Robust Design of Artificial Neural Networks Methodology in Neutron Spectrometry

José Manuel Ortiz-Rodríguez,
Ma. del Rosario Martínez-Blanco,
José Manuel Cervantes Viramontes and
Héctor René Vega-Carrillo

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/51274

1. Introduction

Applications of artificial neural networks (ANNs) have been reported in literature in various areas. [1–5] The wide use of ANNs is due to their robustness, fault tolerant and the ability to learn and generalize, through training process, from examples, complex nonlinear and multi input/output relationships between process parameters using the process data. [6–10] The ANNs have many other advantageous characteristics, which include: generalization, adaptation, universal function approximation, parallel data processing, robustness, etc.

Multilayer perceptron (MLP) trained with backpropagation (BP) algorithm is the most used ANN in modeling, optimization classification and prediction processes. [11, 12] Although BP algorithm has proved to be efficient, its convergence tends to be very slow, and there is a possibility to get trapped in some undesired local minimum. [4, 10, 11, 13]

Most literature related to ANNs focused on specific applications and their results rather than the methodology of developing and training the networks. In general, the quality of the developed ANN is highly dependable not only on ANN training algorithm and its parameters but also on many ANN architectural parameters such as the number of hidden layers and nodes per layer which have to be set during training process and these settings are very crucial to the accuracy of ANN model. [8, 14–19]

Above all, there is limited theoretical and practical background to assist in systematical selection of ANN parameters through entire ANN development and training process. Due to this the ANN parameters are usually set by previous experience in trial and error procedure which is very time consuming. In such a way the optimal settings of ANN parameters for achieving best ANN quality are not guaranteed.

The robust design methodology, proposed by Taguchi, is one of the appropriate methods for achieving this goal. [16, 20, 21] Robust design is a statistical technique widely used to study the relationship between factors affecting the outputs of the process. It can be used to systematically identify the optimum setting of factors to obtain the desired output. In this work, it was used to find the optimum setting of ANNs parameters in order to achieve minimum error network.

1.1. Artificial Neural Networks

The first ones works about neurology were carried out by Santiago Ramón y Cajal (1852-1934) and Charles Scott Sherrington (1852-1957). Starting from their studies it is known that the basic element that conforms the nervous system is the neuron. [2, 10, 13, 22] The model of an artificial neuron it is an imitation of a biological neuron. Thus, the ANNs try to emulate the processes carried out by biological neural networks trying to build systems capable to learn from experience, pattern recognition and to realize predictions. ANN are based on a dense interconnection of small processors called nodes, neuronodes, cells, unit or processing elements or neurons.

A simplified morphology of an individual biological neuron is showed in figure 1, where can be distinguished three fundamental parts: the soma or cell body, dendrites and the cylinder-axis or axon.

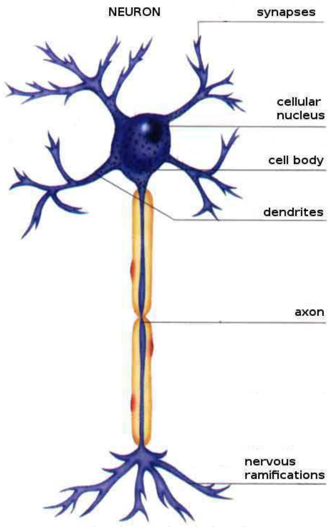


Figure 1. Simplified morphology of an individual biological neuron

Dendrites are fibers which receive the electric signals coming from other neurons and transmit them to the soma. The multiple signals coming from dendrites are processed by the soma and transmitted to the axon. The cylinder-axis or axon is a fiber of great longitude, compared with the rest of the neuron, connected to the soma for an end and divided in the other one in a series of nervous ramifications; the axon picks up the signal of the soma and transmits it to other neurons through a process known as synapses.

An artificial neuron it is a mathematical abstraction of the working of a biological neuron. [23] Figure 2, shows an artificial neuron. From a detailed observation of the biological process, the following analogies with the artificial system can be mentioned:

- The input X_i represents the signals that come from other neurons and are captured by dendrites.
- The weights W_i are the intensity of the synapses that connects two neurons; X_i and W_i are real values.
- θ is the threshold function that the neuron should exceed to be active; this process happens biologically in the body of the cell
- the input signals to the artificial neuron $X_1, X_2, ..., X_n$ are continuous variables instead of discrete pulses, as are presented in a biological neuron. Each input signal passes through a gain or weight called synaptic weight or strength of the connection whose function is similar to the synaptic function of the biological neuron.
- Weights can be positive (excitatory) or negatives (inhibitory), the summing node accumulates all the input signals multiplied by the weights and pass to the output through a threshold or transfer function.

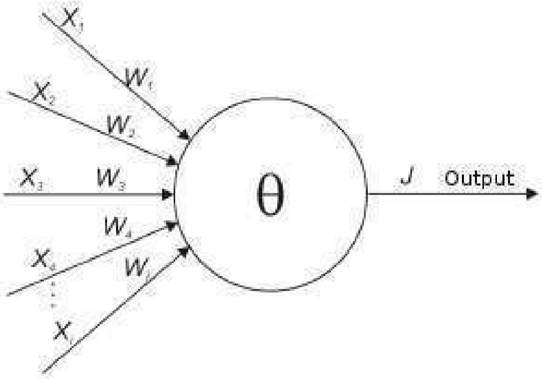


Figure 2. Artificial neuron model

An idea of this process is shown in figure 3, where can be observed a group of inputs entering to an artificial neuron.

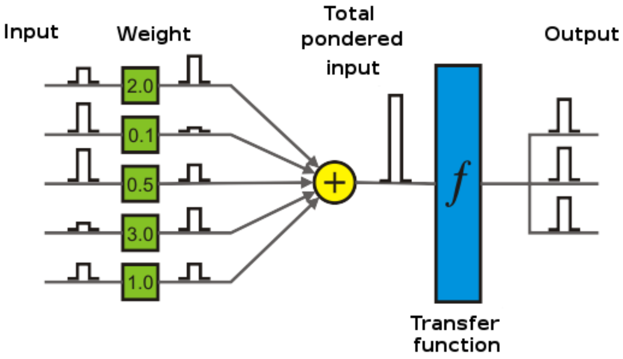


Figure 3. Analogies of an artificial neuron with a biological model

The input signals are pondered multiplying them for the corresponding weight that would correspond in the biological version of the neuron to the strength of the synaptic connection; the pondered signals arrive to the neuronal node that acts as a summing of the signals; the output of the node is denominated net output, and it is calculated as the summing of the pondered entrances plus a *b* value denominated gain. The net outpu is used as entrance to the transfer function providing the total output or answer of the artificial neuron.

The representation of figure 3 can be simplified as is shown in figure 4. From this figure, the net output of neuron *n*, can be mathematically represented as follows:

$$n = \sum_{i=1}^{r} p_i w_i + b \tag{1}$$

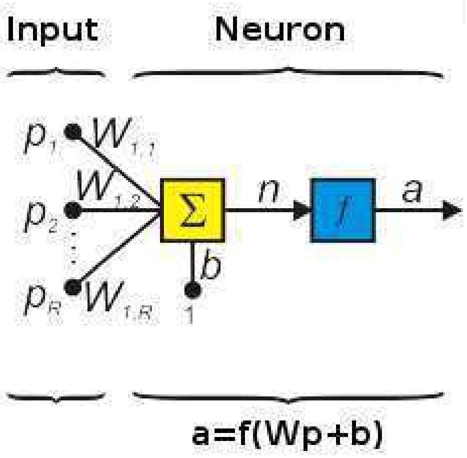


Figure 4. Didactic model of an artificial neuron

The neuronal response to the input signals *a*, can be represented as:

$$a = f(n) = \sum_{i=1}^{r} p_i w_i + b \tag{2}$$

A more didactic model, showed in figure 5, facilitates the study of a neuron.

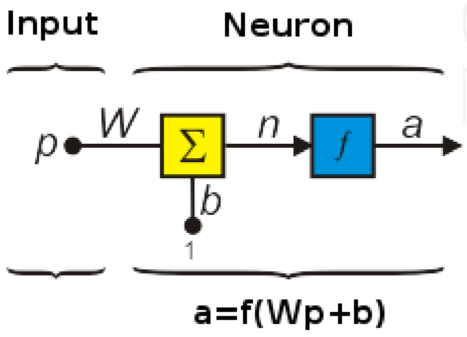


Figure 5. Didactic model of a single artificial neuron

From this figure can be seen that the net inputs are present in the vector *p*, an alone neuron only has one element; *W* represents the weights and the new input b is a gain that reinforce the output of the summing n, which is the net output of the network; the net output it is determined by the transfer function which can be a lineal or non-lineal function of n, and is chosen depending of the specifications of the problem that the neuron wants to solve.

Generally, a neuron has more than one entrance. In figure 4, can be observed a neuron with R inputs; the individual inputs $p_1, p_2, ..., p_R$ are multiplied by the corresponding weights $w_{1,1}, w_{1,2}, ..., w_{1,R}$ belonging to the weight matrix W. The sub-indexes of the weigh matrix represent the terms involved in the connection. The first sub-index represents the neuron destination, and the second represents the source of the signal that feeds to the neuron. For example, the $w_{1,2}$ indexes indicate that this weight is the connection from the second entrance to the first neuron.

This convention becomes more useful when there is a neuron with too many parameters; in this case the notation of figure 4, can be inappropriate and it is preferred to use the abbreviated notation represented in figure 6.

