

**A COMPARISON BETWEEN THE BACK-PROPAGATION (BPN) AND
GENERAL-REGRESSION NETWORKS (GRNN) IN THE MODELING OF THE
SPOT WELDING PROCESS**

Uğur EŞME and İrem ERSÖZ

M.U., Faculty of Tarsus Technical Education, Tarsus/Türkiye

Arif ÖZBEK

Ç.U., Department of Mechanical Engineering, Adana/Türkiye

Funda KAHRAMAN and Aysun SAĞBAŞ

M.U., Faculty of Tarsus Technical Education, Tarsus/Türkiye

ABSTRACT : *The use of neural networks techniques to model spot welding is explored in this paper. Both the back-propagation (BPN) and General Regression Neural Networks (GRNN) are used to associate the welding process parameters with the features of the tensile shear strength. A comparison between the back-propagation and general regression neural networks in the modeling of the spot welding process is given. It is shown that both the back-propagation and general regression neural networks can model the spot welding process with reasonable accuracy. However, general regression neural network has better learning ability for the spot welding process than the back-propagation network, although the back-propagation network has better generalization ability for the spot welding process than does the general regression neural network.*

**DİRENÇ KAYNAĞININ MODELLENMESİNDE GERİ YAYILMALI VE GENEL
REGRESYONLU SİNİR AĞLARININ KARŞILAŞTIRILMASI**

ÖZET : *Bu çalışmada direnç kaynağının modellenmesinde geri yayılmalı (BPN) ve genel regresyonlu (GRNN) yapay sinir ağları metodu kullanılmıştır. Çekme mukavemeti her iki modelleme tekniği ile modellenerek sonuçlar birbiri ile kıyaslanmıştır. Bu çalışma ile direnç kaynağının modellenmesinde her iki modelleme yönteminin kullanılabileceği kanıtlanmıştır. Elde edilen sonuçlar, genel regresyonlu sinir ağları metodunun bu yöntemde daha kesin ve düşük hata ile sonuç verdiğini göstermiştir. Bu çalışma aynı zamanda, geri yayılmalı yapay sinir ağlarının direnç kaynağın da parametre genelleştirmesinde geri yayılmalı yapay sinir ağlarının genel regresyonlu modellemeden daha etkin olduğu nu göstermiştir.*

1. INTRODUCTION

In recent years, neural networks have become a very useful tool in the modeling of the input–output relationships of complicated systems [1-2]. This is because neural networks have an excellent ability to learn and to generalize (interpolate) the complicated relationships between input and output variables. Several applications of neural networks to the welding process have been reported in the literature [3–5]. In this paper, the use of neural network techniques to model the Spot welding process is studied. In general, the back-propagation network (BPN) seems to be the most utilized neural network [1,6]. This is primarily because the back-propagation network is a multi-layer neural network. The multi-layer network has a greater representational power for dealing with highly non-linear, strongly coupled, multi-variable systems. In this paper, both the back-propagation and general regression networks are used to model the spot welding process. In what follows, the back-propagation and general regression networks are described first. The use of the back propagation and general regression networks in the modeling of the spot welding process is studied.

2. NEURAL NETWORKS

It is well known that modeling the relationships between the input and output variables for non-linear, coupled, multi-variable systems is very difficult. In recent years, neural networks have demonstrated great potential in the modeling of the input–output relationships of complicated systems [1-2]. Consider that $X=\{x_1, x_2, \dots, x_m\}$ is the input vector of the system where m is the number of input variables and $Y=\{y_1, y_2, \dots, y_n\}$ is the corresponding output vector of the system where n is the number of output variables as shown in Figure 1. In this section, the use of back-propagation and general regression networks to construct the relationships between the input vector X and output vector Y of the system will be explored.

