Processing Ambiguous Fault Signals with Three Models of Feedforward Neural Networks

Sergio L. Martínez[1], Enrique E. Tarifa[1,2], and Samuel Franco Dominguez[1]

(1) Facultad de Ingeniería, Universidad Nacional de Jujuy,
Gorriti 237, S. S. de Jujuy, Jujuy, Argentina
{smartinez, eetarifa, sfdominguez}@fi.unju.edu.ar
(2) CONICET, Argentina.
eetarifa@arnet.com.ar

Abstract. In the industrial technological field, running equipment or processes usually is monitored through automatic diagnosis systems. Within several Technologies for implementing such systems, the artificial neuronal networks are the most successful and widely spread. The data signals coming from the equipments or processes under supervision are interpreted by the neuronal networks so as to diagnose the presence of any fault. In this work three models of artificial neural networks and two methods of training are analyzed so as to establish, based on real experiences, the best combination of the neuronal model and the training method for recognizing in an efficient way the ambiguous patterns of faults.

Keywords. Neural Networks. Diagnosis. Ambiguous Fault Signals. Optimized training.

1 Introduction

In the automatic diagnosis of faults in equipments or processes, especially industrial, several sensors associated to these, give data sequences which, precisely interpreted, can reveal the working status, normal or abnormal, of such equipments or processes. Such data are analyzed by a diagnosis system in order to determine the cause (fault) o fan eventual abnormality, so as to establish the necessary correcting actions.

Formally the mission of a diagnosis system is to analyze the status of the process under supervision, so as to determine if any fault (detection stage) has appeared. In such case, the diagnosis system must analyze how is the evolution of the process status so as to identify the fault which originated the abnormal condition (diagnosis stage). This task is complicated since generally there isn't a simple relation between the appearance of a fault and the evolution that it makes in the supervised process. Different faults can cause similar evolutions; on the other hand, the same fault can cause several types of evolutions. This situation is quite a critical problem for the traditional methods of diagnosis.

The diagnosis systems of faults based on artificial neuronal networks (ANN) have achieved a wide level of development and spreading fundamentally because these structures, with architectures of high parallelism, association and fast answer times, make up a very efficient tool for developing in the field of recognizing patterns [4] [9].

In the present work, a theoretical study is realized, based on practical experiences, of the diagnosis process of faults based on neuronal networks. From this study a model of neural network is obtained and a training method with an important modification starting from standard configurations. The experimental results obtained show the superiority of one of the neuronal models proposed and trained with a variation of a classical backpropagation algorithm.

2 Fault Diagnosis

During the operation of a system under supervision (an equipment, a process, a sector of a plant, a complete plant, etc.), the values assumed by the most important variables of the process are captured by a set of sensors strategically put. When the normal status of the process corresponds to a static status, the variables adopt constant values; yet, due to the own noise of the process, it is considered the status as normal while the value evolution of each variable remains within a predefined interval of the original static value; this interval is called band or stripe of normality.

When a fault occurs, the affected variables evolve following defined paths by the fault. At the time when they abandon their respective normality bands, the diagnosis system detects an abnormality and starts the analysis of the observed paths so as to try to identify the problem that originated them.

So as to illustrate this situation, a study of the simplified process is proposed, having one only variable to supervise (for example X temperature), where in addition four potential faults are established identified as f_1, f_2, f_3 and f_4, such as shown in Fig. 1.a.

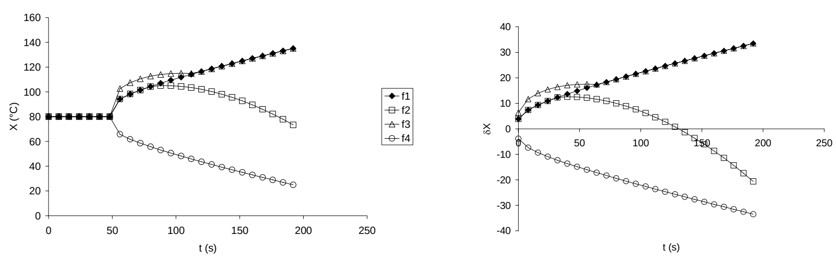


Fig. 1. (a-left) Absolute position regarding path time of X (temperature) for four potential faults. **(b-right)** Typified deviation paths δX transformed by the detector module.