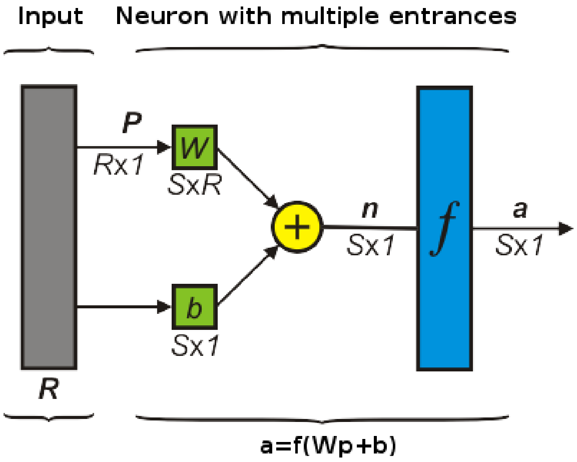


Figure 6. Abbreviated didactic model of an artificial neuron

The entrance vector p is represented by the vertical solid bar to the left. The dimensions of p are shown in the inferior part of the variable as $Rx1$, indicating that the entrance vector is a vectorial row of R elements. The entrances go to the weight matrix W, which has R columns and just one row for the case of a single neuron. A constant 1 enters to the neuron multiplied by the scalar gain b. The exit of the net a it is a scalar in this case. If the net had more than a neuron a would be a vector.

ANN are highly simplified models of the working of the brain. [10, 24] An ANN is a biologically inspired computational model which consists of a large number of simple processing elements or neurons which are interconnected and operate in parallel. [2, 13] Each neuron is connected to other neurons by means of directed communication links, which constitute the neuronal structure, each with an associated weight. [4] The weights represent information being used by the net to solve a problem.

ANNs are usually formed by several interconnected neurons. The disposition and connection varies from one type of nets to other, but in a general way the neurons are grouped by layers. A layer is a collection of neurons; according to the location of the layer in the neural net, this receives different names:

- **Input later**: receives the input signals from the environment. In this layer the information is not processed, for this reason, it is not considered as a layer of neurons.
- **Hidden layers**: these layers do not have contact with the exterior environment; the hidden layers pick up and process the information coming from the input layer; the numbers of hidden layers and neurons per layer and the form in that are connected, vary from some nets to others. Their elements can have different connections and these determine the different topologies of the net.
- **Output layer**: receives the information from the hidden layers and transmits the answer to the external means.

Figure 7 shows an ANN with two hidden layers. The outputs of first hidden layer are the entrances of the second hidden layer. In this configuration, each layer have its own weight matrix *W*, the summing, a gain vector *b*, net inputs vector *n*, the transfer function and the output vector *a*. This ANN can be observed in abbreviated notation in figure 8.

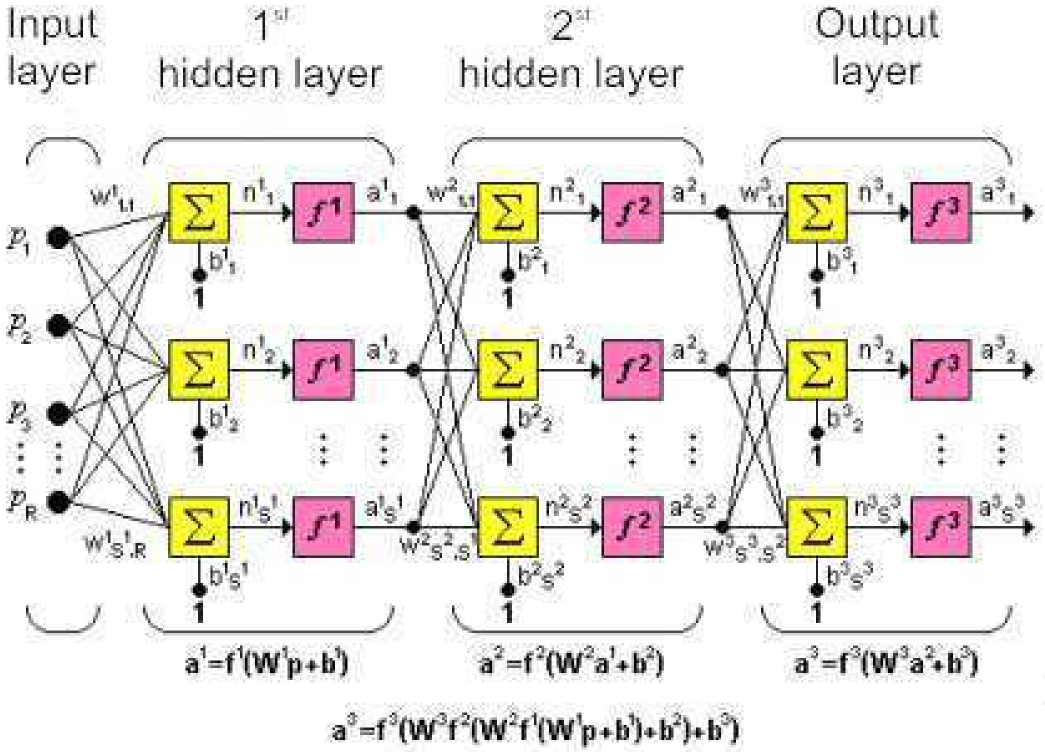


Figure 7. Artificial Neural Network with two hidden layers

Figure 8 shows a three-layer network using abbreviated notation. From this figure can be seen that the network has $\mathbf{R}^1$ inputs, $\mathbf{S}^1$ neurons in the first layer, $\mathbf{S}^2$ neurons in the second layer, etc. A constant input 1 is fed to the bias for each neuron. The outputs of each intermediate layer are the inputs to the following layer. Thus layer 2 can be analyzed as a one-layer network with $\mathbf{S}^1$ inputs, $\mathbf{S}^2$ neurons, and an $\mathbf{S}^2 x \mathbf{S}^1$ weight matrix $\mathbf{W}^2$. The input to layer 2 is $\mathbf{a}^1$; the output is $\mathbf{a}^2$. Now that all the vectors and matrices of layer 2 have been identified, it can be treated as a single-layer network on its own. This approach can be taken with any layer of the network.

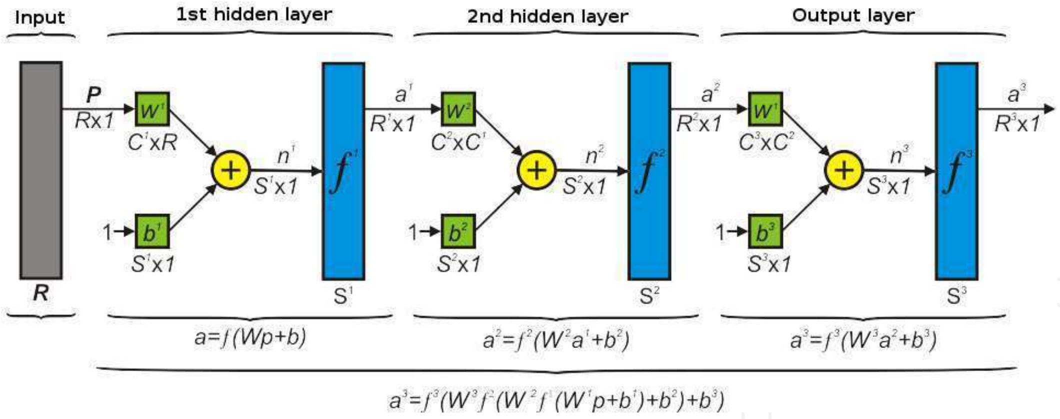


Figure 8. Artificial Neural Network with two hidden layers in abreviated notation

The arrangement of neurons into layers and the connection patterns within and between layers is called the *net architecture* [6, 7]. According to the absence or presence of feedback connections in a network, two types of architectures are distinguished:

- **Feedforward architecture**. There are no connections back from the output to the input neurons; the network does not keep a memory of its previous output values and the activation states of its neurons; the perceptron-like networks are feedforward types.
- **Feedback architecture**. There are connections from output to input neurons; such a network keeps a memory of its previous states, and the next state depends not only on the input signals but on the previous states of the network; the Hopfield network is of this type.

Back propagation feed forward neural nets is a network with supervised learning which uses a propagation-adaptation cycle of two phases. Once a pattern has been applied to the input of the network as a stimulus, this is propagated from the first layer through the superior layers of the net until generate an output. The output signal is compared with the desired output and a signal error is calculated for each one of the outputs.

The outputs errors are back propagated from the output layer toward all the neurons of the hidden layer that contribute directly with the output. However, the neurons of the hidden layer only receive a fraction of the signal from the whole error signal, based on the relative contribution that has contributed each neuron to the original output. This process is repeated for each layer until all neurons of the network have received an error signal which describes its relative contribution to the total error. Based on the perceived signal error the connection synaptic weights of each neuron are upgrade to make that the net converges toward a state that allows to classify correctly all the patterns of training.

The importance of this process consists in that as trains the net, those neurons of the intermediate layers are organized themselves in such a way that the neurons learn how to recognize different features of the whole entrance space. After the training, when they are presented an arbitrary pattern of entrance that contain noise or that it is incomplete, the neurons of the hidden layer of the net will respond with an active output if the new entrance contains a pattern that resembles each other to that characteristic that the individual neurons have learned how to recognize during their training. And to the inverse one, the units of the hidden layers have one tendency to inhibit their output if the entrance pattern does not contain the characteristic to recognize, for which they have been trained.

During the training process, the Backpropagation net tends to develop internal relationships among neurons with the purpose to organize the training data in classes. This tendency can be extrapolated to arrive to the hypothesis that all the units of the hidden layer of a Backpropagation are associated somehow to specific characteristic of the entrance pattern as consequence of the training. That the association is or not exact, it cannot be evident for the human observer, the important thing it is that the net has found one internal representation that allows him to generate the wanted outputs when are given the entrances in the training process. This same internal representation can be applied to entrances that the net has not seen before, and the net will classify these entrances according to the characteristics that share with the examples of training.

