REVISTA MEXICANA DE FÍSICA S **54** (1) 22–29 FEBRERO 2008

Learning limits of an artificial neural network

J.J. Vega and R. Reynoso
*Departamento del Acelerador, Gerencia de Ciencias Ambientales, Instituto Nacional de Investigaciones Nucleares
Apartado Postal 18-1027, México D.F. 11801, México.*

H. Carrillo Calvet
*Laboratorio de Dinámica no Lineal, Facultad de Ciencias, Universidad Nacional Autónoma de México,
México, D.F. 04510.*

Recibido el 14 de mayo de 2007; aceptado el 26 de octubre de 2007

Technological advances in hardware as well as new computational paradigms give us the opportunity to apply digital techniques to Pulse Shape Analysis (PSA), requiring powerful resources. In this paper, we present a PSA application based on Artificial Neural Networks (ANNs). These adaptive systems offer several advantages for these tasks; nevertheless it is necessary to face the particular problems linked to them as: the selection of the learning rule and the ANN architecture, the sizes of the training and validation data sets, overtraining, the effect of noise on the pattern identification ability, etc. We will present evidences of the effect on the performance of a back-propagation ANN as a pattern identifier of both: the size of the noise that the Bragg curve spectrometer signal present and of overtraining. In fact, these two effects are related.

Keywords: Neural networks; Bragg curve spectroscopy; digital pulse-shape analysis; pattern identification.

Los avances tecnológicos del hardware lo mismo que los nuevos paradigmas computacionales brindan la oportunidad de aplicar técnicas digitales al Análisis de Forma de Pulsos (PSA), lo cual requiere de recursos poderosos. En este trabajo, se presenta una aplicación de PSA basada en Redes Neuronales Artificiales (ANNs). Estos sistemas adaptivos ofrecen varias ventajas para estas tareas; sin embrago es necesario enfrentar los problemas particulares asociados a ellos como: la selección de la ley de aprendizaje y de la arquitectura de la ANN, los tamaños de los conjuntos de datos de entrenamiento y de validación, el sobreentrenamiento, el efecto del ruido sobre la habilidad para identificar patrones, etc. Se presentarán evidencias del efecto sobre el rendimiento de una ANN de retro-propagación como reconocedor de patrones del: tamaño del ruido que la señal de un espectrómetro de curva de Braga presenta así como del sobreentrenamiento. De hecho, estos dos efectos están relacionados.

Descriptores: Redes neuronales; espectroscopia de curva de Braga; análisis digital de forma de pulsos; identificación de patrones.

PACS: 07.05.Kf; 07.05.Mh; 29.40.Cs

1. Introduction

Digital pulse shape analysis, DPSA, is becoming a quite helpful technique in demanding situations where complex interesting signals need to be analyzed. Fortunately, due to relatively recent hardware and software developments, see a summary in [1], DPSA techniques are being developed [1-11]. Bragg curve spectroscopy, BCS, [12-15], is one field where DPSA may be applied [1,2,9-11]. BCS is used as a particle identification technique, and, traditionally, it is based on the measurement of two parameters, the total energy of an ion or particle, E_{Tot}, and its Bragg peak amplitude, *BP*, maximum of the specific stopping power curve of the ion when traversing a gas medium ($S(E) = dE = dx \equiv$ Bragg curve). These two signals are obtained by feeding the output from the anode of a Bragg curve spectrometer to two amplification electronic branches, one with a large integration time, E_{Tot} signal, and the other with a short one, *BP* signal [12-15].

