



Optimum design for artificial neural networks: an example in a bicycle derailleur system

T.Y. Lin, C.H. Tseng*

Department of Mechanical Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan

Received 1 June 1998; accepted 1 September 1999

Abstract

The integration of neural networks and optimization provides a tool for designing network parameters and improving network performance. In this paper, the Taguchi method and the Design of Experiment (DOE) methodology are used to optimize network parameters. The users have to recognize the application problems and choose a suitable Artificial Neural Network model. Optimization problems can then be defined according to the model. The Taguchi method is first applied to a problem to find out the more important factors, then the DOE methodology is used for further analysis and forecasting. A Learning Vector Quantization example is shown for an application to bicycle derailleur systems. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Neural networks; Optimization; Taguchi method; Design of experiments; Bicycle derailleur systems

1. Introduction

Artificial Neural Networks (ANNs) are receiving much attention currently because of their wide applicability in research, medicine, business, and engineering. ANNs provide better and more reasonable solutions for many problems that either can or cannot be solved by conventional technologies. Especially in engineering applications, ANNs offer improved performance in areas such as pattern recognition, signal processing, control, forecasting, etc.

In the past few years, many ANN models with different strengths have been introduced for various applications. According to the different ANN models used, many training algorithms have been developed to improve the accuracy and convergence of the models. Although a lot of research is being concen-

trated in these two fields, there is still a conventional problem in ANN design. Users have to choose the architecture and determine many of the parameters in a selected network. For instance, in a “Multilayer Feedforward (MLFF) Neural Network”, the architecture, such as the number of layers and the number of units in each layer, has to be determined. If a “Backpropagation with Momentum” training algorithm is selected, many parameters, such as the learning rate, momentum term, weight initialization range, etc., have to be selected. It is not easy for a user to choose a suitable network even if he is an experienced designer. The “trial-and-error” technique is the usual way to get a better combination of network architecture and parameters.

Therefore, there must be an easier and more efficient way to overcome this disadvantage. Especially in engineering applications, an engineer, with or without an ANN background, should not spend so much time in optimizing the network. In recent years, the Taguchi method (Taguchi, 1986; Peace, 1993) has become a new approach that can be used for solving the optimiz-

* Corresponding author. Tel.: +886-35-712-121; fax: +886-35-720-634.

E-mail address: chtseng@cc.nctu.edu.tw (C.H. Tseng).

ation problems in this field. The parameters and architectures of an MLFF network were selected by using the Taguchi method in Khaw et al. (1995). This can improve the original network design to obtain a better performance. The same technique has been used to optimize Neocognitron Networks (Teo and Sim, 1995) and another MLFF network (Lin and Tseng, 1998). The Taguchi Method is a type of optimization technique, which is very well suited to solving problems with continuous, discrete and qualitative design variables. Therefore, any ANN model can be optimized by this method. Another method, the genetic algorithm, which requires a large computational cost, has been applied to populations of descriptions of networks in order to learn the most appropriate architecture (Miller et al., 1989).

In this study, a systematic process is introduced to obtain the optimum design of a neural network. The Taguchi method and the Design of Experiments technique (DOE) (Montgomery, 1991) are the main techniques used. Unlike previous studies, the Taguchi method is used here to simplify the optimization problems. Then, DOE is more easily performed. Because of the stronger statistical basis of DOE methodologies, many analyses can be executed. Finally, a Learning Vector Quantization (LVQ) network is demonstrated as an example. The method proposed in this paper can also be applied to any ANN model. The integration of optimization and ANNs in this paper was simulated by a computer program which can be executed automatically and easily.

2. Optimization process

Optimization techniques are used to obtain an improved solution under given circumstances. In ANN design, it is helpful to improve the original settings of a network in order to get a better performance. For the convenience of further analysis, the parameters in ANNs must be classified as follows.

2.1. Design parameter classification

ANNs are defined by a set of quantities, some of which are viewed as variables during the design process. These parameters are classified into three parts according to the numerical quantities. For an n -vector $\underline{x}=(x_1, x_2, \dots, x_n)$, there are

1. Continuous design parameters:

$$x_{kl} \leq x_k \leq x_{ku} \quad k = 1, 2, \dots, n \quad (1)$$

where $x_k \in \mathcal{R}^n$, x_{kl} is the lower bound of x_k , x_{ku} is the upper bound of x_k .

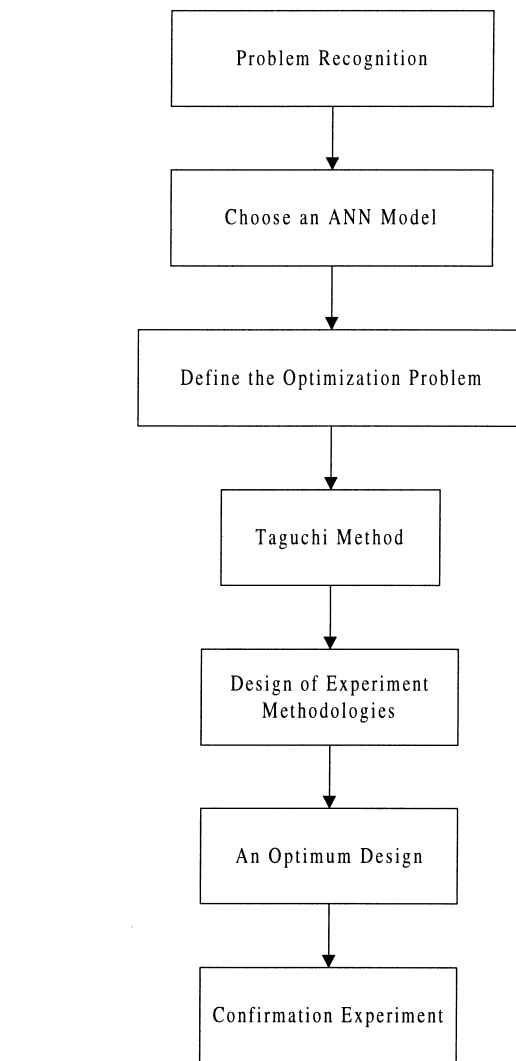


Fig. 1. Optimization process.