In recent years, there is increasing interest in using ANNs for modeling, optimization and prediction. The advantages that ANNs offer are numerous and are achievable only by developing an ANN model of high performance. However, determining suitable training and architectural parameters of an ANN still remains a difficult task mainly because it is very hard to know beforehand the size and the structure of a neural network one needs to solve a given problem. An ideal structure is a structure that independently of the starting weights of the net, always learns the task, i.e. makes almost no error on the training set and generalizes well.

The problem with neural networks is that a number of parameter have to be set before any training can begin. Users have to choose the architecture and determine many of the parameters in a selected network. However, there are no clear rules how to set these parameters. Yet, these parameters determine the success of the training.

As can be appreciated in figure 9, the current practice in the selection of design parameters for ANN is based on the trial and error procedure, where a large number of ANN models are developed and compared to one another. If the level of a design parameter is changed and does not have effect in the performance of the net, then a different design parameter is varied, and the experiment is repeated in a series of approaches. The observed answers are examined in each phase, to determine the best level in each design parameter.

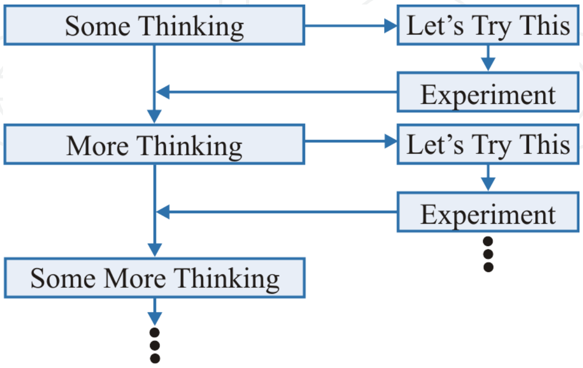


Figure 9. Trial-and-error procedure in the selection of ANN parameters

The serious inconvenience of this method is that a parameter is evaluated while the other ones are maintained in an only level. Of here, the best selected level in a design variable in particular, could not necessarily be the best at the end of the experimentation, since those other parameters could have changed. Clearly, this method cannot evaluate interactions among parameters since only combines one at the same time it could lead to an ANN impoverished design in general.

All of these limitations have motivated researchers to generate ideas of merging or hybridizing ANN with other approaches in the search for better performance. A form of overcoming this disadvantage, is to evaluate all the possible level combinations of the design parameters, i.e., to carry out a complete factorial design. However, since the number of combinations can be very big, even for a small number of parameters and levels, this method is very expensive and consumes a lot of time. The number of experiments to be carried out can be decreased making use of the factorial fractional method, a statistical method based on the robust design of Taguchi philosophy.

The Taguchi technique is a methodology for finding the optimum setting of the control factors to make the product or process insensitive to the noise factors. Taguchi based optimization technique has produced a unique and powerful optimization discipline that differs from traditional practices.[16, 20, 21]

1.2. Taguchi philosophy of robust design

Designs of experiments involving multiple factors was first proposed by R. A. Fisher, in the 1920s to determine the effect of multiple factors on the outcome of agricultural trials (Ranjit 1990). This method is known as *factorial design of experiments*. A full factorial design identifies all possible combinations for a given set of factors. Since most experiments involve a significant number of factors, a full factorial design may involve a large number of experiments.

Factors are the different variables which determines the functionality or performance of a product or system as for example, design parameters that influence the performance or input that can be controlled.

Dr. Genichi Taguchi is considered the author of the robust design of parameters. [8, 14–21] This is an engineering method for the design of products or processes focused in diminishing the variation and/or sensibility to the noise. When it is used appropriately, Taguchi design provides a powerful and efficient method for the design of products that operate consistent and optimally about a variety of conditions. In the robust design of parameters, the primary objective is to find the selection of the factors that decrease the variation of the answer, while the processes are adjusted on the objective.

The distinct idea of Taguchi's robust design that differs from the conventional experimental design is the simultaneous modeling of both mean and variability in the designing. However, Taguchi methodology is based on the concept of fractional factorial design.

By using Orthogonal Arrays (OAs) and fractional factorial instead of full factorial, Taguchi's approach allows to study the entire parameter space with a small number of experiments. An OA is a small fraction of full factorial design and assures a balanced comparison of levels of any factor or interaction of factors. The columns of an OA represent the experimental parameters to be optimized and the rows represent the individual trials (combinations of levels).

Taguchi's robust design can be divided into two classes: static and dynamic characteristics. The static problem attempts to obtain the value of a quality characteristic of interest as close as possible to a single specified target value. The dynamic problem, on the other hand, involves situations where a system's performance depends on a signal factor.

Taguchi also proposed a two-phase procedure to determine the factors level combination. First, the control factors that are significant for reducing variability are determined and their settings are chosen. Next, the control factors that are significant in affecting the sensitivity are identified and their appropriate levels are chosen. The objective of the second phase is to adjust the responses to the desired values.

The Taguchi method is applied in four steps.

1. **Brainstorm the quality characteristics and design parameters important to the product/process**.

   In Taguchi methods there are variables that are under control and variables that are not. These are called design and noise factors, respectively, which can influence a product and operational process. The design factors, as controlled by the designer, can be divided into: (1) signal factor, which influences the average of the quality response and (2) control factor, which influences the variation of the quality response. The noise factors are uncontrollable such as manufacturing variation, environmental variation and deterioration.

   Before designing an experiment, knowledge of the product/process under investigation is of prime importance for identifying the factors likely to influence the outcome. The aim of the analysis is primarily to seek answers to the following three questions:

   (a) What is the optimum condition?

   (b) Which factors contribute to the results and by how much?

   (c) What will be the expected result at the optimum condition?

2. **Design and conduct the experiments**.

   Taguchi's robust design involves using an OA to arrange the experiment and selecting the levels of the design factors to minimize the effects of the noise factors. That is, the settings of the design factors for a product or a process should be determined so that the product's response has the minimum variation, and its mean is close to the desired target.

   To design an experiment, the most suitable OA is selected. Next, factors are assigned to the appropriate columns, and finally, the combinations of the individual experiments (called the trial conditions) are described. Experimental design using OAs is attractive because of experimental efficiency.

   The array is called orthogonal because for every pair of parameters, all combinations of parameter levels occur an equal number of times, which means the design is balanced so that factor levels are weighted equally. The real power in using an OA is the ability to evaluate several factors in a minimum of tests. This is considered an efficient experiment since much information is obtained from a few trials. The mean and the variance of the response at each setting of parameters in OA are then combined into a single performance measure known as the signal-to-noise (S/N) ratio.

3. **Analyze the results to determine the optimum conditions**.

   The S/N ratio is a quality indicator by which the experimenters can evaluate the effect of changing a particular experimental parameter on the performance of the process or product. Taguchi used S/N ratio to evaluate the variation of the system's performance, which is derived from quality loss function. For static characteristic, Taguchi classified them into three types of S/N ratios:

   (a) Smaller-the-Better (STB)

   (b) Larger-the-Better (LTB)

   (c) Nominal-the-Best (NTB)

   For the STB and LTB cases, Taguchi recommended direct minimization of the expected loss. For the NTB case, Taguchi developed a two-phase optimization procedure to obtain the optimal factor combinations.

   For the dynamic characteristics the SN ratio

$$SN_i = 10log_{10}(\beta_i / MSE_i) \tag{3}$$

   is used for evaluating the S/N ratio, where the mean square error (MSE) represents the mean square of the distance between the measured response and the best fitted line; denotes the sensitivity.

4. **Run a confirmatory test using the optimum conditions**.

   The two major goals of parameter design are to minimize the process or product variation and to design robust and flexible processes or products that are adaptable to environmental conditions. Taguchi methodology is useful for finding the optimum setting of the control factors to make the product or process insensitive to the noise factors.

   In this stage, the value of the robustness measure is predicted at the optimal design condition; a confirmation experiment at the optimal design condition is conducted, calculating the robustness measure for the performance characteristic and checking if the robustness prediction is close to the predicted value.

Today, ANN can be trained to solve problems that are difficult for conventional computers or human beings, and have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems. Recently, the use of ANN technology has been applied with relative success in the research area of nuclear sciences,[3] mainly in the neutron spectrometry and dosimetry domains. [25–31]

1.3. Neutron spectrometry with ANNs

The measurement of the intensity of a radiation field with respect to certain quantity like angle, energy, frequency, etc., is very important in radiation spectrometry having, as a final result, the radiation spectrum. [32–34] The radiation spectrometry term can be used to describe measurement of the intensity of a radiation field with respect to energy, frequency or momentum. [35] The distribution of the intensity with one of these parameters is commonly referred to as the "spectrum". [36] A second quantity is the variation of the intensity of these radiations as a function of angle of incidence on a body situated in the radiation field and is referred as “dose”. The neutron spectra and the dose are of great importance in radiation protection physics. [37]

Neutrons are found in the environment or are artificially produced by different ways; these neutrons have a wide energy range extending from few thousandths of eV to several hundreds of MeV. [38] Also, they are in a broad variety of energy distributions, named neutron-fluence spectrum or simply neutron spectrum, $\Phi_E(E)$.