The process operated in a static status, that is to say, while operating normally X adopts a constant predefined value, for example 80ªC. Each time that one of the faults is seen in the process, the X temperature will stop being constant and will evolve with some of the characteristics paths, where the paths have been generated with a simple interval Δt=8 s.

The proposed diagnosis system of faults and associated to the previous process, uses a set of artificial neural networks (ANNs) specialized in the individual recognizing of each potential fault of the supervised process (Fig. 2). The ANNs, operating in real time and monitoring the evolution of the paths of all and each one of the measured variables, analyze the data coming from the process looking for symptoms or tests for their respective faults.

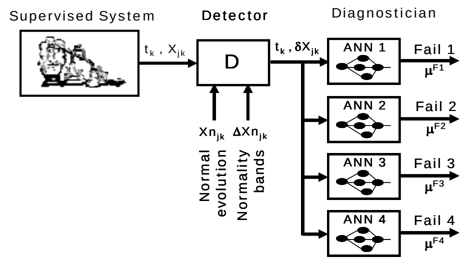


Fig. 2. Diagram of a general system of process-detection-diagnosis.

The result of this analysis is the certainty degree $\mu^F \in [0 , 1]$ that each fault supports. When one of the ANNs produces a null value ($\mu^F=0$) it means that all the tests are against the fault that corresponds to it. On the contrary, an equal value to the unit ($\mu^F=1$) implies that there are no proofs against the own fault of the network. Intermediate values represent intermediate supports for the own fault. As can be seen in Fig. 1.a, the variable –temperature X–, remains in its normal value until one of the faults occurs in the activation time t_a=48 s.

Taking as reference fault f_1, at the beginning the f_2 is undistinguishable, but from the t=88s it starts to differentiate. On the other hand, the f_3 path is at the beginning different from f_1, but from t = 112 s it is mixed up with the reference fault. Finally, the f_4 path is in every moment different from the rest of the faults. The two first cases raise a great difficulty to the diagnosis system because different faults originate paths which at some time they are identical. To overcome this problem an optimized method for training the ANNs was developed.

2.1 The Detector Module

For the example under study, a detector module has been used [7] as it is shown in Fig. 2, which converts the absolute paths of X (Fig. 1.a) in the typified deviation paths δX (Fig. 1.b).

In order to comply with its function, the detector gets a data acquisition system –in each sample interval Δt–, the value of X; being the sample k taken in time $t_k=k\cdot\Delta t$. On the other hand, the detector also knows the normal value Xn and uses it to calculate the quantitative deviation ΔX=X-Xn. Alter the quantitative deviation is calculated, the detector uses the normal band ΔXn, which is also known, to calculate the typified deviation δX, as follows:

$$\delta X_k = \frac{\Delta X_k}{\Delta Xn} \,. \qquad (1)$$

Fig. 1.b shows the transformation of absolute paths of X in the typified deviation paths δX, where the regions that present data ambiguity are observed and which must be solved by the diagnosis system.

2.2 The Diagnosis Module

In the example analyzed, the diagnosis system estimates the degree of certainty $\mu_f^F(k)$ which supports fault f in time t_k with the following recursive formulae:

$$\begin{aligned} \mu_f^F(k) &= \mu_f^F(k-1)\ \mu_f^X(k)\,. \\ \mu_f^F(-1) &= 1\,. \end{aligned} \quad (2)$$

where μ_f^X where μx represents the certainty which supports variable X supposing that fault f occurs, as is calculated as:

$$\mu_f^X = \mathrm{fd}\left(\delta X_1,\ \delta X_f^0\right). \quad (3)$$

being fd(δX, δX^0) the evaluation function which is used to evaluate the difference between the observed value δX, originated by an unknown fault, and the expected value δX^0 [7] [8].

Considering that ANN1 (Fig. 2) has been specialized in recognizing fault f_1 and observing the exit of this network, Fig. 3 represents the paths of the μ^F that would generate the first neuronal network, –already trained–, of the diagnosis system, while comparing the paths δX of Fig. 1.b, originated by each potential fault of the process, with which it would originate if f_1 were the fault that is really happening. The rest of the networks would produce similar answers when getting data of their respective faults of specialization.

