Supervised SOM Based Architecture versus Multilayer Perceptron and RBF Networks

David Gil
Computing Technology and Data Processing
University of Alicante, Spain

Magnus Johnsson
Lund University Cognitive Science
Lund, Sweden

April 21, 2010

Abstract

We address a contrastive study between the well known Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) neural networks and a SOM based supervised architecture in a number of data classification tasks. Well known databases like Breast Cancer, Parkinson and Iris were used to evaluate the three architectures by constructing confusion matrices. The results are encouraging and indicate that the SOM based supervised architecture generally achieves results as good as the MLP and slightly higher on some measures than the RBF network.

1 Introduction

The use of classifer systems in many areas is increasing gradually. Recent advances in the field of artificial intelligence have led to the emergence of expert systems and Decision Support Systems (DSSs) for economics, linguistics, management science, mathematical modelling, psychology, etc. Artificial Neural Networks (ANNs) have been utilized for improving the classification tasks because of its property called black-box learning. In fact, they are one of the popular methods for classification problems [13] [12].

Compared to most traditional classification approaches, ANNs are nonlinear, nonparametric, and adaptive. They can theoretically approximate any fundamental relationship with arbitrary accuracy. They are ideally suitable for problems where observations are easy to obtain but the data structure or underlying relationship is unknown.

Although there are different types of learning techniques, this paper proposes the study of supervised learning. The learning system may label (classification) a set of vectors choosing one between several categories (classes). There are several types of classifiers that have been used with different degrees of accuracy [28] [21] [20].

Some of our related work in the field of the diagnosis has been developed basically by means of Artificial Neural Networks (ANNs) [9] [10].

The aim of this paper is to test a SOM based supervised architecture ANN and compare it in classification tasks with two other ANNs known as the Multi-Layer Perceptron (MLP) and the Radial Basis Function (RBF) network.

We have compared the SOM based supervised architecture with the MLP and the RBF networks as the MLP and RBF neural networks are the two most widely used within the field of task classification. Moreover, the MLP is purely supervised while the RBF is also a hybrid network with a part of unsupervised learning. Certain similar characteristics of the SOM based supervised architecture and the RBF will allow allow the acquisition of good measures of the efficiency of our network.

The remaining part of the paper is organized as follows: First we give a brief description of the networks MLP, RBF and a longer description of the SOM Based Supervised Architecture; Then we proceed by describing the design of our proposal and the training of the ANNs by the available data; Then we continue by describing the subsequent testing carried out in order to analyse the results; Finally we draw the relevant conclusions.

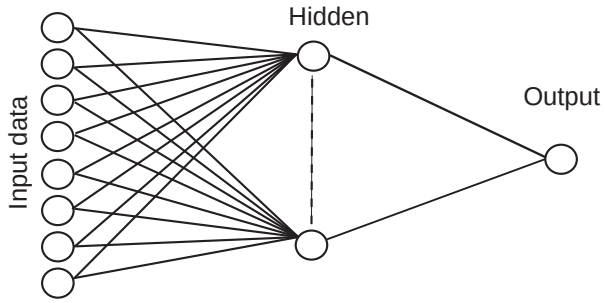

Figure 1: The architecture of the MLP network (input layer, hidden layer and output layer). The input layer represents the input data (the input data is described in section 4.1). The usage of a hidden layer enables the representation of data sets that are not linearly separable. The output layer represents the classification result and it contains as many outputs as the problem has classes, although here only one neuron is shown. The weights and the threshold of the MLP are calculated during an adaptation process.

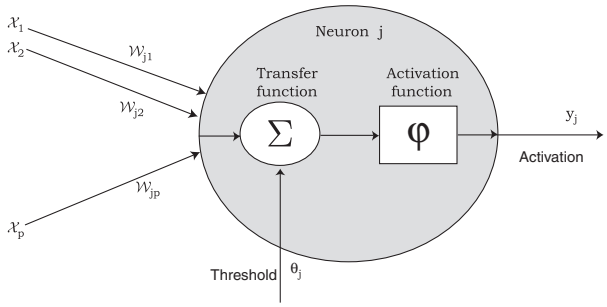

Figure 2: A neuron in the hidden or the output layer in the MLP. In the experimentation section the number of hidden neurons of the MLP will be established.