Determination of neutron dose received by those exposed to workplaces or accidents in nuclear facilities, generally requires knowledge of the neutron energy spectrum incident on the body. [39] Spectral information must generally be obtained from passive detectors which respond to different ranges of neutron energies such as the multispheres Bonner system or Bonner spheres system (BSS). [40–42]

BSS system, has been used to unfold the neutron spectra mainly because it has an almost isotropic response, can cover the energy range from thermal to GeV neutrons, and is easy to operate. However, the weight, time consuming procedure, the need to use an unfolding procedure and the low resolution spectrum are some of the BSS drawbacks. [43, 44]

As can be seen from figure 10, BSS consists of a thermal neutron detector such as $^6LiI(Eu)$, Activation foils, pairs of thermoluminiscent dosimeters or track detectors, which is placed at the centre of a number of moderating spheres made of polyethylene of different diameter to obtain, through an unfolding process the neutron energy distribution, also known as spectrum, $\Phi_E(E)$. [42, 45]

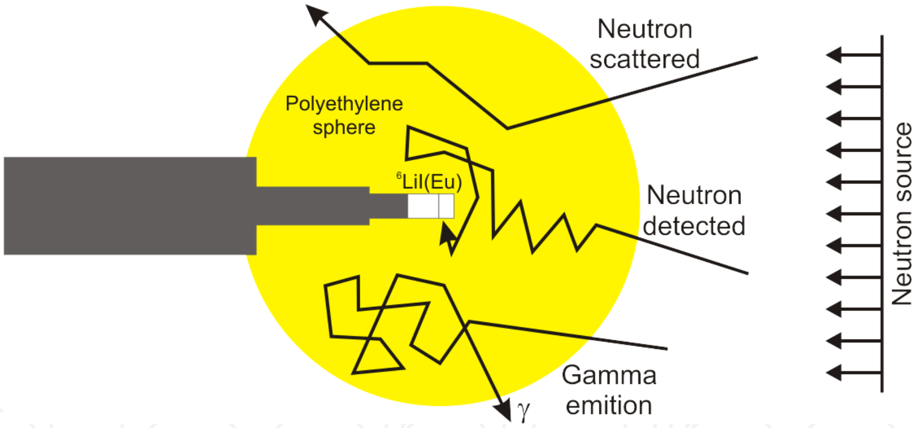


Figure 10. Bonner spheres system for a $^6LiI(Eu)$ neutrons detector

The derivation of the spectral information is not simple; the unknown neutron spectrum is not given directly as a result of the measurements. [46] If a sphere d has a response function $R_d(E)$, and is exposed in a neutron field with spectral fluence $\Phi_E(E)$, the sphere reading M_d is obtained by folding $R_d(E)$ with $\Phi_E(E)$, this means to solve the Fredholm integral equation of the first kind shown in equation 4.

$$M_d = \int R_d(E)\Phi_E(E)dE \tag{4}$$

This folding process takes place in the sphere itself during the measurement. Although the real $\Phi_E(E)$ and $R_d(E)$ are continuous functions of neutron energy, they cannot be described by analytical functions, and, as a consequence, a discretised numerical form is used, showed in the following equation:

$$C_j = \sum_{i=1}^{N} R_{i,j}\Phi_i - - - > j = 1, 2, ..., m \tag{5}$$

where C_j is j^{th} detector's count rate; $R_{i,j}$ is the j^{th} detector's response to neutrons at the i^{th} energy interval; Φ_i is the neutron fluence within the i^{th} energy interval and m is the number of spheres utilized.

Once the neutron spectrum, $\Phi_E(E)$, has been obtained, the dose $\triangle$ can be calculated using the fluence-to-dose conversion coefficients $\delta_\Phi E$, as shown in equation 3.

$$\triangle = \int_{E_{min}}^{E_{max}} \delta_\Phi E \Phi_E(E) dE \tag{6}$$

Equation 5 is an ill-conditioned equations system with an infinite number of solutions which have motivated researches to propose new and complementary approaches. To unfold the neutron spectrum, Φ, several methods are used. [43, 44, 47] ANN technology is a useful alternative to solve this problem; [25–31] however, several drawbacks must be solved in order to simplify the use of these procedures.

Besides many advantages that ANNs offer, there are some drawbacks and limitations related to ANN design process. In order to develop an ANN which generalizes well and be robust, a number of issues must be taken into consideration, particularly related to architecture and training parameters. [8, 14–21] The trial-and-error technique is the usual way to get a better combination of these vales. This method cannot identify interactions between the parameters and do not use systematic methodologies for the identification of the "best" values, consuming much time and does not systematically target a near optimal solution, which may lead to a poor overall neural network design.

Even though the BP learning algorithm provides a method for training multilayer feed forward neural nets, is not free of problems. Many factors affect the performance of the learning and should be treated for having a successful learning process. Those factors include the synaptic weight initialization, the learning rate, the momentum, the size of the net and the learning database. A good election of these parameters could speed up and improve in great measure the learning process to reach the goal, although a universal answer does not exist for such topics.

Choosing the ANN architecture followed by selection of training algorithm and related parameters is rather a matter of the designer past experience since there are no practical rules which could be generally applied. This is usually a very time consuming trial and error procedure where a number of ANNs are designed and compared to one another. Above all, the design of optimal ANN is not guaranteed. It is unrealistic to analyze all combination of ANN parameters and parameter's levels effects on the ANN performance.

To deal economically with the many possible combinations, the Taguchi method can be applied. Taguchi's techniques have been widely used in engineering design, and can be applied to many aspects such as optimization, experimental design, sensitivity analysis, parameter estimation, model prediction, etc.

This work is concerned with the application of Taguchi method for the optimization of ANN models. The integration of ANN and Taguchi's optimization provides a tool for designing robust network parameters and improving their performance. The Taguchi method offers considerable benefits in time and accuracy when is compared with the conventional trial and error neural network design approach.

In this work, for the robust design of multilayer feedforward neural networks trained by backpropagation algorithm in the neutron spectrometry field, a systematic and experimental strategy called *Robust Design of Artificial Neural Networks* (RDANN) methodology was designed. This computer tool, emphasizes simultaneous ANNs parameters optimization under various noise conditions. Here, we make a comparison among this method and conventional training methods. The attention is drawing on the advantages on Taguchi methods which offer potential benefits in evaluating the network behavior.

2. Robust design of artificial neural networks methodology

Neutron spectrum unfolding is an ill-conditioned system with an infinite number of solutions. [27] Researchers have using ANNs to unfold neutron spectra from BSS. [48] Figure 11, shows the classical approach of neutron spectrometry by means ANN technology starting from rate counts measured with BSS.

As can be appreciated in figure 11, neutron spectrometry by means of ANN technology is done by using a neutron spectra data set compiled by the International Atomic Energy Agency (IAEA). [49] This compendium contains a large collection of detector responses and spectra. The original spectra in this report were defined per unit lethargy in 60 energy groups ranging from thermal to 630 MeV.

One challenge in neutron spectrometry using neural nets is the pre-processing of the information in order to create suitable pair input-output training data sets. [50] The generation of a suitable data set is a non trivial task. Because the novelty of this technology in this research area, the researcher spent a lot of time in this activity mainly because all the work is done by hand and a lot of effort is required. From the anterior, it is evident the need to have technological tools that automate this process. At present, work is being realized in order to alleviate this drawback.

In order to use the response matrix known as UTA4, expressed en 31 energy groups, ranging from 10^{-8} up to 231.2 MeV in the ANN training process, the energy range of neutrons spectra was changed through a re-binning process by means of MCNP simulations. [50] 187 neutrons spectra from IAEA compilation, expressed in energy units and in 60 energy bins, were re-binned into the thirty-one energy groups of the UTA4 response matrix, and at the same time, 13 different equivalent doses were calculated per spectra by using the International Commission on Radiological Protection (ICRP) fluence-to-dose conversion factors.

Figure 12, shows the re-binned neutron spectra data set used for training and testing the optimum ANN architecture designed with RDANN methodology.

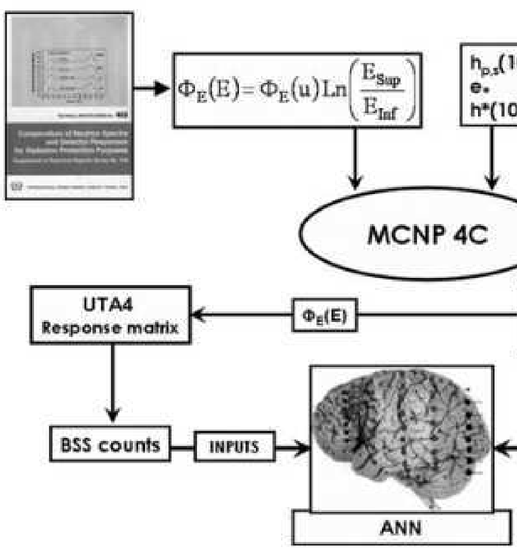


Figure 11. Classical Neutron spectrometry with ANN technology

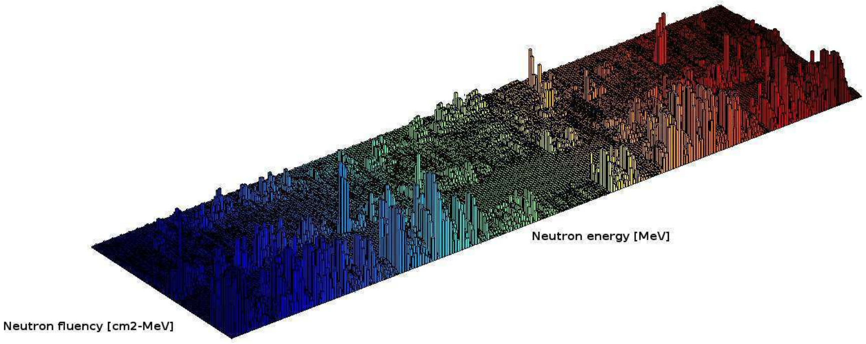


Figure 12. Re-binned neutron spectra data set used to train the optimum ANN architecture designed with RDANN methodology

Multiplying re-binned neutron spectra by UTA4 response matrix, the rate counts data set was calculated. Re-binned spectra and equivalent doses are the desired output of ANN and its corresponding calculated rate counts the entrance data.

The second one challenge in neutron spectrometry by means ANN, is the determination of the net topology. In the ANN design process, the choice of the ANN's basic parameters often determines the success of the training process. The selection of these parameters follows in practical use no rules, and their value is at most arguable. This method consuming much time and does not systematically target a near optimal solution to select suitable parameter values. The ANN designers have to choose the architecture and determine many of the parameters through the trial and error technique, which produces ANN with poor performance and low generalization capability, spending often large amount of time.

