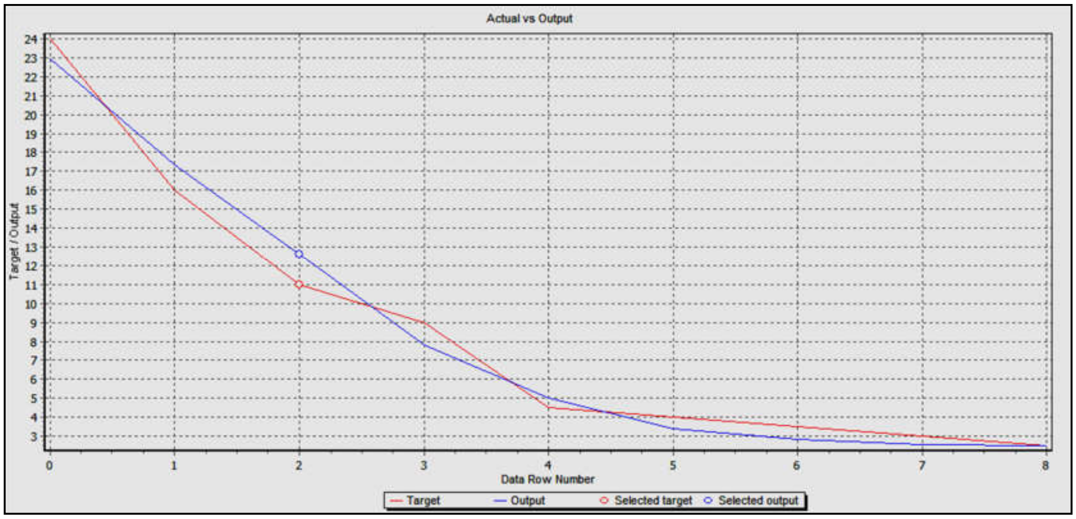

Fig. 16. Curing time of adhesive GRE pipe using the proposed neural network model vs. the target

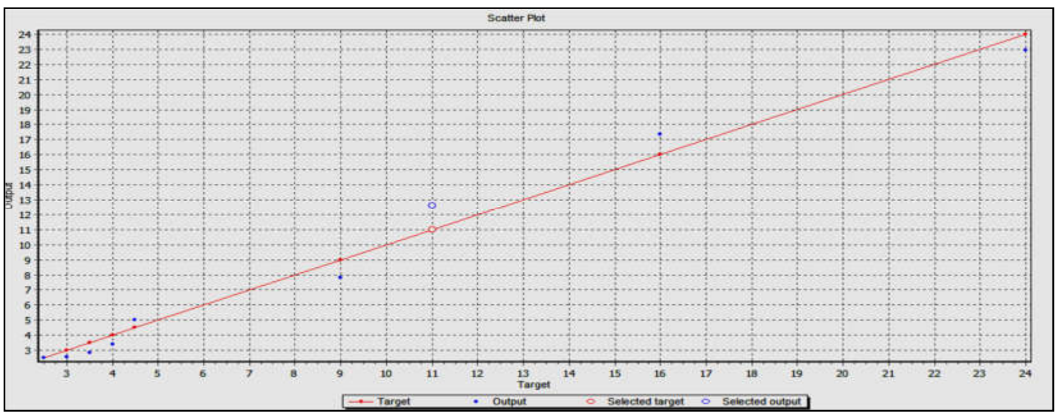

Fig. 17. The scatter plot of target and network output of curing time of adhesive

6. Conclusions

A neural network model with one hidden layer and five hidden layers neurons trained by conjugate gradient descent algorithm by using Logistic activation function showed good performance results for estimating future thickness. This model showed best results between the targets and the network outputs (network response), the correlation coefficient of this model is (0.99999) and R-Squared is (0.9967) that make the network reliable for new operating conditions. A second neural network model with one hidden layer and six hidden layers neurons trained by Quasi – Newton algorithm by using Hyperbolic Tangent activation function showed a good performance results for estimating curing time of adhesive GRE pipe. The correlation coefficient is (0.9999) and R-squared is (0.9958).