2 Supervised Architectures

2.1 Multilayer Perceptron

In this study we have used a Multi-Layer Perceptron (MLP) network [24] [11][5]. A typical MLP consists of three layers of neurons: an input layer that receives external inputs, one hidden layer, and an output layer which generates the classification results (see figure 1). Note that unlike other layers, no computation is involved in the input layer. The principle of the network is that when data are presented at the input layer, the network neurons run calculations in the consecutive layers until an output value is obtained at each of the output neurons. This output will indicate the appropriate class for the input data.

Each neuron (see figure 2) in the input and the hidden layers is connected to all neurons in the next layer by weighted connections. The neurons of the hidden layers (see figure 2) compute weighted sums of their inputs and adds a threshold. The resulting sums are used to calculate the activity of the neurons by applying a sigmoid activation function.

This process is defined as follows:

$$\nu_j = \sum_{i=1}^{p} w_{ji} x_i + \theta_j \quad , \qquad y_j = f_j(\nu_j) \qquad (1)$$

where ν_j is the linear combination of inputs $x_1, x_2, ..., x_p$, and the threshold θ_j , w_{ji} is the connection weight between the input x_i and the neuron j, and f_j is the activation function of the j_{th} neuron, and y_j is the output. The sigmoid function is a common choice of activation function. It is defined as:

$$f(t) = \frac{1}{1 + e^{-t}} \qquad (2)$$

A single neuron in the MLP is able to linearly separate its input space into two subspaces by a hyperplane defined by the weights and the treshold. The weights define the direction of this hyperplane whereas the threshold term θ_j offsets it from origo.

The MLP network uses the backpropagation algorithm [25], which is a gradient descent method, for the adaptation of the weights (the backpropagation training parameters are showed in Table 1). This algorithm runs as follows:

All the weight vectors w are initialized with small random values from a pseudorandom sequence generator. Then and until the convergence (i.e. when the error E is below a preset value) we repeat the three basic steps:

- The weight vectors w_i are updated by

$$w(t+1) = w(t) + \Delta w(t) \qquad (3)$$

- where

$$\Delta w(t) = -h\partial E(t)/\partial w \tag{4}$$

- Compute the error E(t+1).

where t is the iteration number, w is the weight vector, and h is the learning rate.

The backpropagation MLP is a supervised ANN. This means the network is presented with input examples as well as the corresponding desired output. The backpropagation algorithm adapts the weights and the thresholds of the neurons in a way that minimizes the error function E

$$E = \frac{1}{2}\sum_{p=1}^{n}(d_p - y_p)^2 \tag{5}$$

where y_p is the actual output and d_p the desired output for input pattern p.

The minimization of E can be accomplished by gradient descent, i.e. the weights are adjusted to change the value of E in the direction of its negative gradient. The exact updating rules can be calculated by applying derivatives and the chain rule (for the weights between the input and the hidden layer).

2.2 Radial Basis Function Network

In this section, the basic concepts of the Radial Basis Function (RBF) network are described. Radial Basis Function Networks are a type of ANN where the hidden layer is composed of radial-basis functions which are similar to normal distribution curves.

The RBF neural network [3] is generally composed of three layers: input layer, hidden layer and output layer. The input layer feeds the input data to each of the neurons of the hidden layer. The hidden layer differs greatly from other neural networks in that each neuron represents a data cluster with a given radius and which is centred at a particular point in the input space. Each neuron in the hidden layer calculates the distance from the input vector to its own center. The calculated distance is transformed via some basis function and the result is the output from the neuron. The output from the neuron is multiplied by a constant or weighting value and fed into the output layer. The output layer consists of as many classes or outputs as the problem has. It acts to sum the outputs of the previous layer and to yield a final output value [18] [7]. A generic architecture of an RBF network with p input and n hidden neurons is illustrated in figure 3, where x_i are data points, $\varphi(||x - x^i||)$ are the RBFs, x^i are the centres of the basis functions, and w_i are the weights. A very common RBF is the Gaussian RBF:

$$\varphi(||x - x^i||) = exp(-\frac{||x - x^i||^2}{2\sigma_i^2}), i = 1, 2, ..., N \tag{6}$$

The activity $F(X)$ of the output neuron is given by:

$$F(X) = \sum_{j=1}^{N} w_j \varphi(||x - x^i||) \tag{7}$$

The learning process used in the RBF network is done in two phases thus calculating the parameters of the hidden layer and output layer (note that the input layer does not perform calculations at all). In this way, we can speed up the learning process considerably compared to the MLP backpropagation. First of all, for the hidden layer, we calculate the number of centres of the basis functions or centroids (figure 3 and 4), using the K-means algorithm. The number of these centroids depend on each case the problem addressed. Table 2 and the explanation in the experimental section help to understand this process. In the second phase we proceed to the training of the output neurons which is an easy task since we know the values. The calculation is then done by simply applying the equation 7.