An easier and more efficient way to overcome this disadvantage is to use the RDANN methodology, showed in figure 13, which has become in a new approach to solve this problem.

RDANN is a very powerful method based on parallel processes where all the experiments are planed a priori and the results are analyzed after all the experiments are completed. This is a

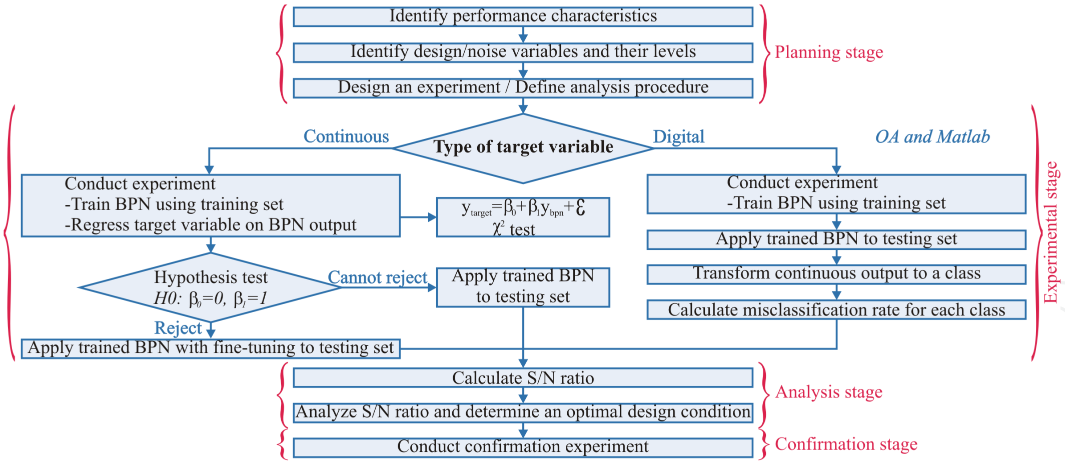

Identify performance characteristics
Identify design/noise variables and their levels
Design an experiment / Define analysis procedure
Planning stage
Continuous
Type of target variable
Digital
OA and Matlab
Conduct experiment
-Train BPN using training set
-Regress target variable on BPN output
$y_{target}=\beta_0+\beta_1 y_{bpn}+\varepsilon$
χ^2 test
Conduct experiment
-Train BPN using training set
Apply trained BPN to testing set
Transform continuous output to a class
Calculate misclassification rate for each class
Hypothesis test
$H0: \beta_0=0, \beta_1=1$
Cannot reject
Apply trained BPN to testing set
Reject
Apply trained BPN with fine-tuning to testing set
Experimental stage
Calculate S/N ratio
Analyze S/N ratio and determine an optimal design condition
Analysis stage
Conduct confirmation experiment
Confirmation stage

Figure 13. Robust design of artificial neural networks methodology

systematic and methodological approach of ANN design, based on the Taguchi philosophy, which maximize the ANN performance and generalization capacity.

The integration of neural networks and optimization provides a tool for designing ANN parameters improving the network performance and generalization capability. The main objective of the proposed methodology is to develop accurate and robust ANN models. In other words, the goal is to select ANN training and architectural parameters, so that the ANN model yields best performance.

From figure 13 can be seen that in ANN design using Taguchi philosophy in RDANN methodology, the designer 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 need to be determined (Planning stage). Using OAs, simulations, i.e., training of ANNs with different net topologies can be executed in a systematic way (experimentation stage). From simulation results, the response can be analyzed by using S/N ratio from Taguchi method (Analysis stage). Finally, a confirmation experiment at the optimal design condition is conducted, calculating the robustness measure for the performance characteristic and checking if the robustness prediction is close to the predicted value (Confirmation stage).

To provide scientific discipline to this work, in this research the systematic and methodological approach called RDANN methodology was used to obtain the optimum architectural and learning values of an ANN capable to solve the neutron spectrometry problem.

According figure 13, the steps followed to obtain the optimum design of the ANN are described:

1. **Planning stage**

   In this stage it is necessary to identify the objective function and the design and noise variables.

   (a) *The objective function*. The objective function must be defined according to the purpose and requirements of the problem.

   In this research, the objective function is the prediction or classification errors between the target and the output values of BP ANN at testing stage, i.e., the performance or mean square error (MSE) output of the ANN is used as the objective function as is showed in the following equation:

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\Phi_E(E)_i^{ANN} - \Phi_E(E)_i^{ORIGINAL})^2} \tag{7}$$

Where N is the number of trials, $\Phi_E(E)_i^{ORIGINAL}$ is the original spectra and $\Phi_E(E)_i^{ANN}$ is the spectra unfolded with ANN.

(b) *Design and noise variables*. Based in 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.

Among the various parameters that affect the ANN performance, four design variables were selected, as is showed in the table 1:

Design Var.	Level 1	Level 2	Level 3
A	L1	L2	L3
B	L1	L2	L3
C	L1	L2	L3
D	L1	L2	L3

Table 1. Design variables and their levels

where **A** is the number of neurons in the first hidden layer, **B** is the number of neurons in the second hidden layer, **C** is the momentum and **D** is the learning rate.

Noise variables are shown in table 2. These variables in most cases are not controlled by the user. The initial set of weights, U, usually is randomly selected; In the training and testing data sets, V, the designer must decide how much of the whole data should be allocated to the training and testing data sets. Once V is determined, the designer must decide which data of the whole data set to include in the training and testing data set, W.

Design Var.	Level 1	Level 2
U	Set 1	Set 2
V	6:4	8:2
W	Tr-1/Tst-1	Tr-2/Tst-2

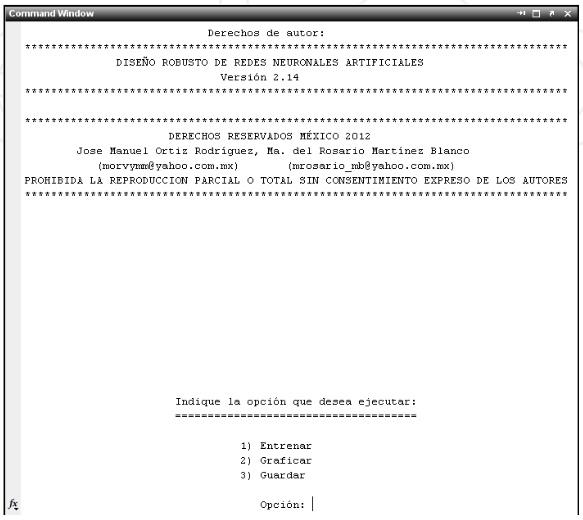
Table 2. Noise variables and their levels

where **U** is the initial set of random weights, **V** is the size of training set versus size of testing set, i.e., **V** = 60% / 40%, 80% / 20% and **W** is the selection of training and testing sets, i.e., **W** = Training1/Test1, Training2/Test2.

In practice, these variables are randomly determined, and are not controlled by designer. Because the random nature of this selection processes, the ANN designer must create these data sets starting from the whole data set. This procedure is very time consuming when is done by hand without the help of technological tools.

RDANN methodology was designed in order to fully automate in a computer program, developed under Matlab environment and showed in figure 14, the creation of the noise variables and their levels. This work is done before the training of the several net topologies tested at experimentation stage.

Besides the automatic generation of noise variables, another programming routines were created in order to train the different net architectures, and to statistically analyze and graph the obtained data. When this procedure is done by hand, is very time consuming. The use of the designed computer tool saves a lot of time and effort to ANN designer.



```
                            Derechos de autor:
***********************************************************************************
              DISEÑO ROBUSTO DE REDES NEURONALES ARTIFICIALES
                              Versión 2.14
***********************************************************************************

***********************************************************************************
                      DERECHOS RESERVADOS MÉXICO 2012
        Jose Manuel Ortiz Rodríguez, Ma. del Rosario Martínez Blanco
           (morvymm@yahoo.com.mx)       (mrosario_mb@yahoo.com.mx)
PROHIBIDA LA REPRODUCCION PARCIAL O TOTAL SIN CONSENTIMIENTO EXPRESO DE LOS AUTORES
***********************************************************************************








                       Indique la opción que desea ejecutar:
                       =====================================

                                 1) Entrenar
                                 2) Graficar
                                 3) Guardar

                                    Opción:
```

Figure 14. Robust Design of Artificicial Neural Networks Methodology

After the factors and levels are determined, a suitable OA can be selected for training process. The Taguchi OAs are denoted by $L_r(s^c)$ where r is the number of rows, c is the number of columns and s is the number of levels in each column.

2. **Experimentation stage**. The choice of a suitable OA is critical for the success of this stage. OA allow to compute the main interaction effects via a minimum number of experimental trails. In this research, the columns of OA represent the experimental parameters to be optimized and the rows represent the individual trials, i.e., combinations of levels.

   For a robust experimental design, Taguchi suggest to use two crossed OAs with a $L_9(3^4)$ y $L_4(3^2)$ configuration, as is showed in table 3.

   From table 3, can be seen that a design variable is assigned to a column of the OA. Then, each row of the design OA represents a specific design of ANN. Similarly, a noise variable is assigned to a column of the noise OA, each row corresponds to a noise condition.

3. **Analysis stage**.

   The S/N ratio is a measure of both, the location and dispersion of the measured responses. It transforms the row data to allow quantitative evaluation of the design parameters considering their mean and variation. It is measured in decibels using the formula:

$$S/N = 10log_{10}(MSD) \tag{8}$$

where MSD is a measure of the mean square deviation in performance, since in every design, more signal and less noise is desired. The best design will have the highest S/N ratio. In this stage, the statistical program JMP was used to select the best values of the ANN being designed.