In [1], we presented a novel way to extract relevant parameters associated with the outgoing ions from nuclear reactions. It was based on digitizing the signals provided by a Bragg curve spectrometer, allowing the implementation of more thorough DPSA. Due to the complexity of this task, it was required to take advantage of new and more powerful computational paradigms. This was fulfilled using a back-propagation ANN as a pattern identifier of synthetic BCs. We used the common technique of early stopping [16] in order to take care of overtraining, which is a known problem during the training stage of an ANN [16-30]. The patterns analyzed in Ref. 1 were synthetic noisy BCs. A synthetic noise component was added to simulate any possible source of noise that normally goes with the experimental signal of interest. As it was expected, overtraining, *i.e.*, overfitting the data by the ANN during training, was observed. In [2], we determined the effect of the size of the noise component on the appearance of overfitting during the training stage. Here, we present a continuation of the search for the effect of overfitting with noise components sizes equal to 1%, 5%, and 9%, and compare the results to the 3 cases studied in [2] corresponding to 2%, 6% and 10%.

2. DPSA and Bragg curve spectroscopy

The new powerful DPSA approach pursued in this paper is described in Ref. 1. It is essentially based on analyzing synthetic BCs and saving them as 81-tuples of bins, discrete values or parameters $\{\mathrm{S(i)}\}_{i=1;81}$. In Figs. 1a-c, it is shown examples of the used ideal BCs (solid line) together with a simulated synthetic experimental BC (dots) that includes a fast changing component taking into account any possible origin of experimental noise of sizes equal to 1%, 5% and 9% (size of the standard deviation).

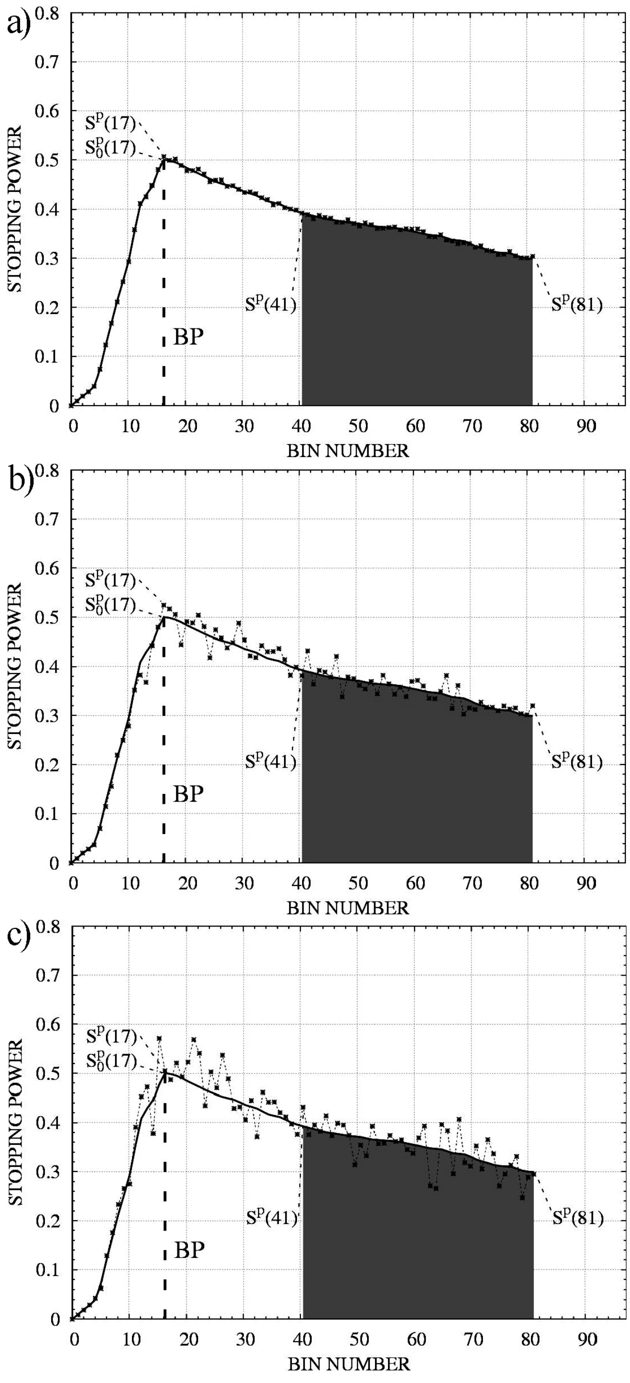

FIGURE 1. Plots of an ideal synthetic BC (solid line) together with a simulated synthetic experimental BC (dots) which includes a fast changing component that takes into account any possible origin of experimental noise equal to: a) 1%, b) 5%, and c) 9%. The training and validation data sets were built using BCs of a length consisting of at least 41 bins.