2.3 SOM Based Supervised Architecture

This section proposes a "Supervised SOM Based Architecture" (see Figure 5). The hidden layer of an ANN is one of the most complex parts to design in an artificial neural network. Here we propose to use a Self-Organizing Map (SOM) as a hidden layer. Some previous works where similar models are implemented are [27] [6] [23].

In [23], Piela represents a very exciting research where a modern approach to imputation is being discussed. It is said that many traditional methods

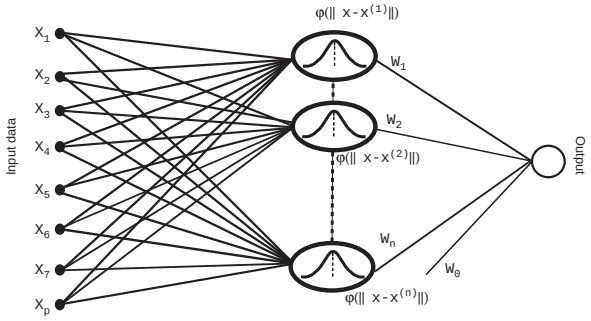

Figure 3: The architecture of the RBF network (input layer, hidden layer and output layer). The input layer represents the input data (the input data is described in the experiment section). The output neuron activity is reflected by equation 7. As in the case of the MLP the output layer of the RBF network consists of as many neurons as the problem has classes, although here only one neuron is shown.

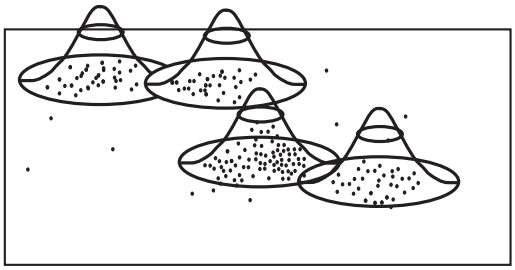

Figure 4: The radial basis functions have a local character since they are functions that reach a level close to their maximum when the input pattern X(n) is close to the centre of the neuron. When the pattern moves away from the centre, the function value is tending to the minimum value. The outputs of the radial basis neural networks are therefore a Gaussian linear combination, where each of the terms in the linear combination is activated for a particular portion of space defined by the input patterns.

of imputation use some kind of classification trying to get observations with missing values into as homogenous groups as possible. SOM is an iterative method for classification and can thus also be used in finding the imputation classes. Therefore, imputations are made within clusters in several ways which can be based on both traditional and neural methods. The main emphasis of this approach is to aid methodological development of knowledge discovery, data analysis, and modelling in general.

The SOM based architecture discussed in this paper consists of two layers (actually two separate but connected neural networks), i.e. a hidden layer and an output layer. The hidden layer consists of a Self-Organizing Map (SOM) [15] which is fully connected with forward connections to the output layer. The output layer consists of a grid of neurons that are adapted by the delta rule to get an activity that converges to the provided desired output.

2.3.1 The Hidden Layer

The hidden layer consists of a version of the SOM that works as follows. It consists of an $I \times J$ grid of a fixed number of neurons and a fixed topology. Each neuron n_{ij} is associated with a weight vector $w_{ij} \in R^n$. All the elements of the weight vectors are initialized by real numbers randomly selected from a uniform distribution between 0 and 1, after which all the weight vectors are normalized, i.e. turned into unit vectors.

At time t each neuron n_{ij} receives an input vector $x(t) \in R^n$.