2. *Discrete design parameters*: $x_k \in (x_{k1}, x_{k2}, \dots, x_{km})$ and m is the size of the discrete set.
3. *Qualitative design parameter*: x_k is a qualitative variable which cannot be described by a numerical expression.

For example, consider an MLFF neural network with a “backpropagation with momentum” training method. The continuous design parameters are the learning rate, momentum term and weight initialization range. The discrete design parameters are the number of hidden layers, the number of units in each layer and the number of training data items. The qualitative design parameters are the activation function type, the network typologies and the numerical method, such as the gradient descent, conjugate gradient and BFGS (Arora, 1989).

2.2. Optimization problem

In order to obtain an optimum design for a neural network, an optimization process is proposed in Fig. 1. First, choose a suitable ANN model for the application. The optimization problem can be formulated as follows.

Find an n -vector $\underline{x} = (x_1, x_2, \dots, x_n)$ of design variables to minimize a vector objective function

$$F(\underline{x}) = [f_1(\underline{x}), f_2(\underline{x}), \dots, f_q(\underline{x})] \quad (2)$$

subject to the constraints.

$$h_j(\underline{x}) = 0; \quad j = 1, 2, \dots, p$$

$$g_i(\underline{x}) \leq 0; \quad i = 1, 2, \dots, m. \quad (3)$$

The design variables \underline{x} can be classified into three parts: continuous, discrete and qualitative design variables, as defined above. The objective functions represent some criteria that are used to evaluate different designs. In ANN design, the objective function can be the training error, the learning efficiency, the grouping error, etc. For engineering design problems, there are some limitations, called constraints, and design variables are not completely freely selected. Equality as well as inequality constraints often exist in a problem.

2.3. Traditional optimization method

Numerical methods, such as Sequential Linear Programming (SLP) and Sequential Quadratic Programming (SQP) (Arora, 1989), which are employed to solve optimization problems, are usually referred to as “traditional methods”. In ANN design, it is not appropriate to use these schemes to solve problems. The reasons can be stated as follows.

1. There exist qualitative design parameters and these qualitative design parameters cannot be described by a numerical expression. Therefore, they cannot be solved using numerical methods.
2. There exist non-pseudo-discrete design parameters. These discrete parameters, which occur when the solution to a continuous problem is perfectly meaningful but cannot be accepted due to extraneous restrictions, are termed as “pseudo-discrete parameters” which can be solved by traditional methods (Gill et al., 1981). For instance, the variable in a design problem could be the diameter of a pipe. The diameter is a continuous variable, but only specific values, such as 1 in, 1.5 in and 2 in, can be found in the market. This kind of variable is called a “pseudo-discrete” design parameter. Many non-pseudo-discrete parameters that are intrinsically

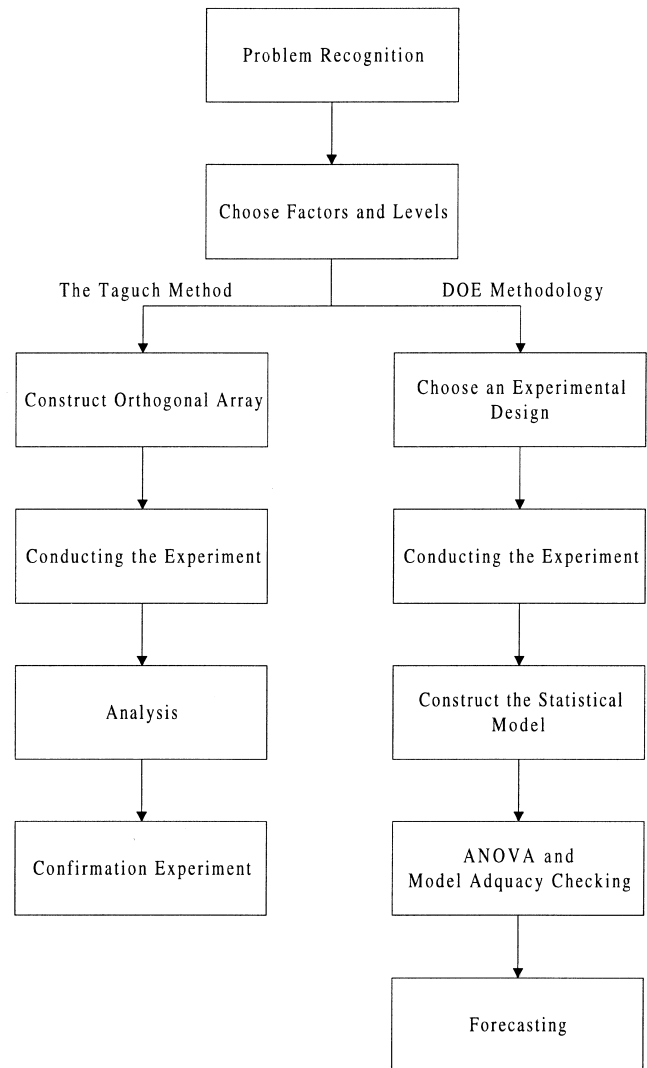


Fig. 2. The Taguchi method and DOE methodology.

discrete, such as the number of units and layers, have to be determined in ANN design.

3. The objective function is complicated. In applying the traditional methods, first order or second order differentials of the objective function have to be checked before using SLP or SQP. However, in ANN design, it is difficult or impossible to write the numerical expressions of the objective function. For example, the grouping error is treated as the objective function, but the grouping error of every training process may be calculated from software or a user subroutine, which is seen as a “black box”. Therefore, only the implicit form of the objective function can be obtained. There is no explicit form of the objective function for checking.

For the above reasons, traditional optimization methods cannot be performed well in ANN design. On the other hand, there are no such limitations when



Fig. 3. A complete derailleur system on a bicycle.

using other kinds of optimization methods, such as the Taguchi method and DOE methodology.