3. Artificial neural networks

A way to measure how well an ANN is learning its task is by observing, as a function of the number of training epochs, the reduction in the training and validation sum of squares error functions calculated over the entire training and validation data sets, D^T and D^V, respectively, *i.e.*:

$$E^T(\rho) = \frac{1}{K} \sum_{p\in D^T}^{\left|\vec{y}^t - f[\vec{x}_p;\vec{w}(\rho)]\right|^2}$$

or

$$E^V(\rho) = \frac{1}{K} \sum_{p\in D^V}^{\left|\vec{y}^t - f[\vec{x}_p;\vec{w}(\rho)]\right|^2} \qquad (1)$$

where $\vec{w}(\rho)$ represents the link array after training the ANN for ρ epochs, K represents the number of patterns in D^T or D^V and $f[\vec{x}_p;\vec{w}(\rho)]$ is the ANN output for pattern p after ρ training epochs. In an error-correction learning algorithm, the goal of the learning process is to adjust the free parameters, the link array $\vec{w}$ spanning the link space, so as to minimize the training sum of squares error function, $E^T(\rho)$, considered it as a cost functional over the number of training epochs ρ. The initial values of the link array components are chosen according to a uniform random distribution over the interval [-0.5, 0.5]. In order that the ANN keeps its generalization capability (the ability to identify patterns from the validation data set rather than the training data set), during the training stage, overtraining has to be prevented, this means, one should impose an early stopping of the training process, this is, ones $E^V(\rho)$ reaches a minimum value, say at ρ_{min}, even though $E^T(\rho)$ keeps on decreasing. In [1], it was shown that this is not exactly correct, in some cases it is required to keep on training the ANN a little bit more over ρ_{min}.

For the problem of BCs identification using a feed-forward ANN, we chose the error back-propagation learning law including a momentum term [31], since it has proved to be efficient for pattern identification tasks [1,2,8,32-34].

4. Pattern representation and ANN architecture

Rather than building the data sets using synthetic BCs spanning the whole range of 81 bins corresponding to the different possible discrete values of E_{Tot}, we used BCs corresponding to the 41 largest total energies only, the shaded region in Figs. 1a-c. In this way, one warrants that all the analyzed BCs achieve their BP, which, for all our ideal synthetic BCs, corresponds to the 17^{th} bin, considerably simplifying, in this way, the ANN task. The training and validation sets are defined as:

$$D^T \equiv \{[\{S^p(i; E^t_{Tot}(n), BP^t)]_{i=1,81}\}_{p=1,K}$$

and

$$D^V \equiv \{[\{S^p(i; E^t_{Tot}(n), BP^t)]_{i=1,81}\}_{p=1,K}, \qquad (2)$$

where, for a given p, $[\{S^p(i;E^t_{Tot}(n),BP^t)]_{i=1,81}$ models an experimental synthetic noisy BC corresponding to a total ideal energy target value equal to and to an ideal Bragg peak target value equal to BP^t, where: $41 \leq n \leq 81$ and $S^p(i;E^t_{Tot}(n),BP^t) = 0$ if $n < i$. K is the number of patterns in each one of the data sets, and it is equal to 45,100, corresponding to 100 BCs for each one of the 451=11x41 different classes of BCs, 11 different BP^t values times the 41 different $E^t_{Tot}(n)$ discrete values. The experimental noisy BCs were defined as:

$$S^p(i;E^t_{Tot}(n),BP^t) = S^p_0(i;E^t_{Tot}(n),BP^t) + S^p_G(i;E^t_{Tot}(n),BP^t) \tag{3}$$