The net input s_{ij} is calculated using the standard cosine metric

$$s_{ij}(t) = \frac{x(t) \cdot w_{ij}(t)}{||x(t)||||w_{ij}(t)||}, \tag{8}$$

The activity y_{ij} in the neuron n_{ij} is calculated by using the softmax function [4]

$$y_{ij}(t) = \frac{(s_{ij}(t))^m}{\max_{uv}(s_{uv}(t))^m} \tag{9}$$

where u and v ranges over the rows and the columns of the neural network and m is the softmax exponent.
The neuron c associated with the weight vector $w_c(t)$ most similar to the input vector $x(t)$, i.e. the neuron with the strongest activation, is selected:

$$c = \arg\max_c\{|x(t) \cdot w_c(t)|\} \tag{10}$$

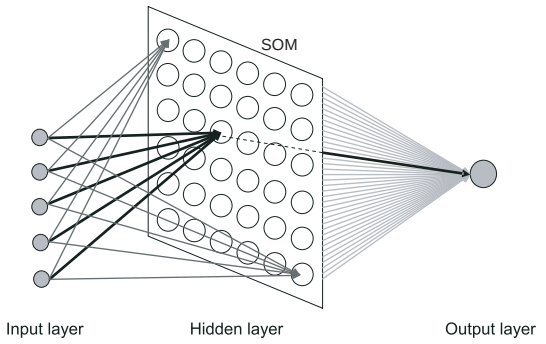

Figure 5: Supervised SOM Based Architecture. For simplicity reasons, the output layer is shown with only one neuron as in the case of the MLP and the RBF networks. However, in reality we use as many neurons in the output layer as the problem has classes. For example, one of the databases in our experiments section, the iris database, has 3 classes and in that case the output layer therefore has three neurons.

The weights w_{ijk} are adapted by

$$w_{ijk}(t+1) = w_{ijk}(t) + \alpha(t)G_{ijc}(t)\left[x_k(t) - w_{ijk}(t)\right] \quad (11)$$

where $0 \leq \alpha(t) \leq 1$ is the adaptation strength with $\alpha(t) \to 0$ when $t \to \infty$.

The neighbourhood function is:

$$G_{ijc}(t) = e^{-\frac{||r_c - r_{ij}||}{2\sigma^2(t)}} \quad (12)$$

where $r_c \in R^2$ and $r_{ij} \in R^2$ are location vectors of neurons c and n_{ij}, G is a Gaussian function decreasing with time. σ is the neighbourhood radius which at time t is updated by multiplying σ at time $t-1$ with 0.99 as it is showed in table 3.

All weights $w_{ijk}(t)$ are normalized after each adaptation.

2.3.2 The Output Layer

The output layer consists of an $I \times J$ grid of a fixed number of neurons and a fixed topology. Each neuron n_{ij} is associated with a weight vector $w_{ij} \in R^n$. All the elements of the weight vector are initialized by real numbers randomly selected from a uniform distribution between 0 and 1, after which the weight vector is normalized, i.e. turned into unit vectors.

At time t each neuron n_{ij} receives an input vector $x(t) \in R^n$.

The activity y_{ij} in the neuron n_{ij} is calculated using the standard cosine metric

$$y_{ij}(t) = \frac{x(t) \cdot w_{ij}(t)}{||x(t)||||w_{ij}(t)||}, \quad (13)$$

During the learning phase the weights w_{ijl}, are adapted by

$$w_{ijl}(t+1) = w_{ijl}(t) + \beta x_l(t)\left[d_{ij}(t) - y_{ij}(t)\right] \quad (14)$$

where β is the adaptation strength and $d_{ij}(t)$ is the desired activity for the neuron n_{ij}.

3 Experimentation

3.1 Methods

The databases used in this study are Breast Cancer, Parkinson and Iris. These databases are taken from the University of California at Irvine (UCI) machine learning repository [1] [17] [16] and are used for training and testing in the experiments. The main reason to use these particular datasets is that they are well known to professionals of artificial intelligence.

We have used Matlab and in particular the Neural Network toolbox for our experimentation with MLP and RBF. The reason for using matlab is due to the wide scope of problems addressed and the effectiveness conducted with these [19] [26] [8]. The SOM Based Supervised Architecture has been implemented under Ikaros [2].

The method to evaluate the three methods is to obtain some measures as classification accuracy, sensitivity, specificity, positive predictive value, negative predictive value and a confusion matrix. A confusion matrix [14] contains information about actual and predicted classifications done by a classification system.

Moreover, we have also evaluated the learning process regarding how fast every method learns.