2.4. The Taguchi method

Dr. Genichi Taguchi's methods were developed after World War II in Japan. His most notable contributions lie in quality improvement, but in recent years, the basic concepts of the Taguchi method has been widely applied in solving optimization problems, especially in zero order problems. Because the Taguchi method is a kind of fractional factorial DOE, the simulation or experiment at times can be reduced, compared to DOE. For example, if there are seven two-level factors in a design problem, only eight simulations have to be done in the Taguchi method. In DOE, however there are $2^7 = 128$ simulations that have to be done.

Fig. 2 shows the process of the Taguchi method. The engineers must recognize the application problem well and choose a suitable ANN model. In the selected model, the design parameters (factors) which need to be optimized have to be determined. Using orthogonal arrays, simulations can be executed in a systematic way. From simulation results, the responses can be analyzed by level average analysis and signal-to-noise (S/N) ratio in the Taguchi method (Taguchi, 1986).

2.5. DOE methodology

DOE is a test or series of tests in which the designer may observe and identify the reasons for changes in

the output response from the changes in the input parameters. Fig. 2 also shows the process of DOE. Unlike the Taguchi method, a statistical model is constructed for the simulations and the experiment. Therefore, some assumptions and validations of the model (model adequacy checking) have to be made, both before and after the experiment. The experimental strategy is to change one parameter and keep the rest of the parameters constant in each step. Therefore, the experiment and simulation times are much longer than in the Taguchi method as mentioned before. The experimental response, such as the training error and convergence speed, can be analyzed and forecast by "Analysis of Variance" (ANOVA) and other statistical techniques (Montgomery, 1991).

2.6. The Taguchi method vs. DOE methodology

In the optimization process shown in Fig. 1, the Taguchi method is treated as a pre-running of the design parameters. For some engineering applications, it is quite sufficient to use the Taguchi method. There are many reasons to do so. In design problems, there are sometimes a large number of design parameters. It is not efficient to use DOE methodologies at this time because of too many training cases. Therefore, the Taguchi method is used to reduce the training cases, and to find the more important parameters that affect the response of the neural network. Afterwards, the DOE methodology can be easily completed using a smaller number of important parameters, keeping

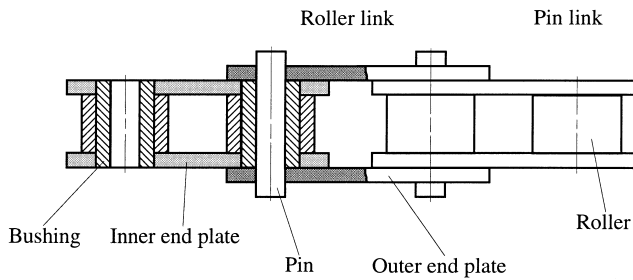


Fig. 4. The construction and nomenclature for roller chain links.

other parameters constant from the conclusions of the Taguchi method.

In the DOE methodology, the experimental matrix contains all the combinations of factors and levels. Therefore, the experimental data are sufficient to construct the statistical models for the analytical phase. Because of the stronger statistical base in DOE methodology, ANOVA can be executed in DOE but it cannot be executed in the Taguchi method. ANOVA provides the sensitivity analysis in DOE, and the characteristics of the parameters can be realized. Also, a forecast can be made to find the optimal combinations of the design parameters. The process of using DOE and the Taguchi method is described in Appendix A.

3. The LVQ example: part 1

In order to demonstrate the optimum design processes, an application with a Learning Vector Quantization (LVQ) model is shown. In this example, the purpose is to distinguish the type of chain engagement to be used in the rear derailleur system of a bicycle.

3.1. Problem description

The derailleur system in a bicycle is similar to the gear box in a motor vehicle. A complete derailleur system, as shown in Fig. 3, consists of five components: chainwheel and freewheel, front and rear derailleurs, shift levers, cables and a chain. Riders change gears by moving shift levers, causing the derailleurs to guide the chain to engage larger sprockets or to drop to smaller sprockets. In derailleur system designs, two types of chain engagement, Type I and Type II, have to be considered (Wang et al., 1996). Fig. 4 shows the construction and nomenclature of a roller chain. Notice that a roller chain consists of two alternating types of link: roller links (or inner links) and pin links (or outer links). Therefore, in a chain drive system, each tooth can be engaged by a roller link or a pin link; thus there are two shifting patterns, Type I and Type II, as

shown in Fig. 5. This can easily be seen from the first engagable tooth and roller in Fig. 5(a) and (b). The chainwheel and freewheel sprocket design of the two types are different. Therefore, it is very important for the designers to know which type occurs during each gear shift so that different design defects for the two types can be found and rectified.

In a real riding or testing environment, it is very difficult to distinguish which type of chain engagement occurs. There must be a camera for monitoring purposes. This costs a lot of money and it is not very easy to install. The purpose of this example is to establish a better and easier method to distinguish the chain engagement type during gear shifts, using a neural network model.

In the experiments, the vibration signals during the shifting of gears can be recorded by computer (Lin and Tseng, 1998) and fed to the neural network. Fig. 6(a) and 6(b) are the typical time-domain signals of Type I and Type II chain engagements. The x -axis of Fig. 6 is the time sequence and the y -axis is the voltage