where$S^p_0(i;E^t_{Tot}(n),BP^t)$ represents a smooth ideal BC (solid lines in Figs. 1a-c), and $S^p_G(i;E^t_{Tot}(n),BP^t)$ a fast noise component that follows a Gaussian distribution with a mean value equal to 0 (dashed lines in Figs. 1a-c) and an energy dependent standard deviation equal to $e \times S^p_0(i;E^t_{Tot}(n),BP^t)$, where e is equal to 0.01, 0.05 and 0.09 corresponding to error sizes of 1%, 5% and 9%. The architecture of the employed ANN is a fully connected feed-forward network with: an input layer of 81 neurons, 5 hidden layers of 9 neurons each, and an output layer of 2 neurons. For all neurons, a sigmoid nonlinear activation function was defined in terms of a logistic function, *i.e.*:

$$g(\vec{x}) = \frac{1}{1 + exp[-(\vec{w} \cdot \vec{x} - \theta)]}, \tag{4}$$

where $\vec{x}$ is the neuron input array with information coming from all neurons from the previous layer which is pondered by the weight array, $\vec{w}$, associated with the links that carry each one of the inputs, and θ is the bias value of the neuron. In Table I, it is presented a summary of all the parameters of the ANN and of the training and validation data sets.

TABLE I. ANN training and validation data set parameters.

PARAMETER	VALUE
Learning law	Back-propagation with momentum term
ANN size	Input layer: 81 units 5 hidden layers: 9 units each Output layer: 2 units Fully connected
αg =learning rate	0.3 and 5.0
μ =momentum term	0.15
w initialization range	[-0.5,0.5]
Order of pattern presentation	Shuffle
Activation function	Sigmoid
Neurons update order	Serial order
$\mathbf{D}^T$ and $\mathbf{D}^V$ data sets	100 samples of each one of the 451 classes: (11 *BP* values) × (41 E_{Tot} values)
CPU time (SUSE LINUX) Intel Pentium 4, 1.7 GHz	30 days of execution time for a training of 1,000,000 epochs

5. Results

Since our main concern is to find out how the signal noise present in the Bragg curves systematically limits the learning capability of our ANN, we trained 3 ANNs corresponding to 3 different values of the signal to noise ratio (S/N): 1%, 5% and 9%, and compare them to the 3 cases analyzed in Ref. 2, 2%, 6% and 10%. From a previous work, [1], we know that due to the link array random initialization when using a uniform random distribution over the interval [-0.5, 0.5], the subsequent ANN learning evolution, as monitored by the shapes of the error curves $E^T(\rho)$ and $E^V(\rho)$, may be drastically perturbed, but, eventually, $E^V(\rho)$ will reach its minimum value. According to this, from our noise analysis, we would expect to observe that, the minimum $E^V(\rho_{min})$ as function of the S/N value showed a general increasing trend, which might be modulated by the ANN link array random