For the construction of the MLP architecture we have proceeded as follows:

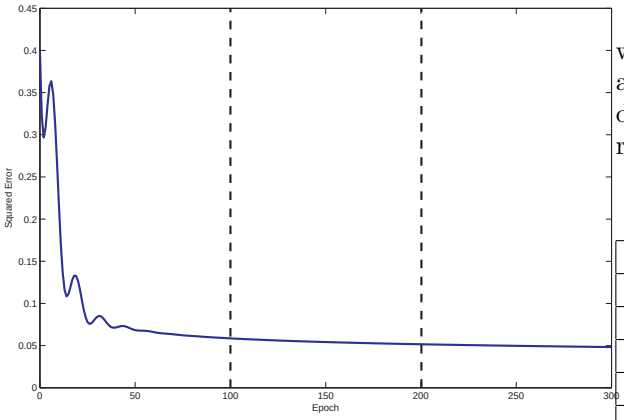

Figure 6: Learning speed in the MLP is slow since it uses the backpropagation method. Generally the backpropagation method always provides a very high precision. The drawbacks are the slowness in learning as well as the risk of over fitting the data learned.

a) Layer 1 corresponds directly to the input vector, that is, all the parameters/fields of the input record.
b) Layer 2 (the hidden layer). The number of hidden neurons for this layer is the most elaborated question in the network's architecture. This number represents a trade off between performance and the risk of over fitting. In fact, the number of neurons in a hidden layer will significantly influence the ability of the network to generalize from the training data to unknown examples [22]. By doing some experiments we discovered that:

- With a low number of neurons for this layer the training and test sets performed badly;
- With a high number of neurons the training set performed good. However there is a high risk of over fitting;
- The optimal solution for this layer has been found to be 24 neurons for Breast Cancer, 12 neurons for Parkinson and 5 neurons for Iris.

c) Layer 3 (the output layer) (Classification). It has two outputs for Breast Cancer and Parkinson and three outputs for Iris.

The learning algorithm used is backpropagation with adaptive learning rate, constant momentum and an optimized algorithm based on the gradient descent method. The backpropagation training parameters are showed in table 1.

Table 1: Backpropagation training parameters.

Parameters	Value
Learning rate	0.01
Adaptive learning rate	0.1
Constant momentum	0.2
Epochs	100-1000-10000
Minimum performance gradient	$\frac{1}{e^{-10}}$

The main parameter we must adjust in order to get a good accuracy with an RBF network is the maximum number of centres. This is a parameter of the center selection algorithm, and is the maximum number of centers/RBFs that is chosen.

We followed the recommendation to set an upper limit between 60% and 70% for the proportion between the number of RBFs and the number of neurons in the input layer. The parameter spread, which is the spread of radial basis functions, helps to construct the hidden layer. The larger the spread is, the smoother the function approximation will be. Spread value represents a compromise between a low value which means low accuracy and a high value with over fitting risk and the possibility that the network may not generalize well.

The RBF network training parameters are shown in table 2.

Table 2: RBF network training parameters.

Parameters	Value
Learning rate	0.01
Adaptive learning rate	0.1
Spread	0.8
Epochs	100-1000-10000
Minimum performance gradient	$\frac{1}{e^{-10}}$

The SOM Based Supervised Architecture training parameters are showed in table 3.

Frequently, the complete data set is divided into two subsets: the training set and the test set. Here, the training set is used to determine the system parameters, and the test set is used to evaluate the diagnosis accuracy and the network general-

Table 3: SOM Based Supervised Architecture.

Parameters	Value
Learning rate	0.1
Learning rate Decay	0.99
Learning rate Minimum	0.01
Neighbourhood Radius (σ)	15
Neighbourhood Decay	0.99
Neighbourhood Minimum	1
Adaptation Strength	0.35
Epochs	100-1000-10000
Minimum performance gradient	$\frac{1}{e^{-10}}$

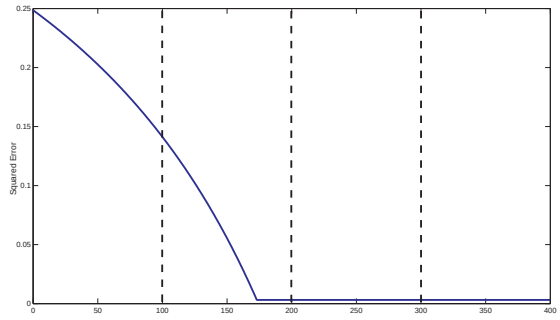

Figure 7: Learning speed in RBF is faster than in the MLP. This is because the RBF network already has some centroids defined due to predefined training which saves time.

ization. Cross-validation has been widely used to assess the generalization of a network. The cross-validation estimate of accuracy determining by the overall number of correct classifications divided by the total number of examples in the dataset.