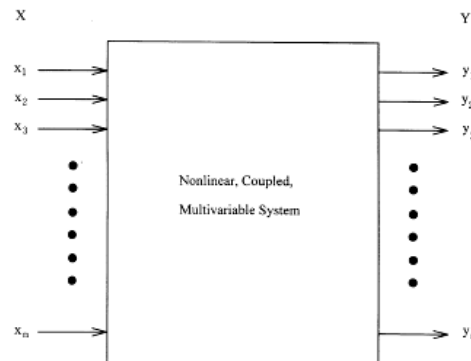


Figure 1. A non-linear, coupled, and multi-variable system [1]

2.1. Back-Propagation Networks (BPN)

The back-propagation network is composed of many interconnected neurons that are often grouped into input, hidden and output layers. The neurons of the input layer are used to receive the input vector X of the system and the neurons of the output layer are used to generate the corresponding output vector Y of the system. The back-propagation network used in this study is shown in Figure 2. For each neuron (shown in Figure 2), a summation function for all the weighted inputs is calculated as:

$$\text{net}_j^k = \sum_j w_{ji}^k o_i^{k-1} \quad (1)$$

where net_j^k is the summation function for all the inputs of the j -th neuron in the k -th layer, w_{ji}^k is the weight from the i -th neuron to the j -th neuron and o_i^{k-1} is the output of the i -th neuron in the $(k-1)$ -th layer.

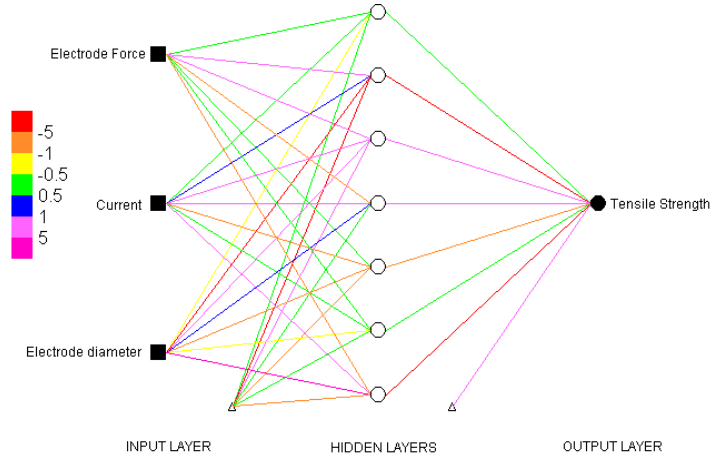


Figure 2. Back Propagation Neural network used for predicting tensile strength

As shown in Eq. (1), the neuron evaluates the inputs and determines the strength of each one through its weighting factor, i.e. the larger the weight between two neurons, the stronger is the influence of the connection [1]. The result of the summation function can be treated as an input to an activation function from which the output of the neuron is determined. The output of the neuron is then transmitted along the weighted outgoing connections to serve as an input to subsequent neurons. To modify the connection weights properly, a supervised learning algorithm [6] involving two phases is employed. The first is the forward phase which occurs when an input vector X is presented and propagated forward through the network to compute an output for each neuron. Hence, an error

between the desired output y_j and actual output o_j of the neural network is computed. The summation of the square of the error E can be expressed as:

$$E = \frac{1}{2} \sum_{j=1}^n (y_j - o_j)^2 \quad (2)$$

The second is the backward phase which is an iterative error reduction performed in a backward direction. To minimize the error between the desired and actual outputs of the neural network as rapidly as possible, the gradient descent method, adding a momentum term [6], is used. The new incremental change of weight $\Delta w_{ji}^k(n+1)$ can be expressed as:

$$\Delta w_{ji}^k(n+1) = -\eta \frac{\partial E}{\partial w_{ji}^k} + \alpha \Delta w_{ji}^k(n) \quad (3)$$

where η is the learning rate, α is the momentum coefficient and n is the index of iteration. Through this learning process, the network memorizes the relationships between input vector X and output vector Y of the system through the connection weights.

2.2. General Regression Neural Networks (GRNN)

The GRNN introduced by Donald Specht in 1990 is a memory-based feed forward neural network based on the approximate estimation of the probability density function (pdf) from observed samples using Parzen-window estimation [7]. It approximates any arbitrary function between input and output vectors. This approach removes the necessity to specify a functional form of estimation.