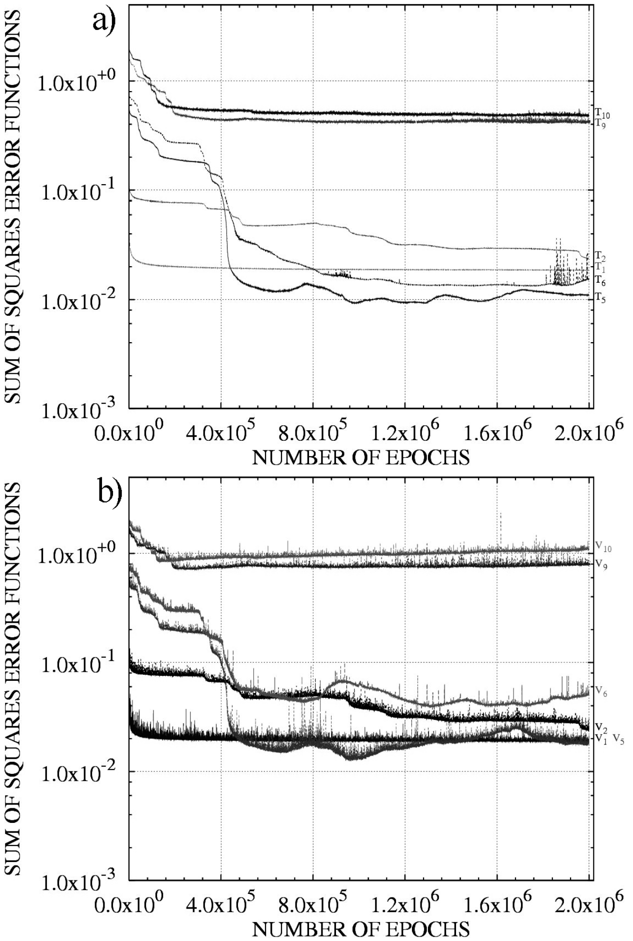

FIGURE 2. $E^T(\rho)$ and $E^V(\rho)$ error curves corresponding to S/N equal to 1%, 2%, 5%, 6%, 9% and 10% over the range from 0 to 2,000,000 epochs.