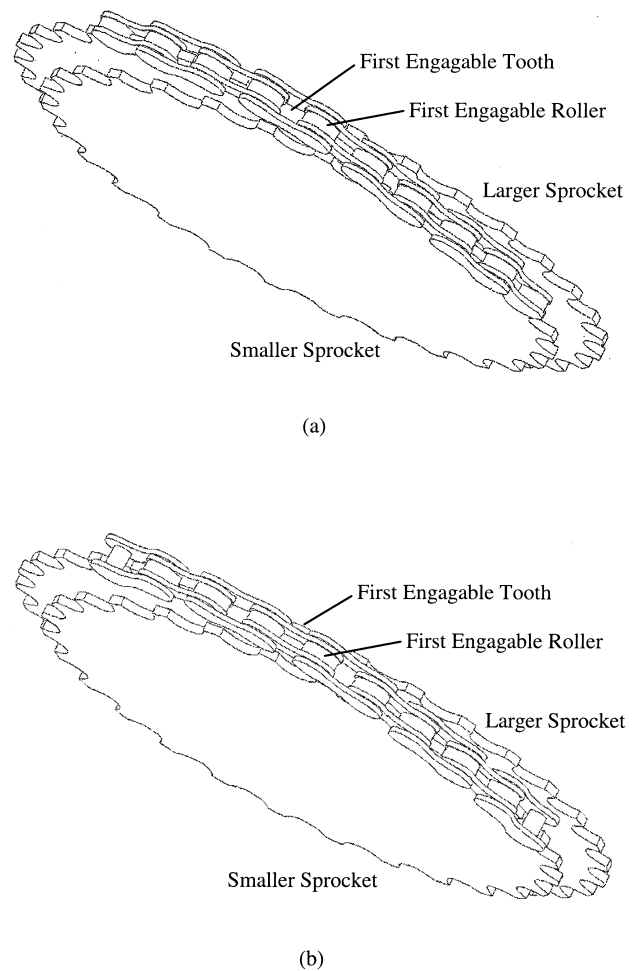


Fig. 5. The solid model: (a) Type I and (b) Type II.



(a)



(b)

Fig. 6. Vibration signals in the time domain: (a) Type I and (b) Type II.

from the accelerometer. The data fed to the network are transferred from the time domain to the frequency domain by the FFT technique (John and Dimitris, 1996). In gathering training data, if the tooth numbers of two adjacent sprockets are both even and the tooth number of the chainwheel sprocket is also even, only one type of chain engagement will occur. If only one chain link is shifted from the previous situation, another type will occur. In this example, 80 items are used to train the network, and 40 items are applied to test the trained network.

3.2. Choosing an ANN model

In supervised learning models, an LVQ example like that shown in Fig. 7 is selected because of its fast training speed, no local minimum traps and better performance in classification (Patterson, 1996). The LVQ is the transformation from the input vector \underline{x} of dimension n to known target output classifications $t(\underline{x}) = t$, where each class is represented by a codeword

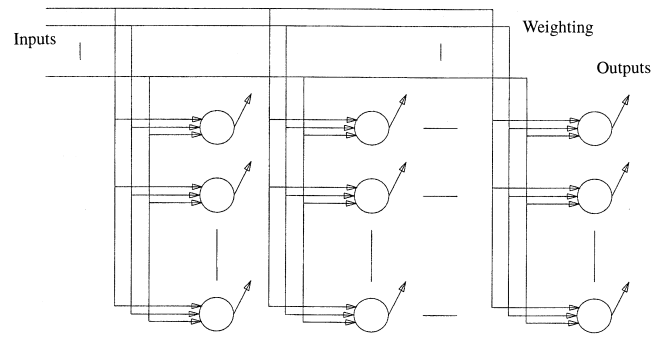


Fig. 7. A Learning Vector Quantization network.

or prototype vector $\underline{w}_i (i = 1, 2, \dots, m)$. The index i is the class label for \underline{x} . Let $C(\underline{x})$ denote the class of \underline{x} computed by the network; \underline{w}_c is the weight vector of the winning unit c . Then, $C(\underline{x})$ is found using

$$\|\underline{w}_c - \underline{x}\| = \min_i \|\underline{w}_i - \underline{x}\|. \quad (4)$$

When the class is correct, i.e. $C(\underline{x}) = t$, the weight vector of the winning unit c is shifted toward the input vector. When an incorrect classification is selected, i.e. $C(\underline{x}) \neq t$, the weight vector is shifted away from the input vector. The update rule for the LVQ can be summarized as follows:

1. Initialize the weights w to small random numbers.
2. Find the prototype unit to represent \underline{x} by computing

$$\|\underline{w}_c - \underline{x}\| = \min_i \|\underline{w}_i - \underline{x}\|. \quad (5)$$

3. Update the weight vectors according to

$$\underline{w}_c^{\text{new}} = \underline{w}_c^{\text{old}} + \alpha^+ (\underline{x} - \underline{w}_c) \quad \text{if } C(\underline{x}) = t$$

$$\underline{w}_c^{\text{new}} = \underline{w}_c^{\text{old}} - \alpha^- (\underline{x} - \underline{w}_c) \quad \text{if } C(\underline{x}) \neq t$$

$$\underline{w}_i^{\text{new}} = \underline{w}_i^{\text{old}} \quad \text{for all } i \neq c \quad (6)$$

where $\alpha^+ > 0$ and $\alpha^- > 0$ are the learning coefficients of two different cases.

4. Repeat steps 2 and 3 until the weights stabilize.

3.3. Define the optimization problem

The optimal physical problem can be covered by a mathematical model of design optimization involving the procedures below.

1. *Choose design variables:* Based on the requirements of the physical problem, users can choose some factors as design variables, which can be varied during the optimization iteration process, and some factors as fixed constants. In this example, the chosen de-

Table 1
Factors and levels

Factor	Level 1	Level 2	Level 3
No. of input unit	256	128	32
η^+	0.05	0.1	0.3
η^-	0.05	0.1	0.3
Weight initial range	± 0.1	± 0.3	± 0.5

sign variables from the LVQ network parameters are the number of input units, α^+ and α^- in Eq. (6), and the weight initialization range. The number of input units is a discrete design parameter, α^+ and α^- are continuous design parameters, and the weight initialization range is a qualitative design parameter.

2. *Define an objective function:* The objective function must be defined according to the purpose and requirements of the problem. The objective function in this example is defined as the grouping error of the network,

$$\text{cost function} = \sum_i f_{\text{diff}}(C_i(\underline{x}), t_i), \quad (7)$$

where

$$\begin{cases} f_{\text{diff}} = 0 & \text{if } C_i(\underline{x}) = t_i \\ f_{\text{diff}} = 1 & \text{if } C_i(\underline{x}) \neq t_i \end{cases} \quad (8)$$

and $i = 1$ to the size of the training data. The output of interest of this example is a “smaller-the-better” quality characteristic.

3. *Identify constraints:* A suggested range of design variables from the solver SNNS (Zell, 1995) will be described in the next section.