Trial No.	A	B	C	D	S1	S2	S3	S4	Mean	S/N
1	1	1	1	1						
2	1	2	2	2						
3	1	3	3	3						
4	2	1	2	3						
5	2	2	3	1						
6	2	3	1	2						
7	3	1	3	2						
8	3	2	1	3						
9	3	3	2	1						

Table 3. ANN measured responses with a crossed OA with $L_9(3^4)$ y $L_4(3^2)$ configuration

4. **Confirmation stage**.
   In this stage, the value of the robustness measure is predicted at the optimal design condition; a confirmation experiment at the optimal design condition is conducted, calculating the robustness measure for the performance characteristic and checking if the robustness prediction is close to the predicted value.

3. Results and discussion

RDANN methodology was applied in nuclear sciences in order to solve the neutron spectrometry problem starting from the count rates of BSS System with a ^{6}LiI(Eu) thermal neutrons detector, 7 polyethylene spheres and the UTA4 response matrix expressed 31 energy bins.

In this work, a feed-forward ANN trained with BP learning algorithm was designed. For ANN training, the *"trainscg"* training algorithm and mse = $1E^{-4}$ were selected. In RDANN methodology an OA with $L_9(3^4)$ and $L_4(3^2)$ configuration, corresponding to design and noise variables respectively, was used. The optimal net architecture was designed in short time and has high performance and generalization capability.

The obtained results after applying RDANN methodology are:

3.1. Planning stage

Tables 4 and 5 shown the design and noise variables selected and their levels.

Design Var.	Level 1	Level 2	Level 3
A	14	28	56
B	0	28	56
C	0.001	0.1	0.3
D	0.1	0.3	0.5

Table 4. Design variables and their levels

where **A** is the number of neurons in the first hidden layer, **B** is the number of neurons in the second hidden layer, **C** is the momentum, **D** is the learning rate.

Design Var.	Level 1	Level 2
U	Set 1	Set 2
V	6:4	8:2
W	Tr-1/Tst-1	Tr-2/Tst-2

Table 5. Noise variables and their levels

where **U** is the initial set of random weights, **V** is the size of training set versus size of testing set, i.e., **V** = 60% / 40%, 80% / 20% and **W** is the selection of training and testing sets, i.e., **W** = Training1/Test1, Training2/Test2.

3.2. Experimentation Stage

In this stage by using a crossed OA with $L_9(3^4)$, $L_4(3^2)$ configuration, 36 different ANNs architectures were trained and tested as is showed in table 6.

Trial No.	Resp-1	Resp-2	Resp-3	Resp-4	Median	S/N
1	3.316E-04	2.416E-04	2.350E-04	3.035E-04	2.779E-04	3.316E-04
2	2.213E-04	3.087E-04	3.646E-04	2.630E-04	2.894E-04	2.213E-04
3	4.193E-04	3.658E-04	3.411E-04	2.868E-04	3.533E-04	4.193E-04
4	2.585E-04	2.278E-04	2.695E-04	3.741E-04	2.825E-04	2.585E-04
5	2.678E-04	3.692E-04	3.087E-04	3.988E-04	3.361E-04	2.678E-04
6	2.713E-04	2.793E-04	2.041E-04	3.970E-04	2.879E-04	2.713E-04
7	2.247E-04	7.109E-04	3.723E-04	2.733E-04	3.953E-04	2.247E-04
8	3.952E-04	5.944E-04	2.657E-04	3.522E-04	4.019E-04	3.952E-04
9	5.425E-04	3.893E-04	3.374E-04	4.437E-04	4.282E-04	5.425E-04

Table 6. ANN measured responses with a corssed OA with L9,L4 configuration

The signal-to-noise ratio was analyzed by means of Analysis of Variance (ANOVA) by using the statistical program JMP. Since an error of $1E^{-4}$ was established for the objective function, from table 6, can be seen that all ANN performances reach this value. This means that this particular OA has a good performance.

3.3. Analysis stage

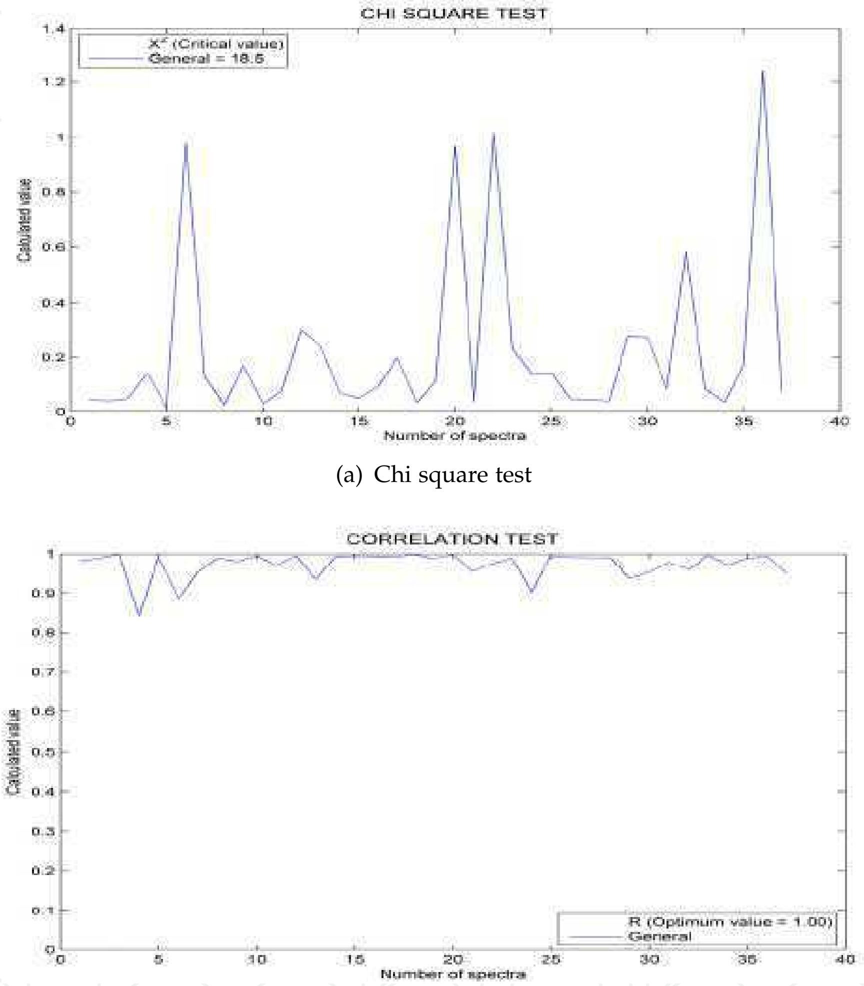
The signal-to-noise ratio is used in this stage to determine the optimum ANN architecture. The best ANN's design parameters are showed in table 7.

Trial No.	A	B	C1	C2	C3	D
1	14	0	0.001	0.001	0.001	0.1
2	**14**	**0**	0.001	**0.1**	0.3	**0.1**
3	56	56	0.001	0.1	0.1	0.1

Table 7. Best values used in the confirmation stage to design the ANN

3.4. Confirmation stage

Once optimum design parameters were determined, the confirmation stage was performed to determine the final optimum values, highlighted in table 7. After the best ANN topology was determined a final training and testing was made to validate the data obtained with the ANN designed. At final ANN validation and using the designed computational tool, correlation and Chi square statistical tests were carried out as shown in figure 15.