References

[1] Alyuda NeuroIntelligenc: AlyudaNeuroIntelligenc Version 2.2 (577) Document, www.Alyuda.com, Copyright 2001-2005.

[2] Andrej Krenker1, Janez Bešter2 and Andrej Kos, "Introduction to the Artificial Neural Networks", Faculty of Electrical Engineering, University of Ljubljana, Slovenia, 2011.

[3] Daniel Svozil, Vladimir KvasniEka , and JiE Pospichal, "Introduction to multi-layer feed-forward neural networks", Chemometrics and Intelligent Laboratory Systems, Vol. 39 ,pp. 43-62, 1997.

[4] Gabriella Bolzon, TaoufikBoukharouba, Giovanna Gabetta, MimounElboujdaini, and MekkiMellas , "Integrity of Pipelines Transporting Hydrocarbons: Corrosion, Mechanisms, Control, and Management", Springer Netherlands, pp. 322, 2011.

[5] Hasan Abdulsahib Mezaal,"Effect of Sintering Time on the Mechanical Properties of Smart Alloy (Cu, Al, Ni)", M.Sc Thesis, Mechanical Department, University of Baghded, 2017.

[6] Hopkins P.,"The Structural Integrity Of Oil And Gas Transmission Pipelines", Penspen Ltd UK, Elsevier Publishers, 2002.

[7] L.A. Dobrzański, and R. Honysz," Application of artificial neural networks in modelling of normalised structural steels mechanical properties", Journal of Achievements in Materials and Manufacturing Engineering, Vol. 32, Issue 1, pp. 37 – 45, January 2009.

[8] Monish.H.S, and Dr. Ashwini kodipalli, "A Study on Expert System and Applications in Education Field", International Journal of Innovative Research in Computer and Communication Engineering, Vol.5, Special Issue 5, pp. 40 – 44,June 2017.

تخمين السمك المستقبلي لأنابيب الصلب الكاربوني وزمن معالجة مادة اللصق في الأنبوب المقوى بالألياف الزجاجية باستخدام نموذج الشبكة العصبية

مصطفى محمد علي منصور

أ.د قاسم محمد دوس
قسم الهندسة الميكانيكية
جامعة بغداد/ العراق

الخلاصة:

الهدف الرئيسي لهذا البحث هو تخمين السمك المستقبلي لأنابيب الصلب الكاربوني وزمن معالجة مادة اللصق في الأنبوب المقوى بالألياف الزجاجية بأستخدام نموذج الشبكة العصبية الصناعية. تم استخدام برنامج Alyuda NeuroIntelligence للحصول على هذه النماذج. هذه النماذج تعتمد على شبكة عصبية متعددة الطبقات ذات التغذية الى الامام ومن خلال تطبيق تجربتين لكل حالة تم التوصل الى افضل الشبكات العصبية التي يمكن استخدامها من اجل تخمين هذه الحالات. النتائج اظهرت بأن الشبكة العصبية بخمسة عصبونات مخفية والتي تم تدريبها بطريقة الـ conjugate gradient descent وباستخدام logistic activation function للطبقة الخارجية والمخفية اعطت اداء جيد لتخمين السمك المستقبلي وقد اعطت هذه الشبكة مخرجات قريبة جدا للاهداف حيث كان correlation (0.9999) و R-Squared (0.9967) بينما الشبكة العصبية بستة عصبونات مخفية والتي تم تدريبها بطريقة الـ Quasi – Newton وباستخدام Hyperbolic Tangent activation function للطبقة الخارجية والمخفية اعطت اداء جيد لتخمين زمن المعالجة وقد اعطت هذه الشبكة مخرجات قريبة جدا للاهداف حيث كان correlation (0.9999) و R-Squared (0.9958) .