3.4. The Taguchi method

The theories and principles used in the Taguchi Method (Taguchi, 1986; Peace, 1993) will not be

described in detail; only the key points and analyzed results are shown below. From the optimization model, four factors, including the number of input units, α^+ , α^- and the weight initialization range, have to be determined. The factors and selected levels are shown in Table 1.

1. For the number of input units, the vibration signals during gear-shifting are transformed into the frequency domain by the FFT to 256 data points. Using data compression techniques, 256 points can be compressed to 128 and 32 points. Therefore, in this factor (the number of input units), three levels (256, 128 and 32 points) are selected.
2. For α^+ and α^- , the two parameters are continuous design variables and need to be transferred to discrete ones. In optimization formulation, there must be lower and upper bounds for the design variables. According to the suggestion of the software (Zell, 1995), 0.1 is suitable for general purposes. Therefore, $0.05 \leq \alpha^+, \alpha^- \leq 0.3$ is assumed, and three levels (0.05, 0.1 and 0.3) are selected. If the optimization results are located at the boundaries, an expansion has to be made in the following DOE process. Otherwise, if the results are acceptable for an engineering application, no expansion is needed for the boundaries.
3. For the weight initialization range, a small range is suggested by the software (Zell, 1995). Therefore, ± 0.1 , ± 0.3 and ± 0.5 are selected.

After the factors and levels are determined, a suitable orthogonal array can be selected for the training process. Table 2 is the $L_9(3^4)$ orthogonal array for the factors and levels in this example. For instance, in the first training experiment, there are 256 input units, η^+ and η^- are set to 0.05, and the weight initialization range is between +0.1 and -0.1. After nine training experiments have been made, the grouping errors of the 80 training data are summarized in Table 2. Since there are no local minimum traps in this model, replicate training is not needed for the same parameters. For some other models, the final results may be

Table 2
Training results

	Input units	η^+	η^-	Weight initial range	Grouping error	S/N (dB)
1	256	0.05	0.05	± 0.1	1/80	19.03
2	256	0.1	0.1	± 0.3	14/80	7.56
3	256	0.3	0.3	± 0.5	34/80	3.72
4	128	0.05	0.1	± 0.5	18/80	6.48
5	128	0.1	0.3	± 0.1	34/80	3.72
6	128	0.3	0.05	± 0.3	5/80	12.04
7	32	0.05	0.3	± 0.3	35/80	3.59
8	32	0.1	0.05	± 0.5	43/80	2.70
9	32	0.3	0.1	± 0.1	33/80	3.85

Table 3
Response table

Factor	Level	Error	Factor	Level	Error
Input units	256	16	η^-	0.05	16
	128	19		0.1	22
	32	37		0.3	34
η^+	0.05	18	Weight initial range	± 0.1	23
	0.1	30		± 0.3	18
	0.3	34		± 0.5	32

affected by different initial designs. Therefore, replicated training is necessary for the following S/N analysis.

The last column in Table 2 is the signal-to-noise ratio (S/N). The equation for calculating the S/N for the smaller-the-better quality characteristic is Eq. (A1). In this example, there is only one replicate, therefore, the physical meaning of S/N is similar to the grouping errors in Table 2. The grouping errors are used here instead of S/N for easier understanding.

The next step in the Taguchi method is Level Average Analysis. The goal is to identify the strongest effects and to determine the combination of factors and levels that can produce the most desired results. Table 3 is the response table, which shows the average experiment result for each factor level. The total effect of the 256 input units is 16. This is the average grouping error of the first three rows in Table 2 $((1 + 14 + 80)/3 = 16)$. Other response values can be calculated by using a similar method. For the number of input units, 256 units can get a smaller grouping error than other levels. The same principle can be used to make η^+ and η^- equal to 0.05, and the weight initialization range to be between $+0.3$ and -0.3 . Fig. 8 shows the response curves for the four factors. It shows that the four factors do have a strong effect on the grouping errors. Therefore, the recommended factor levels are: 256 input units, $\eta^+ = 0.05$, $\eta^- = 0.05$ and a weight initialization range of ± 0.3 .

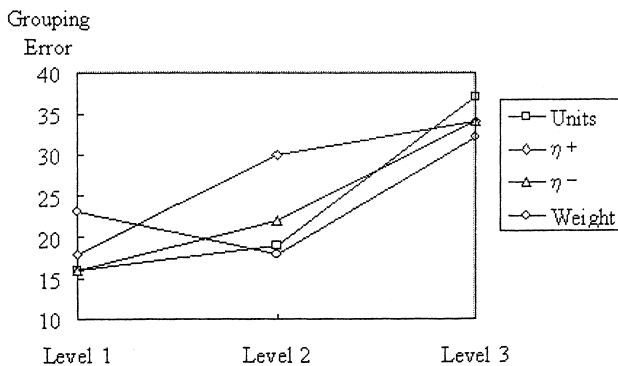


Fig. 8. Response curves.

4. The LVQ example: part 2

From the Taguchi method, an improved design of the LVQ network is obtained. Fig. 2 shows the next step of the optimization process — the DOE methodology. The main purpose of this step is to further analyse the results of the Taguchi method, and to get more accurate settings for the factors. The DOE theories and principles (Montgomery, 1991) will not be described here; only the key points and results are shown below.

4.1. Choosing an experimental design

For the number of input units, it is obvious that a larger number will cause smaller grouping errors in Fig. 5, and that 256 units is the maximum. Therefore, the number of input units will remain at 256 in this step. For the weight initialization range ± 0.3 is not located at the variable boundaries and seems to be a local minimum in Fig. 8. Therefore, this parameter will also remain constant here. For η^+ and η^- , the minimum response is located at the lower bound of the variables; therefore, an expansion of the boundaries is required in further analysis. In summary, only the two continuous parameters, η^+ and η^- , are treated as design variables in the DOE methodology, and the other two parameters are kept constant.

The new parameter sets are shown in Table 4. The upper and lower bounds of η^+ and η^- are changed from 0.05 and 0.3 to 0.01 and 0.1. Therefore, three levels (0.1, 0.05 and 0.01) are designed for training, while the other two parameters are kept constant. Nine training cases have to be simulated in this example.