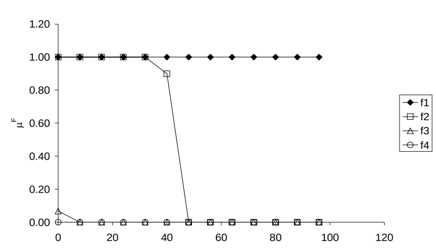


Fig. 3. Paths of μ^F of the ANN1 that supports f_1 facing each potential fault.

As can be seen in Fig. 3, when to the first neuronal network (ANN1) enters the path originated by f_1, μ^F it is kept in 1 backing up at every moment to such fault. On the other hand, when the path originated by f_2 is entered, the system does not rule out that it be f_1 since the paths of both faults are similar up to t=32 s. From t=40 s, the system rules out that fault f_1 is decreasing μ^F. When all the paths of f_3 or f_4 are presented, the ANN1 immediately recognizes that they do not correspond to its specialization fault, and fault f_1 is left aside very early.

3 Neural Networks

The artificial neural networks (ANN) can be considered as mathematical models representatives of the brain activity, with the capacity of learning, memorizing and generalizing the learnt information under a diagram of high tolerance to noise, which makes them powerful and versatile tools for the processing of basically numeric information [1] [2].

In this work, three different architectures of neural networks are analyzed: the classical feedforward architecture (Fig. 4) [2] [5], a feedforward architecture with delay windows (Fig. 6) [2] [5] and a feedback or recurrent architecture (Fig. 8) [2] [6], based on equation (2). In addition, two supervised training/learning methods are applied, the traditional method and an optimized method –both based on the backpropagation algorithm–, with the aim of determining the configuration and learning method which are more efficient to solve the raised problem: to recognize the patterns that allow the identification of the fault that gave them rise.

The difference between the two training methods is deep and determining. In the traditional method the output value μ^F for training is set up according to the fault that generated the δX path which is fed as an input to each ANN; that is, μ^F must assume 1 when it enters the δX path of the fault that was assigned to ANN which is being trained, and must assume 0 when it enters the δX path of a different fault, without considering the shape of this path with the rest of them. The latter constitutes the main weakness of the method, originated in the ambiguity of the information that the coinciding segments of the paths δX give caused by different faults (Fig. 1.b), as is the case of the example under analysis. This situation causes the network to try to adjust to contradictory orders.

For the example, in the training of ANN1, specialized in fault f_1, when the δX path is fed generated by f_1, it is taught to make an output $\mu^F=1$; on the contrary when the path δX generated by f_2, ANN is taught to produce an output $\mu^F=0$; yet, since the first 32 s of both paths are indistinguishable, the network will not be able to decide if μ^F must be 0 or 1 during the time when both paths coincide.

Such said problem is solved applying an optimization variant to the training algorithm, when the μ^F value is set according to the coincidence between the observed δX path and the expected path for the network fault and not in the label of the path that corresponds to each fault, such as is done by the traditional method. For the example which is being studied, when the δX path is fed made by f_1, ANN1 is taught to have an output $\mu^F=1$ because the observed path is the same to the expected one; when the δX path is fed made by f_2, ANN1 is taught to have an output $\mu^F=1$ during the first 32 s since during that time the observed path is equal to the expected; but from that moment, the observed path starts to be different to the expected, and the μ^F value shows this fact decreasing proportionally its value, as shown in Fig. 3. In this way, the ANN1 has no conflict during the learning since for similar inputs δX they must have similar outputs.

Three ANNs models are presented configured with similar architectures and equivalent training conditions so as to be able to compare the results. In relation to practical experiences, only the results of the specialized ANN1 network in recognizing fault f_1 are presented, since it is the one that supports the most unfavorable conditions because of the special configuration of the fault paths.

In relation to the specific training data, each fault path has 13 samples, generated by simulation at intervals of $\Delta t=8$ s (Fig. 1.b). Then, the sequences were concatenated of the four faults, configuring a general sequence of 52 samples.