The method utilizes a probabilistic model between an independent random vector X (input) and a dependent scalar random variable Y (output). Let x and y be the particular measured values of X and Y , respectively, and $g(X, Y)$ be the joint continuous probability density function of X and Y . A good choice for a non-parametric estimate of the probability density function g is the Parzen window estimator as proposed by Parzen and performed for multidimensional cases by Cacoullos [8]. Given a sample of n real D dimensional x_i vectors and corresponding scalar y_i values, the estimate of joint probability density in GRNN is given by;

$$\hat{g}(x, y) = \frac{1}{(2\pi)^{(d+1)/2} \sigma^{d+1}} \frac{1}{n} \sum_{i=1}^n \left[\exp\left(-\frac{(x-x_i)^T(x-x_i)}{2\sigma^2}\right) \exp\left(-\frac{(y-y_i)^2}{2\sigma^2}\right) \right] \quad (4)$$

where σ is the window width of a sample probability, called the smoothing factor of the kernel [9]. The expected value of Y given x (the regression of Y on x) is given by;

A COMPARISON BETWEEN THE BACK-PROPAGATION (BPN) AND GENERAL-REGRESSION
NETWORKS (GRNN) IN THE MODELING OF THE SPOT WELDING PROCESS

$$E[Y/x] = \frac{\int_{-\infty}^{+\infty} Y \cdot g(x, Y) dY}{\int_{-\infty}^{+\infty} g(x, Y) dY} \quad (5)$$

Using Eq. (4), Eq. (5) becomes;

$$\hat{y}(x) = E[Y/x] = \frac{\sum_{i=1}^n [y_i \exp(d_i)]}{\sum_{i=1}^n \exp(d_i)} \quad (6)$$

where d_i is the distance between the input vector and the i^{th} training vector, and is given by;

$$d_i^2 = -\frac{(x - x_i)^T (x - x_i)}{2\sigma^2} \quad (7)$$

The estimate $\hat{y}(x)$ is thus a weighted average of all the observed y_i values where each weight is exponentially proportional to its Euclidean distance from x .

As shown in Figure 3, the structure of the GRNN consists of 4 layers; the input layer, the hidden (pattern) layer, the summation layer and the output layer. As a preprocessing step, all input variables of the training data are scaled. Then, they are copied as the weights into the pattern units.

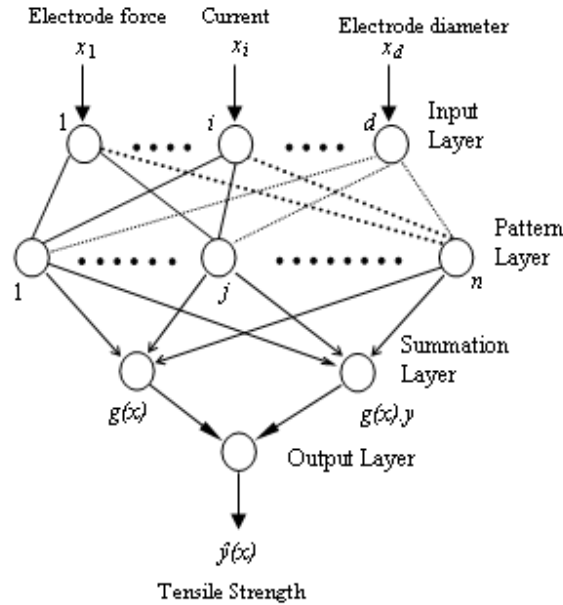


Figure 3. Constructed GRNN network

As a preprocessing step, all input variables of the training data are scaled. Then, they are copied as the weights into the pattern units. The summation layer has two units that can be denoted as the numerator and the denominator of eq. (6). The output layer gives the estimate of the expected value of $\hat{y}(x)$. If y and $\hat{y}$ are the vector variables, the results above are generalized by adding with one summation unit for each component of $\hat{y}$ in the output layer.