4.2. Conducting the experiment

The same 80 items of training data used in the Taguchi method are also applied here. The results of grouping errors are also shown in Table 4. Upon completion of the training experiments, DOE analysis techniques can be executed.

Table 4
DOE array and training results

		η^+		
		0.1	0.05	0.01
η^-	0.1	14/80 (4.0)	26/80 (7.0)	28/80 (8.0)
	0.05	23/80 (6.0)	1/80 (1.5)	10/80 (3.0)
	0.01	29/80 (9.0)	19/80 (5.0)	1/80 (1.5)

4.3. The statistical model

This is a two-factor factorial design. The statistical model of this example is

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk}, \quad \begin{cases} i = 1, 2, \dots, a \\ j = 1, 2, \dots, b \\ k = 1, 2, \dots, n \end{cases} \quad (9)$$

where

- y_{ijk} is the ijk th observation (grouping error),
- μ is the overall mean,
- τ_i is the i th η^+ effect (fixed effect),
- β_j is the j th η^- effect (fixed effect),
- $(\tau\beta)_{ij}$ is the interaction between η^+ and η^- ,
- ε_{ijk} is the random error component. $\varepsilon_{ijk} \sim NID(0, \sigma^2)$.

There are three levels in each factor and only one replicate is done; therefore, $a = 3$, $b = 3$ and $k = 1$. With only one replicate, there are no error estimations. One approach applied to the following analysis is to assume a negligible higher order interaction between η^+ and η^- combined with an error degree of freedom.

4.4. Analysis of variance (ANOVA)

Using the statistical model and the training results, the ANOVA technique can be executed. Table 5 is the ANOVA table from the SAS software, and the error estimation is taken from the interaction between η^+ and η^- . If a 95 percent confidence interval is assumed (i.e., $\alpha = 0.05$, the normal setting for applications), $F_{\alpha, v_1, v_2} = F_{0.05, 2, 2} = 19.0$. For the null hypotheses of $\tau_i = 0$ and $\beta_j = 0$,

$$F_{0, \eta^+} = 0.42 < F_{0.05, 2, 2} = 19.0 \quad \text{and}$$

$$F_{0, \eta^-} = 0.61 < F_{0.05, 2, 2} = 19.0.$$

Therefore, the null hypotheses are accepted. The conclusion is that there is no significant difference between the three levels of η^+ and η^- . The results can also be observed from the Pr value in the ANOVA table. $Pr = 0.68$ and $Pr = 0.59$ mean that the probability of rejecting null hypotheses is very high (compared to 0.05). The sensitivity of η^+ and η^- to grouping errors

Table 5
The ANOVA table

Source	Sum square	Degree of freedom	Mean square	F_0	$Pr > F$
η^+	141.56	2	70.78	0.42	0.6827
η^-	205.56	2	102.78	0.61	0.5868
Error	673.11	4	168.27		
Total	1020.23	8			

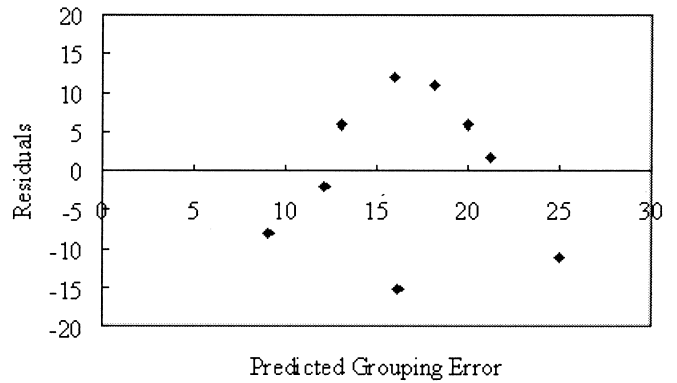


Fig. 9. Plot of residuals vs. predicted grouping error.

is not very high. In some cases, the conclusion may be drawn that there are significant differences between the levels. This means that the factor is very sensitive to the output response.

4.5. Model adequacy checking

In the statistical model of this problem, three assumptions are made: normality, independence and equal variance. Some tests have to be executed to verify these assumptions. For checking the normality, the Kruskal–Wallis Test (Montgomery, 1991) is used. In Table 4, the values in the brackets are the data ranks, R_{ijk} , for the experiment.

$$S^2 = \frac{1}{N-1} \left[\sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n R_{ijk}^2 - \frac{N(N+1)^2}{4} \right] = \frac{1}{8} \left[284.5 - \frac{9 \times 10^2}{4} \right] = 7.4375 \quad (10)$$

$$H = \frac{1}{S^2} \left[\sum_{i=1}^a \frac{R_{ijk}^2}{n} - \frac{N(N+1)^2}{4} \right] = \frac{1}{55.32} \left[284.5 - \frac{9 \times 10^2}{4} \right] = 1.0756. \quad (11)$$

Since $H < \chi_{0.05, 8}^2 = 15.51$, one would accept the null hypothesis of $\tau_i = 0$ and $\beta_j = 0$. There is no significant difference between the three levels of η^+ and η^- . The conclusion here is the same as that given by the usual analysis of the variance F test. Therefore, the normality assumption is justified.

On the other hand, because this example is a single replicated factorial, a regression method is applied (Montgomery, 1991) for a residual plot. The linear regression model is

$$\hat{y} = 7.276867 + 101.366120\eta^+ + 76.775956\eta^-. \quad (12)$$

Fig. 9 plots the residuals vs. the predicted values, \hat{y} , for the grouping errors. There is no obvious pattern apparent, therefore, the independence and equal variance assumptions are justified.

4.6. Forecasting

In Table 4, the training results show that two cases of η^+ and η^- using the combination ((0.01, 0.01) and (0.05, 0.05)) will get a better grouping error. Only one grouping error occurs among 80 training data. In order to get more precise results, the regression method is used. Using the General Regression Model in the SAS software (Montgomery, 1991), the highest order Fitting Response Surface is

$$\begin{aligned} z = & -13.89 + 1284.54\eta^+ + 800.09\eta^- \\ & - 8422.84\eta^{+2} - 3867.28\eta^{-2} - 68236\eta^+\eta^- \\ & + 561574\eta^+\eta^{-2} + 589352\eta^{+2}\eta^- \\ & - 5262346\eta^{+2}\eta^{-2}. \end{aligned} \quad (13)$$

The surface is shown in Fig. 10. Using the partial differential method, the minimum z value and the correspondence η^+ and η^- can be obtained: $\eta^+ = 0.046$ and $\eta^- = 0.05$.