So as to make the experimental check up the software MatLab® R.9 was used, executed on PC Core 2 Duo equipment with 2 GB of RAM memory. All the ANNs models described in this work have been configured with the neural networks toolbox assistance, available in the software.

3.1 Classical feedforward ANN

It is the typical model of neuronal network used widely for general processes of pattern association. The architecture of each ANN is defined based on the data to be processed and considering the criterion of using an acceptable minimum structure (Fig. 4). According to the Universal Theorem of Function Approximation [2], just one hidden layer is enough for a uniform approximation given a set of training. Due to the dimension of the input and output data, one neuron (fictitious) is used in the input and one in the output. The definition of amount of hidden neurons was experimentally established in four units.

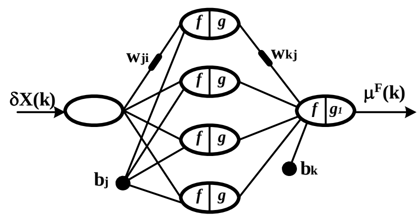


Fig. 4. Classical feedforward ANN architecture.

The exciting or net signal f_j of each neuron is defined –according to a classical sketch–, by the pondered composition of the input signals to each processing unit, i.e.:

$$f_j = \sum_{i=1}^{N} w_{ji}\, x_i - b_j \,. \qquad (4)$$

where x_i is the output of neuron i of the previous layer made up by N neurons, w_{ji} is the weight of the connection between the present neuron j and the one given by signal i, and b_j is the adjustment weight (bias) of neuron j. For the output signal x_j of each hidden neuron the bivalued sigmoid function was adopted, and the positive sigmoid function for the neuron of the output layer.

3.1.1 Experimental data

The behavior of the ANN1 for identifying the paths of Fig. 1.b was quite unfavorable, both when trained with the traditional method and when they were trained with the optimized method (Fig. 5.a and 5.b).

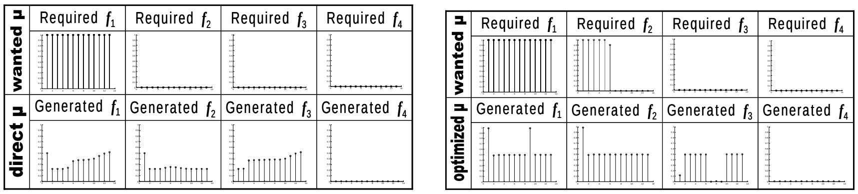


Fig. 5. (a-left) Required/generated output of the classical feedforward model of ANN1. Training: backpropagation standard. **(b-right)** Required/generated output of the classical feedforward model of ANN1. Training: optimized backpropagation.

The training result analysis of the ANN1 network of the diagnosis system –which was trained for recognizing f_1–, reported an μ^F near to 0,5 for the first three faults with both training methods, and a μ^F=0 for the fault f_4. This behavior is incorrect but coherent, since the first three faults present ambiguities which the neuronal network cannot solve, while the fault f_4, since it presents a totally different path than the previous, has been well solved.

The inability of learning of this network is founded in its own structure. In effect, when having to decide the μ^F value (output) according to a unique δX value (input), the ANN1 does not have the necessary information for discriminating among the several paths in which the coinciding segments commit. For example, the input value δX=10 is achieved at some moment by the f_1, f_2 and f_3 paths; and both training methods demand at some moment output values μ^F different for such input, which confuses the network. One way of solving this problem is to increase the amount of available information for the ANN; two alternatives are explored in this way below.

3.2 ANN with delay windows

For certain dynamic processes, having the information of the late past can improve the behavior of the system [3]. In our case, to increase the information sent to the ANN1, the architecture presented in the previous section was widened through the adding of two additional inputs. These new inputs are obtained by keeping, through delays, the last two observed values of δX, adding two temporal windows (Fig. 6).

3.2.1 Experimental data

At first, this differed time architecture is more appropriate than the direct feedforward; however, its performance –although a little better than the previous model–, has not been satisfactory.