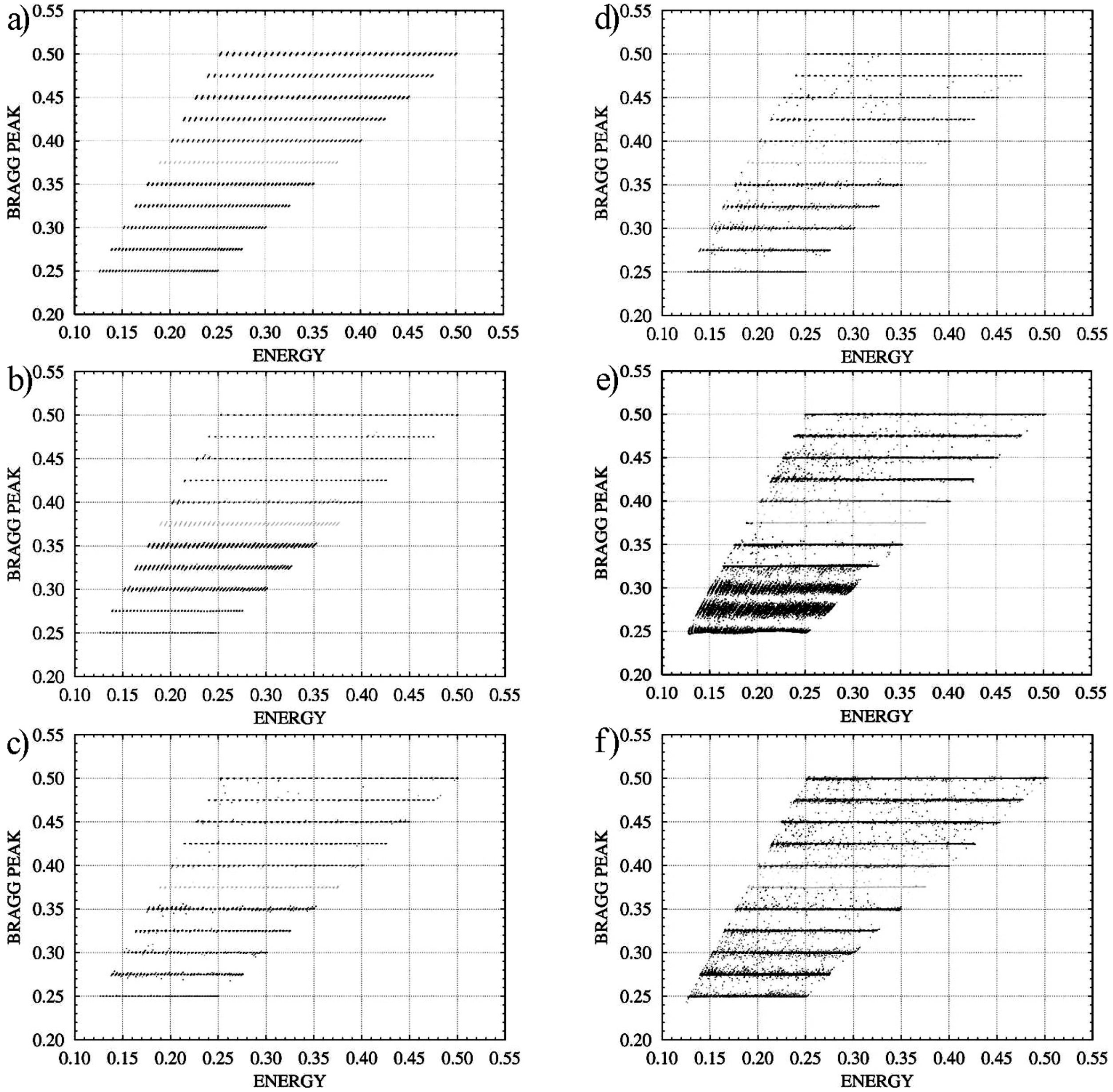

FIGURE 3. *BP* vs. E_{Tot} validation scatter plots corresponding to 2,000,000 training epochs for noise sizes equal to a) 1%, b) 2%, c) 5%, d) 6%, e) 9% and f) 10%. Figs. 3c, 3d, and 3f, look reasonable. In fig. 3a, it can be seen that the ANN has been able to reach a good classification of all the groups (one could say a perfect classification), but it has not learnt all the groups with the same accuracy. In figure 3b, it can be seen that the ANN has been able to learn all the groups reasonably well but, definitively, there are patterns with a bad identification. And, finally, in Fig. 3e, it is clear that the ANN has not been able to learn all the different family classes.

initialization. In Figs. 2a-b, we present the error curves $E^T(\rho)$ and $E^V(\rho)$ corresponding to S/N equal to 1%, 2%, 5%, 6%, 9% and 10% over the range from 0 to 2,000,000 epochs. From these figures it can be seen that, after 2,000,000 epochs, the error curves corresponding to 1% and 2%, when compared to the 5% and 6% curves, seem to indicate that they have not been able to attain their minimum value yet. This suspicion can be verified by looking at the corresponding *BP* vs. E_{Tot} scatter plots shown in Figs. 3a-f. In the case of a 1% noise, it can clearly be seen that, although the ANN is already capable of a perfect classification, it has not reached a perfect identification yet, *i.e.*, it seems that the sizes of the spots corresponding to each one of the 451 different *BP* vs. E_{Tot} classes could be made smaller if one continues training the ANN. In some extent, this also occurs in the case of a 2% noise, suggesting that it also requires additional training. This does not happen to occur for 5%, 6% and 10% noises. Apparently, from the same figures, the 9% case also demands additional training. For that reason we decided to train the corresponding ANNs some additional epochs. In the case of a 1% noise, the ANN was trained up to 10,000,000 epochs, the 2% one up to 5,000,000 epochs, and

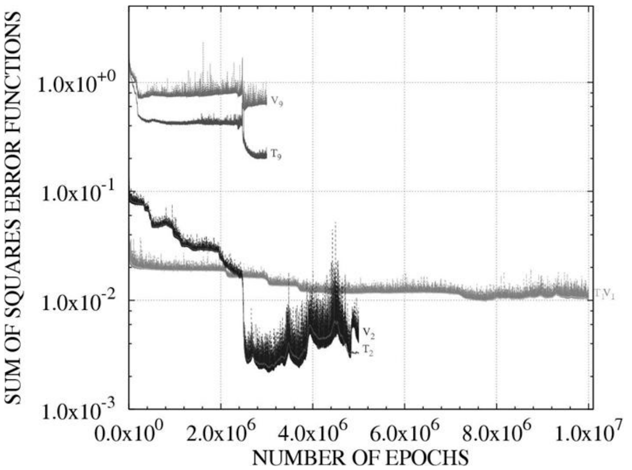

FIGURE 4. $E^T(\rho)$ and $E^V(\rho)$ error curves corresponding to S/N equal to 1%, 2%, and 9% over the ranges from: 0 to 10,000,000, 0 to 5,000,000, 0 to 3,000,000, epochs respectively.