4.7. Confirmation experiment

Using the recommended factor levels:

number of input units: 256,

η^+ :0.046,

η^- :0.05,

weight initialization range: ± 0.3 .

the grouping error after training becomes zero, i.e., all the training data are classified successfully.

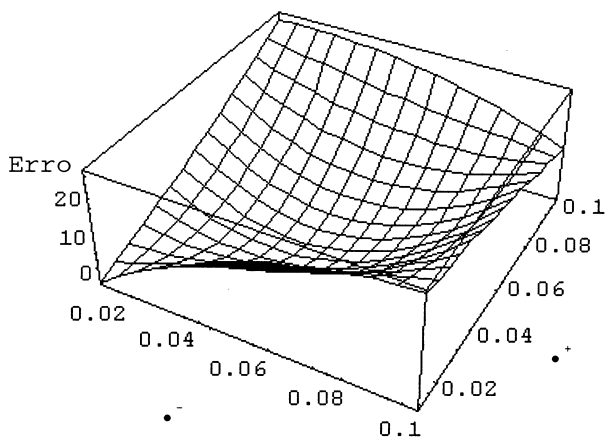


Fig. 10. Fitting response surface.

5. Conclusion

Optimization techniques have been widely used in many applications. In this paper, two major categories, the Taguchi method and the DOE methodology, are applied to improve upon the original designs of ANNs. The users have to recognize the design problem and choose a suitable ANN model. Then, the optimization problems can be defined according to the model. The Taguchi method is first applied to find the more important factors, and to simplify the design problems. DOE methodologies are then used to find the sensitivity and a more precise combination of design parameters. The final results of the examples introduced in this study indeed improve the initial designs and get a better performance.

Although only one ANN model, LVQ, is demonstrated in this paper, other models, such as ADALINE, MADALINE, Hopfield Networks, MLFF, Boltzmann Machines, Recurrent Neural Networks, Neocognitrons, etc., are also suitable. Many benefits can be mentioned. First, this is a systematic method to use for a neural network design. It means that the engineer, whether or not he or she is experienced in ANN, the Taguchi method and DOE, can follow this process easily. Many commercial software packages can be applied, such as SNNS in ANN and SAS in the DOE. Second, it will not take too much computational effort and time. The results of the demonstrated examples can be obtained within 5 min with a Pentium-150 PC. This detail was not emphasized in this paper because it is not the major concern here. Finally, in engineering applications, it is not necessary to get a global optimization of the problems, because that takes too much time or the algorithms may be very complicated. The improvement of the original designs to an acceptable region is helpful for engineers.

Acknowledgements

The support of this research by the National Science Council, Taiwan, R.O.C., under grant NSC-85-2622-E-007-012, is gratefully acknowledged.

Appendix A. The Taguchi method and DOE methodology

Dr. Genichi Taguchi's methods were developed after World War II in Japan, while the DOE methodology was first introduced by R. A. Fisher in the 1920s. The Taguchi method is a kind of fractional factorial DOE; therefore, the simulation or experiment times can be reduced to a smaller number compared to DOE. The most notable contributions of the methodology are in quality improvement, but in recent years, the basic

concepts of the Taguchi method and DOE have been widely applied to the solution of optimization problems, especially in zero order problems. In this section, some of the basic concepts used in the Taguchi method and DOE are described.

A.1. Problem recognition

A clear recognition of the problem often contributes substantially to a better understanding of the phenomena involved, and the final solution of the design problem. Usually, it is necessary to convert physical statements or customer requirements to measurable quantities. Therefore, an optimization problem which consists of design variables, objective (cost) functions, and constraints can be defined.

A.2. Choice of factors and levels

In the problem recognition stage, the experimenter must choose the design variables to be varied. These variables are named as “factors” in the experiment. Some factors are treated as “noise” factors that are uncontrollable, unimportant, or not of concern during the experiment. The factors other than these are of major concern in the Taguchi method or DOE methodology. In every factor, the experimenter also has to choose some specific values, choose which runs will be made and the range over which these factors will be varied. These values are called the “levels” of every factor. For instance, the Multilayer Feedforward (MLFF) neural network shown in Fig. A1 has to be optimized. The factors can be the number of input units, the number of hidden layers, the number of hidden units for each hidden layer, the learning rate, the training methods, etc. In the training methods, the levels can be the steepest descent, conjugate gradient, BFGS or DFP methods (Arora, 1989).

A.3. Choice of an experimental design

Designing the experiment means the construction of

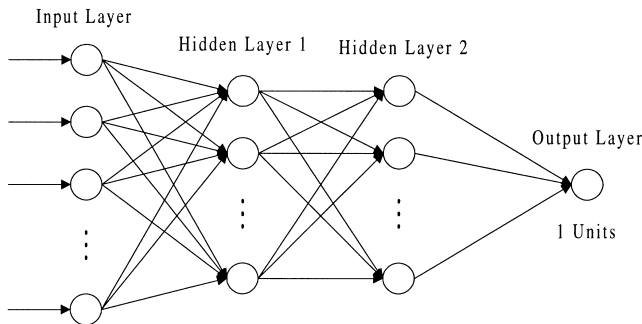


Fig. A1. An MLFF neural network.

Table A1
General arrangement for a two-factor design

		Factor B			
		1	2	...	<i>n</i>
Factor A	1	y_{11}	y_{12}	...	y_{1n}
	2	y_{21}	y_{22}	...	y_{2n}
	...				
	<i>m</i>	y_{m1}	y_{m2}	...	y_{mn}

the experiment at layout, which includes proper assignment of the selected factors, levels and interactions, to provide meaningful results containing all the information required. If there are *m* levels in factor A and *n* levels in factor B, the experimental matrix of the DOE methodology is shown in Table A1 which contains *m* × *n* treatment combinations. Other high order factorial designs can be constructed using a similar method.