$$Acc_{cv} = \frac{1}{n} \sum_{x_i \in S} \delta(I(S_i, x_i), y_i) \qquad (15)$$

where n is the size of the dataset S, x_i is the example of S, y_i is the target of x_i, and S_i is the probable target of x_i by the classifier. Therefore:

$$\delta(i,j) = \begin{cases} 1 & if \quad i \in N_c(t) \\ 0 & otherwise \end{cases} \qquad (16)$$

Specifically, for this study we have applied a five-fold cross-validation method for the performance assessment of every network. The data has been divided in five sets (S1, S2, S3, S4, S5) and

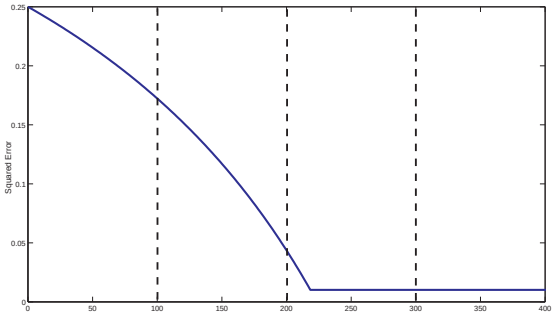

Figure 8: Learning speed in the SOM Based Supervised Architecture is similar as for the RBF network and it is faster than the learning speed of the MLP.

the five experiments performed were:
Experiment 1 - Training: S1, S2, S3, S4; Test: S5
Experiment 2 - Training: S1, S2, S3, S5; Test: S4
Experiment 3 - Training: S1, S2, S4, S5; Test: S3
Experiment 4 - Training: S1, S3, S4, S5; Test: S2
Experiment 5 - Training: S2, S3, S4, S5; Test: S1

The sets of data used for the process of constructing the model (the training data) were of 565, 195 and 150 registers for Breast Cancer, Parkinson and Iris respectively. The other set of data used to validate the model (the test data) was of 113, 39 and 30 registers also for Breast Cancer, Parkinson and Iris respectively. The test data are chosen randomly from the initial data and the remaining data form the training data. The method is called 5-fold cross validation since this process has been performed five times. The function approximation fits a function using the training set only. Then the function approximation is asked to predict the output values for the data in the testing set. The errors it makes are accumulated to provide the mean absolute test set error, which is used to evaluate the model. The results are presented using confusion matrices.

3.2 Results

Table 4 shows the confusion matrix for all the classifiers with a two classes problem: Breast Cancer database. Classification accuracy, sensitivity, specificity, positive predictive value and negative predictive value can be defined (all the equations 17-21 show 5 values for MLP, RBF and

the SOM based architecture respectively) by using the elements of the confusion matrix (table 4).

Table 4: Definition of the confusion matrix with the value for every measure for the MLP, the RBF and the SOM Based Supervised Architecture classifiers with the Breast Cancer database. It has two classes: Possitive (P) and Negative (N). True positive (TP); False negative (FN); False positive (FP); True negative (TN)

	MLP			RBF			SOM	
Act	Predicted			Predicted			Predicted	
	P	N		P	N		P	N
P	184	7		151	56		165	35
	(TP)	(FN)						
N	13	296		46	247		30	268
N	(FP)	(TN)						

$$Classification\quad accuracy(\%) = \frac{TP+TN}{TP+FP+FN+TN}\mathtt{x}100 = 96\%,\ 79.6\%,\ 86.9\% \tag{17}$$

$$Sensitivity(\%) = \frac{TP}{TP+FN}\mathtt{x}100 = 96.3\%,\ 72.9\%,\ 82.5\% \tag{18}$$

$$Specificity(\%) = \frac{TN}{FP+TN}\mathtt{x}100 = 95.8\%,\ 84.3\%,\ 89.9\% \tag{19}$$

$$Positive\quad predictive\quad value(\%) = \frac{TP}{TP+FP}\mathtt{x}100 = 93.4\%,\ 76.6\%,\ 84.6\% \tag{20}$$

$$Negative\quad predictive\quad value(\%) = \frac{TN}{FN+TN}\mathtt{x}100 = 97.7\%,\ 81.5\%,\ 88.4\% \tag{21}$$

Table 5 shows the confusion matrix for all the classifiers with a two classes problem: Parkinson database. Classification accuracy, sensitivity, specificity, positive predictive value and negative predictive value can be defined (all the equations 22-26 show 5 values for MLP, RBF and SOM respectively) by using the elements of the confusion matrix (table 5).