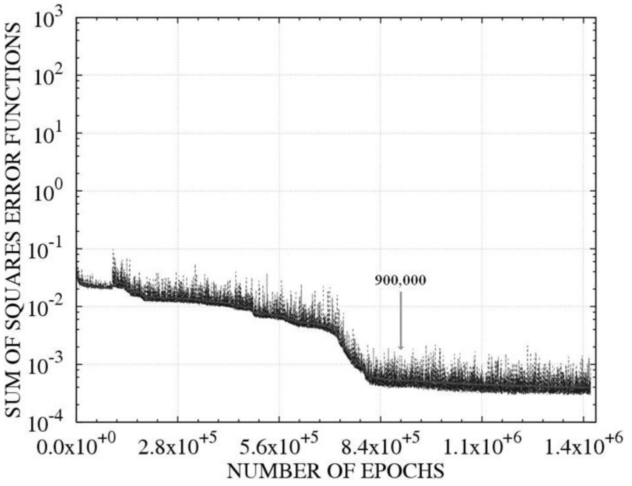

FIGURE 5. $E^T(\rho)$ and $E^V(\rho)$ error curves corresponding to S/N equal to 1% from 0 to 1,420.000 epochs, but the learning rate parameter was changed from 0.3 to 5.0 after 100,000 training epochs.

the 9% one up to 3,000,000 epochs. In Fig. 4, we present $E^T(\rho)$ and $E^V(\rho)$ error curves for all cases. It is seen from this figure that the 2% and 9% cases reach their minimum values at approximately 3,050,000 and 2,515,000 epochs respectively. Although the 1% error curves showed a considerable enhancement, 10,000,000 training epochs were not enough in order to reach the minimum. But, in this last case, from the smoothness of the error curves (after 100.000 epochs) we concluded that the training effect induces a very slow evolution along the error surface, so it was decided to start over the training of the ANN beginning at 100,000 epochs but, this time, the learning factor was changes from 0.3 to 5.0. The corresponding $E^T(\rho)$ and $E^V(\rho)$ error curves are shown in Fig. 5 up to 1,420,000 training epochs. It can be seen that, this time, after 800,000 training epochs using a large learning factor, this is, a total of 900,000 training epochs after tacking into account the initial 100,000 epochs that used a small learning factor, the ANN has been able to reach its minimum value equal to 0.0006. This 100,000 old epochs plus the 800,000 new epochs roughly corresponds to $100,000 + (5/0.3) \times 800,000 \approx 13,433,333$ old epochs.