On the other hand, the foundation for designing the experiment using the Taguchi method is the orthogonal array. Each array can be identified by the form $L_C(B^A)$. *L* means the “Latin Square”. The subscript of *L*, designated by *C*, represents the number of experimental runs or combinations of factors which must be conducted in the experiment. *B* is the number of levels within each factor. The letter *A*, which is the exponent of the base letter *B*, denotes the number of factors (columns) in the experiment. Some suggested orthogonal arrays can be found in Taguchi (1986). In Table A2, the orthogonal array $L_9(3^4)$ contains 9 experimental runs. Within the $L_9(3^4)$, each factor (column) contains 3 levels, and up to 11 factors can be incorporated into the experiment. For instance, the second experimental run consists of level 1 of factor 1, level 2 of factor 2, level 2 of factor 3, and level 2 of factor 4.

A.4. Performing the experiment

Conducting the experiment includes the execution and simulation of the experiment as developed in previous stages. Before the actual running of the exper-

Table A2
 $L_9(3^4)$ orthogonal array

Experiment no.	Factor 1	Factor 2	Factor 3	Factor 4
1	Level 1	Level 1	Level 1	Level 1
2	Level 1	Level 2	Level 2	Level 2
3	Level 1	Level 3	Level 3	Level 3
4	Level 2	Level 1	Level 2	Level 3
5	Level 2	Level 2	Level 3	Level 1
6	Level 2	Level 3	Level 1	Level 2
7	Level 3	Level 1	Level 3	Level 2
8	Level 3	Level 2	Level 1	Level 3
9	Level 3	Level 3	Level 2	Level 1

iment, the test plans (including the experimental order, repetition, randomization, preparation and coordination) have to be developed. These preliminary efforts are essential and are important for smooth and efficient execution of the experiment.

A.5. Analysis for DOE

The analysis phase of the experiment is employed to convert a row of data into meaningful information and to interpret the results. The first step of an analysis for DOE is to assume a statistical model for the experiment. The model contains the effects of the factors, their interactions, and error estimations. Three assumptions (normality, independence and equal variance) are applied in the model. According to the statistical model, the Analysis of Variance (ANOVA) can be accomplished using the SAS software. The ANOVA is the sensitivity analysis for the levels in each factor. Therefore, the effects of different factors, levels, and interactions between factors can be realized. Finally, the model adequacy checking has to be performed to prove the three assumptions in the model. For checking the normality, the Kruskal–Wallis Test (Montgomery, 1991) is always used. For the independence and equal variance assumptions, residual plots (Montgomery, 1991) can be used. If the model is adequate, the General Regression Model in the SAS software (Montgomery, 1991) can be used to forecast the optimum combination of the experiment. The demonstration example and the detailed descriptions of data analysis for DOE are shown in Section 4 of this paper.

A.6. Analysis for the Taguchi method

Dr. Taguchi recommends analyzing the mean response for each experiment in the orthogonal array, and analyzing the variation using the signal-to-noise (S/N) ratio. The S/N ratio for three different objective functions are:

$$\frac{S}{N} = -10 \log \left| \frac{y_1^2 + y_2^2 + \dots + y_n^2}{n} \right|, \quad (\text{A1})$$

for a smaller-the-better characteristic

$$\frac{S}{N} = -10 \log \left| \frac{\frac{1}{y_1^2} + \frac{1}{y_2^2} + \dots + \frac{1}{y_n^2}}{n} \right|, \quad (\text{A2})$$

for a larger-the-better characteristic,

where n is the number of replicates and y_n is the experimental response. Larger S/N values mean that strong signals and little noise (interference) exist during the experiment. Therefore, larger S/N values are desired in the Taguchi Method. After the S/N calculations, the level average analysis can be performed to obtain the optimum solution. The demonstration example and the detailed descriptions of data analysis for the Taguchi method are shown in Section 3 of this paper.

A.7. Confirmatory experiment

After the analysis phase, the optimum combination of the levels in each factor can be obtained. A confirmatory experiment at these settings is vital for checking the reproducibility of the optimum combinations, and for confirming the assumptions used in planning and designing the experiment.

References

- Arora, J.S., 1989. Introduction to Optimum Design. McGraw-Hill, New York.
- Gill, P.E., Murray, W., Moré, H.W., 1981. Practical Optimization. Academic Press, New York.
- John, G.P., Dimitris, G.M., 1996. Digital Signal Processing. Prentice-Hall, Englewood Cliffs, NJ.
- Khaw, B.S.L., John, F.C., Lennie, E.N.L., 1995. Optimum design of neural networks using the Taguchi method. *Neurocomputing* 7, 225–245.
- Lin, T.Y., Tseng, C.H., 1998. Using fuzzy logic and neural network in bicycle derailleur system tests. In: Proceedings of the International Conference on Advances in Vehicle Control and Safety (AVCS'98), Amiens, France, 338–343.
- Miller, G.F., Todd, P.M., Hedge, S.U., 1989. Designing neural networks using genetic algorithms. In: Proceedings of the Third International Conference on Genetic Algorithms, 379–384.
- Montgomery, D.C., 1991. Design and Analysis of Experiments. Wiley, New York.
- Patterson, D.W., 1996. Artificial Neural Networks, Theory and Applications. Prentice-Hall, Englewood Cliffs, NJ.
- Peace, G.S., 1993. Taguchi Methods, A Hands-On Approach. Addison-Wesley, Reading, MA.
- Taguchi, G., 1986. Introduction to Quality Engineering. Asian Productivity Organization, Tokyo.
- Teo, M.Y., Sim, S.K., 1995. Training the neocognitron network using design of experiments. *Artificial Intelligence in Engineering* 9, 85–94.
- Wang, C.C., Tseng, C.H., Fong, Z.H., 1996. A method for improving shifting performance. *International Journal of Vehicle Design* 18 (1), 100–117.
- Zell, A., 1995. SNNS, Stuttgart Neural Network Simulator User Manual, Version 4.1, Report No. 6/95. University of Stuttgart, IPVR, Stuttgart, Germany.