Considering once again the first network of the diagnostician (ANN1) –which was trained for recognizing f_1–, generated some incorrect samples in recognizing its specialization fault with both training methods, but we can see, from Fig. 7.a and 7.b, a better performance due to the delay windows which give more information of the paths.

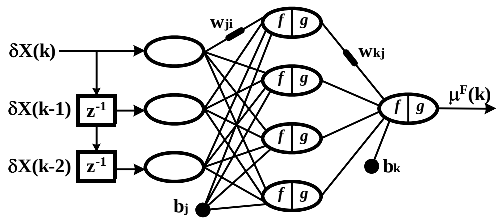


Fig. 6. Feedforward ANN architecture of differed time with two delay windows.

The f_2 fault was better solved with optimized training method and with a bad result with the traditional method, because of contradictory data in the first ambiguous area (Fig. 1.b). The f_3 fault was also recognized with mistakes and the f_4 was well solved as in the previous case.

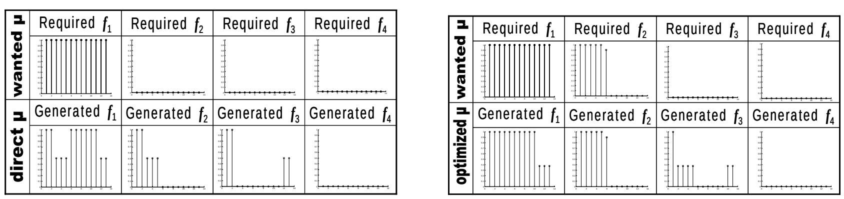


Fig. 7. (a-left) Required/generated output of the feedforward model with windows of the ANN1. Training: backpropagation standard. **(b-right)** Required/generated output of the feed-forward model with windows of the ANN1. Training; backpropagation optimized.

The poor behavior with the traditional method is due to its intrinsic weakness because of the overlapping of some segments of the δX paths, but the failure with the optimized method shows a limitation of the proposed structure. This limitation could be founded in the reduced amount of neurons of the inner layer or for the limited width of the temporal window. Yet, here the analysis of this type of network is halted in favor of keeping a similar structure for comparing the networks and while considering the architecture which is explained in the following section.

3.3 Recurrent ANN

The network shown in Fig. 8 corresponds with the Jordan recurrent model which back feeds the output towards a contextual layer of the input [2]. This structure is based in the equation (2), which gives the theoretical basis, and which was deducted when modeling the desired behavior for the proposed diagnosis system [7] [8].

When having a feedback of the output towards the input, the information which the ANN gets, combines the present status and the total history of the process through the successive product of the feedback outputs [6]. It may be considered that the potentiality of the optimized training method is based on this strategy; once the system has detected that the sequence does not correspond to a fault in consideration, it begins to feedback values of μ^F each time minor until they are cancelled and kept in that status, even when the sequence puts again ambiguous values (such as the case in the second ambiguous area of Fig. 1.b).

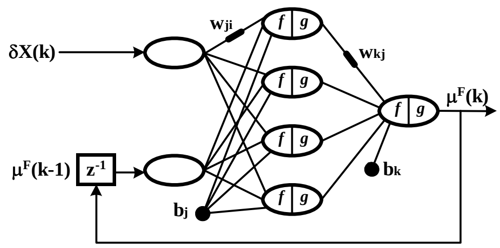
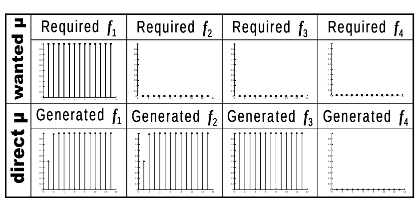


Fig. 8. Recurrent ANN architecture with a delay.

The feedback model of Jordan has a great advantage on the non recurrent architectures that can only see a part of the history of the process; the classical feedforward ANN only has the present simple and the differed time ANN just takes two samples of the inputs backwards. The recurrent model used keeps all the history of the process through a unique numerical additional input: the feedback μ^F. The latter is very important when working with a big number of variables.