The only adjustable parameter of the network is σ , the smoothing factor for the kernel function. It is critical to decide an optimum value for σ . The larger values of this factor cause the density to be smooth, and $\hat{y}(x)$ then converges to the sample mean of the observed y_i . On the other hand, when σ is chosen very small, the density is forced to have non-gaussian shapes. Then, the oscillatory points have a bad effect on the estimate. All values of y_i are taken into account where the points closer to x are given heavier weights, if the optimum value of σ is selected [10].

The main advantage of GRNN according to other techniques is fast learning. It is a one-pass training algorithm, and does not require an iterative process. The training time is just the loading time of the training matrix. Also, it can handle both linear and non-linear data because the regression surface is instantly described everywhere, even just one sample is enough for this. Thus, other existing pattern nodes tolerate faulty samples. Another advantage is the fact that adding new samples to the training set does not require recalibration of the model. As the sample size increases, the estimate surface converges to the optimal regression surface. Thus, it requires many training samples to span the variation in the data and all these to be stored for the future use. Solely, this causes a trouble of an increase in the amount of the computation to evaluate new points. In the course of time, highly improvements in the speed of the computer's processing power prevent this being a major problem. Furthermore, this also can be overcome by applying the various clustering techniques for grouping samples that each center is represented by this group of samples [10]. However, there is only one disadvantage that there is no intuitive method for choosing the optimal smoothing factor.

3. EXPERIMENTAL RESULTS

3.1. Modeling of Spot Welding Process Using Neural Networks

Modeling of the spot welding process with neural networks is composed of two phases: training and testing of the neural networks with experimental welding data. To properly train and test the neural networks, experimental welding data listed in Table 1 is normalized in the range between 0 and 1. The convergence criterion for the developed networks is determined by the average root-mean square (R.M.S.) error E_{RMS} between the desired output value y_j and predicted output value, y'_j i.e.:

A COMPARISON BETWEEN THE BACK-PROPAGATION (BPN) AND GENERAL-REGRESSION NETWORKS (GRNN) IN THE MODELING OF THE SPOT WELDING PROCESS

$$E_{\text{RMS}} = \frac{1}{p} \sum_{i=1}^p \sqrt{\frac{1}{n} \sum_{j=1}^n (y_{ji} - y'_{ji})^2} \quad (8)$$

where p is the number of training or testing data and n is the number of variables in the output vector Y . As given in Table 2, the error sums for BPN and GRNN were found as 5.07 with error average of 0.140 for back propagation network and 2.17 with error average of 0.06 for general regression neural network models. Also, another comparison parameter for both modeling technique is CPU time. The CPU time or elapsed times for BPN and GRNN networks were found to be as 1.50 min and 0.40 sec respectively. As shown in Figure 4, close agreement with very small sum of errors were obtained between actual and predicted results obtained from back propagation and general regression networks.