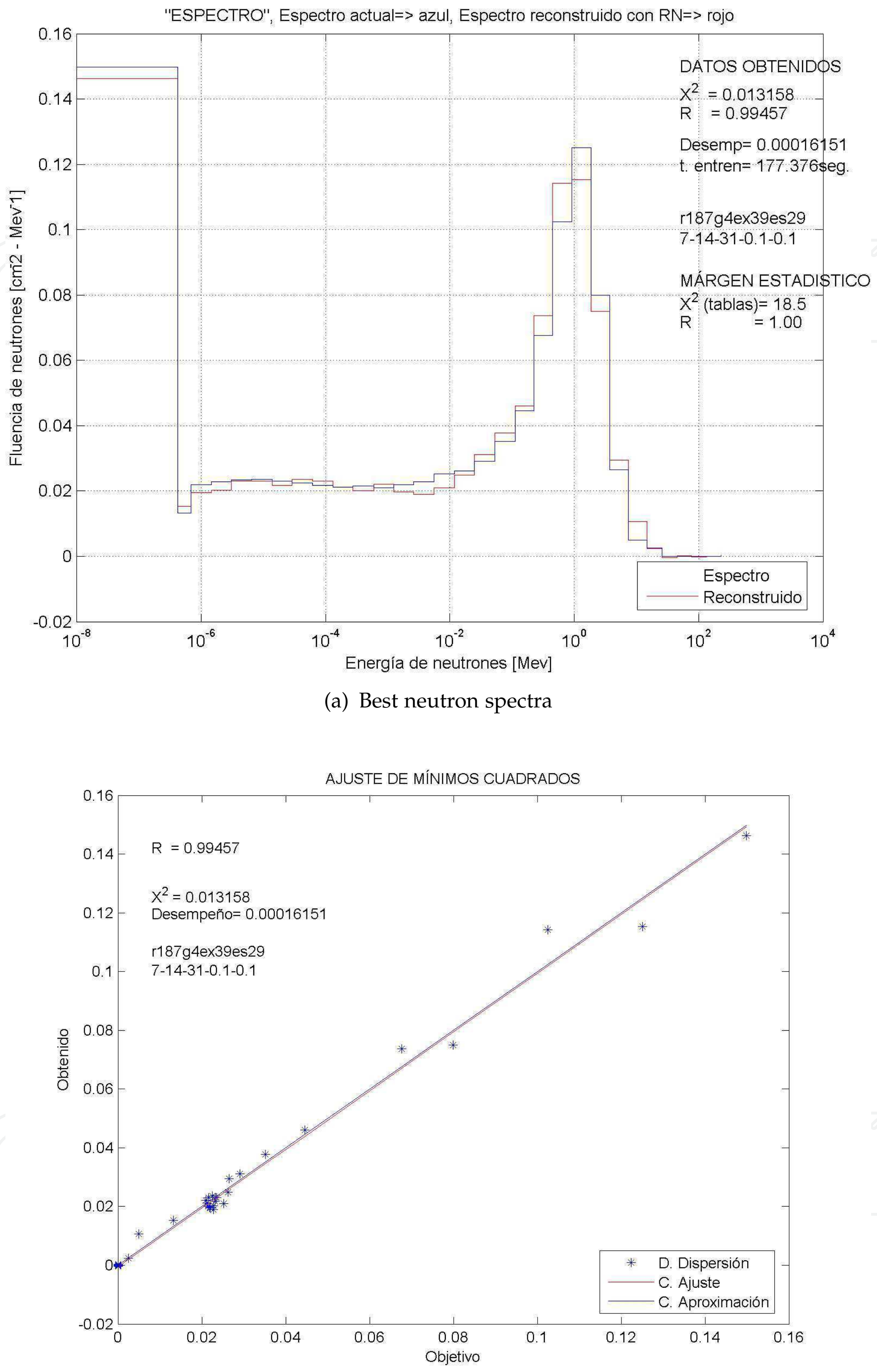
(a) Chi square test

(b) Correlation test

Figure 15. Subfigure with four images

From figure 15(a), can be seen that all neutron spectra pass the Chi square statistical test, which demonstrate that statistically there is not difference among the neutron spectra reconstructed by the designed ANN and the target neutron spectra. Similarly from figure 15(b), can be seen that the whole data set of neutron spectra is near of the optimum value equal to one, which demonstrate that this is an OA with high quality.

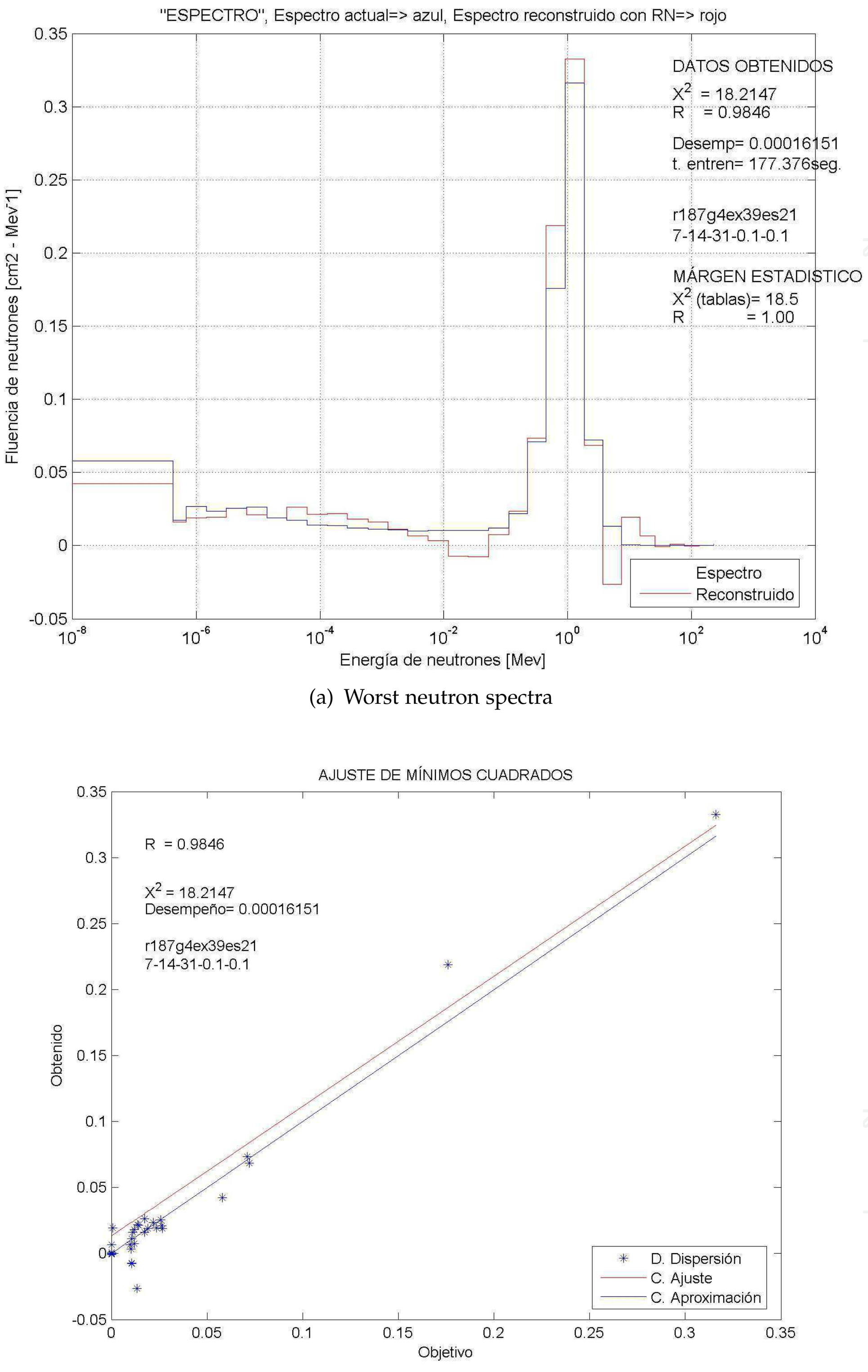
Figure 16 and 17 shown the best and worst neutron spectra unfolded at final testing stage of the designed ANN compared with the target neutron spectra, along with the correlation and chi sqare tests applied to each spectra.



(a) Best neutron spectra

(b) Correlation test

Figure 16. RDANN: Best neutron spectra and correlation test



(a) Worst neutron spectra

(b) Correlation test

Figure 17. RDANN: Worst neutron spectra and correlation test

In ANNs design, the use of the RDANN methodology can help provide answers to the following critical design and construction issues:

- **What is the proper density for training samples in the input space?**. The proper density for training samples in the input space was: 80% for ANN training stage and 20% for testing stage.
- **When is the best time to stop training to avoid over-fitting?**. The best time to stop training to avoid over-fitting is variable and depends of the proper selection of the ANN parameters. In the optimum ANN designed, the best time to train the network avoiding the over-fitting was 120 seconds average.
- **Which is the best architecture to use**?. The best architecture to use is 7 : 14 : 31, a learning rate = 0.1 and a momentum = 0.1, a *trainscg* training algorithm and an mse = $1E^{-4}$.
- **Is it better to use a large architecture and stop training optimally or to use an optimum architecture, which probably will not over-fit the data, but may require more time to train?**. It is better to use an optimum architecture, designed with the RDANN methodology, which not overfit the data and do not require more training time instead of using a large architecture stopping the training over the time or trials which produce a poor ANN.
- **If noise is present in the training data, is best to reduce the amount of noise or gather additional data?, and what is the effect of noise in the testing data on the performance of the network?**. In the random weight initialization is introduced a great amount of noise in training data. Such initialization introduces large negative numbers which is very harmful for the neutron spectra unfolded. The effect of noise in the random weight initialization in the testing data, affects significantly the performance of the network. In this case, the noise produced results negatives in the unfolded neutrons, which has not physics meaning. In consequence, can be concluded that it is necessary to reduce the noise introduced in the random weight initialization.

4. Conclusions

ANNs is a theory that still is in development process; its true potentiality has not still been reached; although researchers have developed potent learning algorithms of great practical value, representations and procedures that the brain is served, are even unknown. The integration of ANN and optimization provides a tool for designing neural network parameters and improving the network performance. In this work, a systematic, methodological and experimental approach called RDANN methdology was introduced to obtain the optimum design of artificial neural networks. The Taguchi method is the main technique used to simplify the optimization problem.

RDANN methdology was applied with success in nuclear sciences to solve the neutron spectra unfolding problem. The factors that are found to be significant in the case study were number of hidden neurons in hidden layer 1 and 2, learning rate and momentum term. The near optimum ANN topology was: 7:14:31 whit a momentum = 0.1 and a learning rate = 0.1, mse = $1E^{-4}$ and a *"trainscg"* learning function. The optimal net architecture was designed in short time and has high performance and generalization capability.

The proposed systematic and experimental approach is a useful alternative for the robust design of ANNs. It offers a convenient way of simultaneously considering design and noise variables, and incorporates the concept of robustness in the ANN design process. The computer program developed to implement the experimental and confirmation stages of the RDANN methodology, reduces significantly the time required to prepare, to process and to present the information in an appropriate way to de designer, and in the search of the optimal net topology being designed. This gives to the researcher time to solve the problem in which he is interested.

The results show that RDANN methodolgy can be used to find better setting of ANNs, which not only results in minimum error, but also significantly reduces training time and effort in the modeling phases.

The optimum setting of ANNS parameters are largely problem-dependent. Ideally and optimization process should be performed for each ANNs application, as the significant factors might be different for ANNs trained for different purpose.

When compared with the trial-and-error approach, which can spent from several days to months to prove different ANN architectures and parameters which may lead to a poor overall ANN design, RDANN methodology reduces significantly the time spent in determining the optimum ANN architecture. With RDANN it takes from minutes to a couple of hours to determine the best and robust ANN architectural and learning parameters allowing to researches more time to solve the problem in question.

Acknowledgements

This work was partially supported by Fondos Mixtos CONACYT - Gobierno del Estado de Zacatecas (México) under contract ZAC-2011-C01-168387.

This work was partially supported by PROMEP under contract PROMEP/103.5/12/3603.