3.3.1 Experimental data

To the structural simplicity of the recurrent ANN it is added an excellent behavior of the tests made with the optimized method of training. In effect, the ANN1 -corresponding to fault f_1–, trained with the optimized method, was able to generate almost exactly the desired μ^F outputs, truly reproducing the curves of Fig. 3; yet, its behavior was not consistent when it was trained with the traditional method due to the failures of the used method and not of the proposed structure, as can be seen in Fig. 9.a and 9.b.



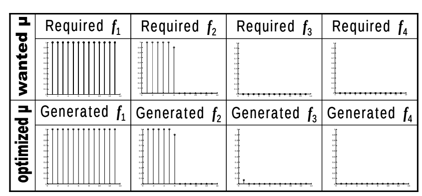


Fig. 9. (a-left) Required/generated output of the recurrent model of the ANN1. Training: backpropagation standard. **(b-right)** Required/generated output of the recurrent model of the ANN1. Training: backpropagation optimized.

The three models of artificial neural networks have experimentally shown their abilities in the recognizing process of patterns. For a better comparison, all the obtained results are briefly shown in the comparison chart of table 1.

Table 1. Comparison chart of the behavior of three neural models and two training methods in the process of recognizing faults.

Models → / Parameters ↓	Standard FF ANN		Delayed FF ANN		Recurrent ANN	
	direct μ	optim. μ	direct μ	optim. μ	direct μ	optim. μ
MSE of training	$1{,}5x10^{-1}$	$1{,}8x10^{-1}$	$4{,}8x10^{-2}$	$3{,}6x10^{-2}$	$9{,}6x10^{-3}$	$2{,}4x10^{-14}$
Non recognized samples	39	37	14	10	27	2
% of recognizing error	75%	71%	27%	19%	52%	3%
Non recognized samples in f_i	13	11	5	3	1	0
Percentage error in fault 1	100%	85%	38%	23%	8%	0%

5 CONCLUSIONS

In this development the efficiency for making identification of paths of faults of three neuronal architectures and two training methods were evaluated. From the study it is deduced that a classical feedforward neural network is not able to assume ambiguous knowledge produced by contradictory information in its training stage. On the other hand, feedforward architecture with delay windows, of comparable complexity to the previous, is also inefficient to learn ambiguous patterns in its training stage, under the two proposed training methods: the Standard backpropagation and the optimized backpropagation.

On the other hand, the combination of a recurrent ANN structure, also of comparable complexity to the previous, and an optimized training method has given an excellent behavior in recognizing fault paths that present coinciding or redundant segments. The feedback structure gave to the network enough information for learning and recognizing complex fault patterns.

Starting from this work, two complementary studies can break down so as to complete the activity of a diagnosis system with neural networks. On one hand, the application of multidimensional fault sequence on models based on ARMA architectures. On the other hand, to improve the training method so that the ANN increases its tolerance related to noise, which could be achieved by adding paths with noise in the training stage, or adding normality bands during the learning stage.

References

1. Anderson, J.: Redes Neurales. Alfaomega Grupo Editor, México (2007)
2. Haykin, S.: Neural Networks – A Comprehensive Foundation. Prentice-Hall, Ontario (1998)
3. Jain, L.C., Martin, N.M.: Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications. CRC Press LLC, Florida (1998)
4. Jain, L., Rao Vemuri, V.: Industrial Applications of Neural Networks. CRC Press LLC, USA (1999)
5. Looney, C.: Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists. Oxford University Press, New York (1997)

6. Mandic, D., Chambers, J.: Recurrent Neural Networks for Prediction. John Wiley and Sons Ltd., New York (2001)
7. Tarifa, E., Martínez, S.: Diagnóstico de Fallas con Redes Neuronales. Parte 1: Reconocimiento de Trayectorias. In Ingeniería e Investigación, vol. 27/1, pp. 68-76, Bogotá (2007)
8. Tarifa, E., Martínez, S.: Diagnóstico de fallas con redes neuronales. Parte I1: Reconocimiento de flujos. In Ingeniería e Investigación, vol. 27/2, pp. 65-71, Bogotá (2007)
9. Zhang, J.: Improved on-line Process Fault Diagnosis through Information Fusion in Multiple Neural Networks. In Computers & Chemical Engineering, vol. 30 – issue 13, pp. 558-571, Elsevier Ltd (2006)