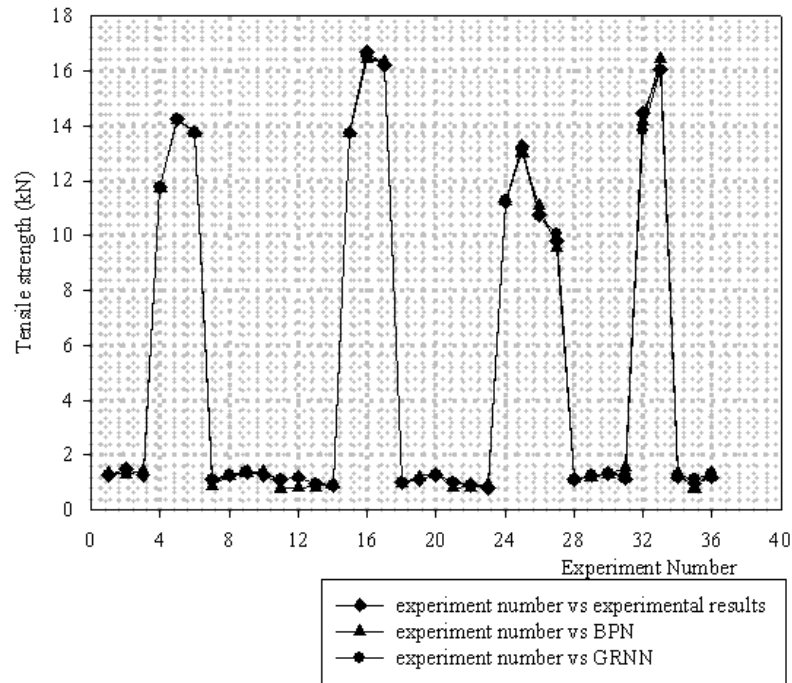


Figure 4. Graphical representation of comparison of the predicted and actual results

Table 1. Experimental results used for testing and train the neural network models

Experiment No	Electrode force F (kN)	Welding current I (A)	Electrode diameter d (mm)	Tensile Strength (kN)
1	4.9	14.40	9.5	1.30
2	4.9	16.20	9.5	1.50
3	4.9	17.00	9.5	1.30
4	4.9	14.40	11	11.75
5	4.9	16.20	11	14.23
6	4.9	17.50	11	13.75
7	4.9	14.40	13	1.10
8	4.9	16.20	13	1.25
9	4.9	17.00	13	1.40
10	4.9	17.50	13	1.30
11	7.85	14.40	9.5	1.10
12	7.85	16.20	9.5	1.20
13	7.85	17.00	9.5	0.95
14	7.85	17.50	9.5	0.88
15	7.85	14.40	11	13.75
16	7.85	17.00	11	16.70
17	7.85	17.50	11	16.20
18	7.85	14.40	13	1.00
19	7.85	16.20	13	1.10
20	7.85	17.00	13	1.30
21	8.83	16.20	9.5	1.00
22	8.83	17.00	9.5	0.90
23	8.83	17.50	9.5	0.80
24	8.83	14.40	11	11.25
25	8.83	16.20	11	13.25
26	8.83	17.00	11	10.75
27	8.83	17.50	11	9.80
28	8.83	14.40	13	1.10
29	8.83	16.20	13	1.25
30	8.83	17.00	13	1.35
31	4.9	17.50	9.5	1.15
32	4.9	17.00	11	14.46
33	7.85	16.20	11	16.03
34	7.85	17.50	13	1.20
35	8.83	14.40	9.5	0.95
36	8.83	17.50	13	1.20

A COMPARISON BETWEEN THE BACK-PROPAGATION (BPN) AND GENERAL-REGRESSION NETWORKS (GRNN) IN THE MODELING OF THE SPOT WELDING PROCESS

Table 2. Comparison of BPN and GRNN results

	Experiment No	Actual Tensile Strength (kN)	Back Propagation Neural Network Results (BPN)		General Regression Neural Network Results (GRNN)	
			<i>predicted</i>	<i>error</i>	<i>predicted</i>	<i>error</i>
TRAIN DATA	1	1.30	1.37	-0.07	1.30	0.00
	2	1.50	1.31	0.19	1.49	0.01
	3	1.30	1.43	-0.13	1.31	-0.01
	4	11.75	11.75	0.00	11.75	0.00
	5	14.23	14.22	0.01	14.23	0.00
	6	13.75	13.75	0.00	13.75	0.00
	7	1.10	0.89	0.21	1.10	0.00
	8	1.25	1.24	0.01	1.26	-0.01
	9	1.40	1.35	0.05	1.37	0.03
	10	1.30	1.39	-0.09	1.32	-0.02
	11	1.10	0.80	0.30	1.10	0.00
	12	1.20	0.81	0.39	1.19	0.01
	13	0.95	0.84	0.11	0.94	0.01
	14	0.88	0.95	-0.08	0.90	-0.02
	15	13.75	13.73	0.02	13.72	0.03
	16	16.70	16.43	0.27	16.52	0.18
	17	16.20	16.32	-0.12	16.25	-0.05
	18	1.00	0.97	0.03	1.00	0.00
	19	1.10	1.21	-0.11	1.11	-0.01
	20	1.30	1.32	-0.02	1.29	0.01
	21	1.00	0.81	0.19	1.00	0.00
	22	0.90	0.84	0.06	0.88	0.02
	23	0.80	0.92	-0.12	0.82	-0.02
	24	11.25	11.25	0.00	11.28	-0.03
	25	13.25	13.03	0.22	13.13	0.12
	26	10.75	11.09	-0.34	10.69	0.06
	27	9.80	9.58	0.22	10.09	-0.29
	28	1.10	1.13	-0.03	1.10	0.00
	29	1.25	1.21	0.04	1.25	0.00
	30	1.35	1.31	0.04	1.34	0.01
TEST DATA	31	1.15	1.57	-0.42	1.30	-0.15
	32	14.46	14.16	0.30	13.81	0.65
	33	16.03	16.42	-0.39	16.03	0.00
	34	1.20	1.37	-0.17	1.30	-0.10
	35	0.95	0.80	0.15	1.12	-0.17
	36	1.20	1.37	-0.17	1.35	-0.15
			ERROR SUM=5.07		ERROR SUM=2.17	
			Average=0.140		Average=0.06	
			CPU time=1.5 min		CPU time=0.40 sec	