$$Classification\quad accuracy(\%) = \frac{TP+TN}{TP+FP+FN+TN}\mathtt{x}100 = 84.6\%,\ 82.1\%,\ 81.5\% \tag{22}$$

Table 5: Definition of the confusion matrix with the value for every measure for the MLP, RBF and SOM classifiers with the Parkinson database. It has two classes: Possitive (P) and Negative (N). True positive (TP); False negative (FN); False positive (FP); True negative (TN)

	MLP			RBF			SOM	
Actual	Predicted			Predicted			Predicted	
	P	N		P	N		P	N
P	132	15		147	35		129	18
N	15	33		0	13		18	30

$$Sensitivity(\%) = \frac{TP}{TP+FN}\mathtt{x}100 = 89.8\%,\ 80.8\%,\ 87.8\% \tag{23}$$

$$Specificity(\%) = \frac{TN}{FP+TN}\mathtt{x}100 = 68.8\%,\ 100\%,\ 62.5\% \tag{24}$$

$$Positive\quad predictive\quad value(\%) = \frac{TP}{TP+FP}\mathtt{x}100 = 89.8\%,\ 100\%,\ 87.8\% \tag{25}$$

$$Negative\quad predictive\quad value(\%) = \frac{TN}{FN+TN}\mathtt{x}100 = 68.8\%,\ 27.1\%,\ 62.5\% \tag{26}$$

Table 6 shows the confusion matrix for all the classifiers with a three classes problem: Iris database. Since it is a different classification problem from the two previous examples, with three output, we only show the equation of Classification accuracy (with 3 values for MLP, RBF and SOM respectively) by using the elements of the confusion matrix (table 6).

$$Classification\quad accuracy(\%) = \frac{TP+TN}{TP+FP+FN+TN}\mathtt{x}100 = 91.7\%,\ 73.3\%,\ 87.6\% \tag{27}$$

4 Conclusion

In this paper we have evaluated the performance of the SOM based supervised architecture. To evaluate the effectiveness of this ANN architecture, we compare it with MLP and RBF networks in classification tasks. The supervised SOM based architecture has similar characteristics as the RBF network,

Table 6: Definition of the confusion matrix with the value for every measure for the MLP, RBF and SOM classifiers with the Iris database. It has three classes: Iris-virginica (A), Iris-versicolor (B) and Iris-setosa (C)

	MLP			RBF			SOM		
	Predicted			Predicted			Predicted		
	A	B	C	A	B	C	A	B	C
A	41	6	0	37	3	10	41	7	0
B	6	39	0	9	42	9	9	43	0
C	3	0	50	4	5	31	0	0	50

and could indeed be seen as an RBF network that automatically finds a suitable number of and suitable locations of RBFs in its hidden layer. Thus a fundamental aspect of this ANN architecture is the use of a SOM as hidden layer. An important aspect of the SOM based architecture is that it helps the designer to get rid of the difficulty and cost of the design of the hidden layers.

The results presented by these three methods (MLP, RBF and SOM based supervised architecture) achieve a high precision level of the confusion matrix regarding the different measurement parameters (accuracy, sensitivity, specificity, positive predictive value and negative predictive value).

With the Breast Cancer database the accuracy of the MLP, RBF and SOM based supervised architecture were very good, especially MLP which showed a high degree of certainty of 96%.

The SOM based architecture accomplish better results than the RBF network. This can be observed not only in the values of the classification accuracy but also in the rest of them. In the case of sensitivity there was a difference of around 10% between the RBF network and the SOM based supervised architecture.

These results as well as those with the Parkinson's and the iris databases are very encouraging because the SOM based supervised architecture is usually better than the RBF and even if its accuracy is a bit lower than the MLP, it learns faster than the latter.

Furthermore, some of the parameters with the SOM based architecture reach very high accuracy such as "Classification accuracy", "Sensitivity" and "Negative predictive value".

The advantages of the supervised SOM architecture are based on both the accuracy, which is not far behind that of the MLP, and especially the faster learning. These benefits will be recommended for use either in problems with a lot of data or with many attributes, where data relationships may be complex. A future line would be to apply this method in such problems as an iterative process leading to features reduction in order to simplify the dependency relationships.

5 Acknowledgment

We want to express our acknowledgements to Asuncion, A. & Newman, D.J, Irvine, CA: University of California, School of Information and Computer Science, who has created the database.