Author details

José Manuel Ortiz-Rodríguez[1,⋆],
Ma. del Rosario Martínez-Blanco[2],
José Manuel Cervantes Viramontes[1] and
Héctor René Vega-Carrillo[2]

⋆ Address all correspondence to: morvymm@yahoo.com.mx

Universidad Autónoma de Zacatecas, Unidades Académicas, 1-Ingeniería Eléctrica, 2-Estudios Nucleares, México

5. References

[1] C.R. Alavala. *Fuzzy logic and neural networks basic concepts & applications*. New Age International Publishers, 1996.

[2] J. Lakhmi and A. M. Fanelli. *Recent advances in artificial neural networks design and applications*. CRC Press, 2000.

[3] R. Correa and I. Requena. Aplicaciones de redes neuronales artificiales en la ingeniería nuclear. *XII congreso Español sobre Tecnologías y Lógica Fuzzy*, 1:485–490, 2004.

[4] G. Dreyfus. *Neural networks, methodology and applications*. Springer, 2005.

[5] B. Apolloni, S. Bassis, and M. Marinaro. *New directions in neural networks*. IOS Press, 2009.

[6] J. Zupan. Introduction to artificial neural network methods: what they are and how to use them. *Acta Chimica Slovenica*, 41(3):327–352, month 1994.

[7] A. K. Jain, J. Mao, and K. M. Mohiuddin. Artificial neural networks: a tutorial. *IEEE: Computer*, 29(3):31–44, month 1996.

[8] T. Y. Lin and C. H. Tseng. Optimum design for artifcial neural networks: an example in a bicycle derailleur system", journal = "engineering applications of artificial inteligence. 13:3–14, 2000.

[9] M.M. Gupta, L. Jin, and N. Homma. *Static and dynamic neural networks: from fundamentals to advanced theory*. 2003.

[10] D. Graupe. *Principles of artificial neural networks*. World Scientific, 2007.

[11] M. Kishan, K. Chilukuri, and R. Sanjay. *Elements of artificial neural networks*. The MIT Press, 2000.

[12] L. Fausett. *Fundamentals of neural networks, architectures, algorithms and applications*. Prentice Hall, 1993.

[13] A.I. Galushkin. *Neural networks theory*. Springer, 2007.

[14] J.A. Frenie and A. Jiju. Teaching the taguchi method to industrial engineers. *MCB University Press*, 50(4):141–149, 2001.

[15] S. C. Tam, W. L. Chen, Y. H. Chen, and H. Y. Zheng. Application of taguchi method in the optimization of laser micro-engraving of photomasks. *International Journal of Materials & Product Technology*, 11(3-4):333–344, 1996.

[16] G. E. Peterson, D. C. St. Clair, S. R. Aylward, and W. E. Bond. Using taguchi's method of experimental design to control errors in layered perceptrons. *IEEE Transactions on Neural Networks*, 6(4):949–961, month 1995.

[17] Y.K. Singh. *Fundamentl of research methodology and statistics*. New Age International Publishers, 2006.

[18] M.N. Shyam. Robust design. Technical report, 2002.

[19] T.T. Soong. *Fundamentals of probability and statistics for engineers*. John Wiley & Sons, Inc., 2004.

[20] M. S. Packianather, P. R. Drake, and Rowlands H. Optimizing the parameters of multilayered feedforward neural networks through taguchi design of experiments. *Quality and Reliability Engineering International*, 16:461–473, month 2000.

[21] M. S. Packianather and P. R. Drake. Modelling neural network performance through response surface methodology for classifying wood veneer defects. *Proceedings of the Institution of Mechanical Engineers, Part B*, 218(4):459–466, month 2004.

[22] M.A. Arbib. *Brain theory and neural networks*. The Mit Press, 2003.

[23] M. H. Beale, M. T Hagan, and H. B. Demuth. *Neural networks toolbox, user's guide*. Mathworks, 1992. www.mathworks.com/help/pdf_doc/nnet/nnet.pdf.

[24] N. K. Kasabov. *Foundations of neural networs, fuzzy systems, and knowledge engineering*. MIT Press, 1998.

[25] M.R. Kardan, R. Koohi-Fayegh, S. Setayeshi, and M. Ghiassi-Nejad. Neutron spectra unfolding in Bonner spheres spectrometry using neural networks. *Radiation Protection Dosimetry*, 104(1):27–30, 2004.

[26] H. R. Vega-Carrillo, V. M. Hernández-Dávila, E. Manzanares-Acuña, G. A. Mercado Sánchez, E. Gallego, A. Lorente, W. A. Perales-Muñoz, and J. A. Robles-Rodríguez. Artificial neural networks in neutron dosimetry. *Radiation Protection Dosimetry*, 118(3):251–259, month 2005.

[27] H. R. Vega-Carrillo, V. M. Hernández-Dávila, E. Manzanares-Acuña, G. A. Mercado-Sánchez, M. P. Iñiguez de la Torre, R. Barquero, S. Preciado-Flores, R. Méndez-Villafañe, T. Arteaga-Arteaga, and J. M. Ortiz-Rodríguez. Neutron spectrometry using artificial neural networks. *Radiation Measurements*, 41:425–431, month 2006.

[28] H. R. Vega-Carrillo, M. R. Martínez-Blanco, V. M. Hernández-Dávila, and J. M. Ortiz-Rodríguez. Ann in spectroscopy and neutron dosimetry. *Journal of Radioanalytical and Nuclear Chemistry*, 281(3):615–618, month 2009a.

[29] H.R. Vega-Carrillo, J.M. Ortiz-Rodríguez, M.R. Martínez-Blanco, and V.M. Hernández-Dávila. Ann in spectrometry and neutron dosimetry. *American Institute of Physics Proccedings*, 1310:12–17, 2010.

[30] H. R. Vega-Carrillo, J. M. Ortiz-Rodríguez, V. M. Hernández-Dávila, M .R. Martínez-Blanco, B. Hernández-Almaraz, A. Ortiz-Hernández, and G. A. Mercado. Different spectra with the same neutron source. *Revista Mexicana de Física S*, 56(1):35–39, month 2009b.

[31] H.R. Vega-Carrillo, M.R. Martínez-Blanco, V.M. Hernández-Dávila, and J.M. Ortiz-Rodríguez. Spectra and dose with ANN of 252Cf, 241Am-Be, and 239Pu-Be. *Journal of Radioanalytical and Nuclear Chemistry*, 281(3):615–618, 2009.

[32] R.R. Roy and B.P. Nigam. *Nuclear physics, theory and experiment*. John Wiley & Sons, Inc., 1967.

[33] R.N. Cahn and G. Goldhaber. *The experimental fundations of particle physics*. Cambridge University Press, 2009.

[34] R.L. Murray. *Nuclear energy, an introduction to the concepts, systems, and applications of nuclear processes*. World Scientific, 2000.

[35] D. J. Thomas. Neutron spectrometry for radiation protection. *Radiation Protection Dosimetry*, 110(1-4):141–149, month 2004.

[36] B.R.L. Siebert, J.C. McDonald, and W.G. Alberts. Neutron spectrometry for radiation protection purposes. *Nuclear Instruments and Methods in Physics Research A*, 476(1-2):347–352, 2002.

[37] P. Reuss. *Neutron physics*. EDP Sciences, 2008.

[38] F. D. Brooks and H. Klein. Neutron spectrometry, historical review and present status. *Nuclear Instruments and Methods in Physics Research A*, 476:1–11, month 2002.

[39] M. Reginatto. What can we learn about the spectrum of high-energy stray neutorn fields from bonner sphere measurements. *Radiation Measurements*, 44:692–699, 2009.

[40] M. Awschalom and R.S. Sanna. Applications of bonner sphere detectors in neutron field dosimetry. *Radiation Protection Dosimetry*, 10(1-4):89–101, 1985.

[41] V. Vylet. Response matrix of an extended bonner sphere system. *Nuclear Instruments and Methods in Physics Research A*, 476:26–30, month 2002.

[42] V. Lacoste, V. Gressier, J. L. Pochat, F. Fernández, M. Bakali, and T. Bouassoule. Characterization of bonner sphere systems at monoenergetic and thermal neutron fields. *Radiation Protection Dosimetry*, 110(1-4):529–532, month 2004.

[43] M. Matzke. Unfolding procedures. *Radiation Protection Dosimetry*, 107(1-3):155–174, month 2003.

[44] M. El Messaoudi, A. Chouak, M. Lferde, and R. Cherkaoui. Performance of three different unfolding procedures connected to bonner sphere data. *Radiation Protection Dosimetry*, 108(3):247–253, month 2004.

[45] R.B. Murray. Use of 6LiI(Eu) as a scintillation detector and spectrometer for fast neutrons. *Nuclear Instruments*, 2:237–248, 1957.

[46] V. Lacoste, M. Reginatto, B. Asselineau, and H. Muller. Bonner sphere neutron spectrometry at nuclear workplaces in the framework of the evidos project. *Radiation Protection Dosimetry*, 125(1-4):304–308, month 2007.

[47] H. Mazrou, T. Sidahmed, Z. Idiri, Z. Lounis-Mokrani, , Z. Bedek, and M. Allab. Characterization of the crna bonner sphere spectrometer based on 6lii scintillator exposed to an 241ambe neutron source. *Radiation Measurements*, 43:1095–1099, month 2008.

[48] H. R. Vega-Carrillo, V. M. Hernández-Dávila, E. Manzanares-Acuña, E. Gallego, A. Lorente, and M. P. Iñiguez. Artificial neural networks technology for neutron spectrometry and dosimetry. *Radiation Protection Dosimetry*, 126(1-4):408–412, month 2007b.

[49] IAEA. Compendium of neutron spectra and detector responses for radiation protection purposes. Technical Report 403, 2001.

[50] M. P. Iñiguez de la Torre and H. R. Vega Carrillo. Catalogue to select the initial guess spectrum during unfolding. *Nuclear Instruments and Methods in Physics Research A*, 476(1):270–273, month 2002.