4. CONCLUSIONS

This paper has described a neural network approach and comparison of Back Propagation Networks (BPN) and General Regression Neural Networks (GRNN) networks for the modeling of spot welding process. Both the BPN and GRNN networks were used to construct the complicated relationships between the welding process parameters and the tensile strength. Various network configurations for the modeling of the spot welding process have been studied. The experimental results show that the GRNN network has better learning ability (with error sum of 2.17) for the spot welding process than the BPN (with error sum of 5.07) network. In addition to this, BPN network consumed more CPU time (elapsed time 1.5 min) than the GRNN (elapsed time 0.40 sec). However, the back-propagation network has better generalization ability for the spot welding process than the general regression neural network modeling.

5. REFERENCES

1. Juang, S.C., Tarng, Y.S. and Lii, H.R., "A comparison between the back-propagation counter-propagation networks in the modeling welding process", *Journal of Materials Processing Technology* 75, 54–62, (1998).
2. Freeman, J.A. and Skapura, D.M., "Neural Networks: Algorithms, Application and Programming Techniques", Addison–Wesley, New York, 1991.
3. Andersen, K., Cook, G.E., Karsai, G. and Ramaswamy, K., "Artificial neural networks applied to arc welding process modeling and control", *IEEE Trans. Ind. Appl.* 26 (5) 824–830, (1990).
4. Zeng, X.M., Lucas, J. and Fang, M.T.C., "Use of neural networks for parameter prediction and quality inspection in TIG welding", *Trans. Inst. Meas. Control* 15 (2), 87–95, (1993).
5. Jin, B., Soeda, M. and Oshima, K., "Control of weldpool width and cooling time in TIG welding using a neural network model", *Welding Int.* 10 (8), 614–621, (1996).
6. McClelland, J. and Rumelhart, D., "Parallel Distributed Processing", vol.1, MIT Press, Cambridge, MA, 1986.
7. Specht, D., "A General Regression Neural Network", *IEEE Trans. Neural Networks*, Vol. 2, No 6, pp. 568–576, (1991).
8. Cacoullos, T., "Estimation of a multivariate density", *Annals of Inst. Stat. Math.*, vol. 18, pp. 178–189, (1996).
9. Yeung, D., Chow, C., "Parzen Window Network Intrusion Detectors" *Proceedings of the Sixteenth International Conference on Pattern Recognition*. Quebec City, Canada, Vol.4, pp.385–388, (2002).
10. Ersoz, I., Ibrikci, T., Cakmak, A., Ersoy, O.K., "Secondary Structure Prediction of Hemoglobin by Neural Networks", *ANNIE 2004 Artificial Neural Networks in Engineering*, University of Missouri-Rolla, Missouri, USA, 2004.