References

[1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[2] C. Balkenius, J. Morén, B. Johansson, and M. Johnsson. Ikaros: Building cognitive models for robots. *Advanced Engineering Informatics*, 24(1):40 – 48, 2010.

[3] C. Bishop. Improving the generalization properties of radial basis function neural networks. *Neural Computation*, 3(4):579–588, 1991.

[4] CM Bishop. Neural networks for pattern recognition: Oxford University Press. *New York, New York, USA*, 1995.

[5] C.M. Bishop. *Neural networks for pattern recognition*. Oxford Univ Pr, 2005.

[6] Fi-John Chang, Li-Chiu Chang, Huey-Shan Kao, and Gwo-Ru Wu. Assessing the effort of meteorological variables for evaporation estimation by self-organizing map neural network. *Journal of Hydrology*, In Press, Corrected Proof:–, 2010.

[7] B. Choi and J.H. Lee. Comparison of generalization ability on solving differential equations using backpropagation and reformulated radial basis function networks. *Neurocomputing*, 2009.

[8] A.E. Diaz and C.A. Hernandez. Tool for the design and implementation of control systems with neural networks. In *IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009. ICIS 2009*, volume 1, 2009.

[9] D. Gil, M. Johnsson, J.M. Garcia Chamizo, A.S. Paya, and D.R. Fernandez. Application of artificial neural networks in the diagnosis of urological dysfunctions. *Expert Systems with Applications*, 36(3P2):5754–5760, 2009.

[10] David Gil and Magnus Johnsson. Using support vector machines in diagnoses of urological dysfunctions. *Expert Systems with Applications*, 37(6):4713 – 4718, 2010.

[11] S. Haykin. Neural Networks: A Comprehensive Foundation, Englewoods Cliffs, 1998.

[12] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, 2008.

[13] C.N. Kim, K.H. Yang, and J. Kim. Human decision-making behavior and modeling effects. *Decision Support Systems*, 45(3):517–527, 2008.

[14] R. Kohavi and F. Provost. Glossary of terms. *Machine Learning*, 30(2/3):271–274, 1998.

[15] T. Kohonen. *Self-Organization and Associative Memory*. Springer Verlag, 1988.

[16] M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, and L.O. Ramig. Suitability of dysphonia measurements for telemonitoring of Parkinson? s disease. *IEEE transactions on bio-medical engineering*, 2008.

[17] M.A. Little, P.E. McSharry, S.J. Roberts, D.A.E. Costello, and I.M. Moroz. Exploiting Nonlinear recurrence and Fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(1):23, 2007.

[18] J.V. Marcos, R. Hornero, D. Alvarez, F. del Campo, M. Lopez, and C. Zamarron. Radial basis function classifiers to help in the diagnosis of the obstructive sleep apnoea syndrome from nocturnal oximetry. *Medical and Biological Engineering and Computing*, 46(4):323–332, 2008.

[19] I. MathWorks. *MATLAB & SIMULINK Instrument Control Toolbox 2*. The MathWorks Inc., 2007.

[20] S. Mehrabi, M. Maghsoudloo, H. Arabalibeik, R. Noormand, and Y. Nozari. Application of multilayer perceptron and radial basis function neural networks in differentiating between chronic obstructive pulmonary and congestive heart failure diseases. *Expert Systems With Applications*, 36(3P2):6956–6959, 2009.

[21] M.R. Narasinga Rao, GR Sridhar, K. Madhu, and A.A. Rao. A clinical decision support system using multi-layer perceptron neural network to predict quality of life in diabetes. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 2009.

[22] M. Pal and University of Nottingham (GB). *Factors Influencing the Accuracy of Remote Sensing Classification: A Comparative Study*. University of Nottingham, 2002.

[23] P. Piela. Introduction to Self-Organizing Maps Modelling for Imputation-Techniques and Technology. *Research in Official Statistics*, 2:5–19, 2002.

[24] B.D. Ripley. *Pattern recognition and neural networks*. Cambridge university press, 1996.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[26] F. Temurtas. A comparative study on thyroid disease diagnosis using neural networks. *Expert Systems with Applications*, 36(1):944–949, 2009.

[27] H. Werner and M. Obach. New neural network types estimating the accuracy of response for ecological modelling. *Ecological Modelling*, 146(1-3):289–298, 2001.

[28] H. Yan, Y. Jiang, J. Zheng, C. Peng, and Q. Li. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Systems with Applications*, 30(2):272–281, 2006.