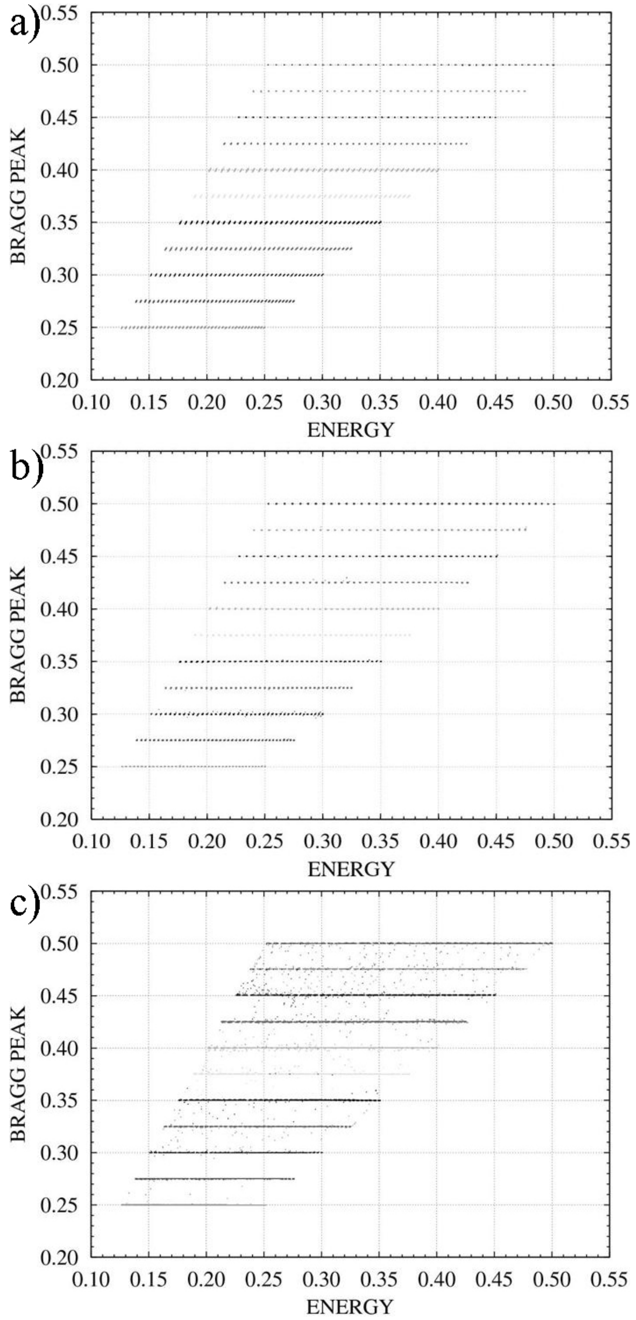

FIGURE 6. a) *BP* vs. E_{Tot} validation scatter plot corresponding to S/N equal to 1% after the minimum value is reached at 900,000 epochs, using a learning factor equal to 0.3 from 0 to 100,000 epochs and equal to 5.0 from 100,000 on. b) *BP* vs. E_{Tot} validation scatter plot corresponding to S/N equal to 2% after the minimum is reached at 3,050,000 epochs. c) *BP* vs. E_{Tot} validation scatter plot corresponding to S/N equal to 9% after the minimum is reached at 2,515,000 epochs.

The *BP* vs. E_{Tot} scatter plots obtained in the 1%, 5% and 9% cases are shown in Figs. 6a-c. In the 1% case the ANN was trained using a learning rate equal to 0.3 from 0 to 100,000 epochs, and equal to 5.0 from 100,000 epochs on, and the minimum was reached after a total of 900,000 training epochs. In the 5% and 9% cases the learning factor used was equal to 0.3 and the corresponding minima were reached at 960,000 and 2,515,000 epochs.

Table II summarizes the number of training epochs that it takes each one of the validation error curves $E^V(\rho)$ to reach its minimum value. It also presents the minimum value $E^V(\rho_{min})$ and its standard deviation, $stdE^V(\rho_{min})$, calculated in a small neighborhood around ρ_{min}. As it is natural to expect, the minimum value reached by $E^V(\rho)$ increases as the corresponding S/N ratio increase. In Figs. 7a-b, we show a plot of $E^V(\rho_{min})$ vs. S/N in linear and logarithmic scales. The dotted lines in Figs. 7a-b were obtained after fitting a straight line to $ln\left[E^V(\rho_{min})\right]$.

The uncertainties shown in Figs. 7a-b are displayed just to give us an idea of the variability of $E^V(\rho)$ in a neighborhood around ρ_{min} rather than the variability of $E^V(\rho_{min})$ around its average value, *i.e.*, the values obtained when repeating the whole training of the ANN several times. The only case in which we are capable of estimating the size of this additional source of uncertainty is in the 10% case, because, in that case, we repeated the training of the ANN 13 times, allowing us to estimate the average value and the standard deviation of all these 13 values. The result obtained in this way for the 10% case is 0.907 ± 0.032 in comparison to 0.868 ± 0.022 that corresponds to the average value of the fastest learning ANN for a 10% S/N, and its standard deviation around the minimum, both quantities were calculated over the interval from 127,000 to 146,000 learning epochs, *i.e.*, around the minimum of the corresponding $E^V(\rho)$ error curve. For the interested reader, in Ref. 1 we explained why although in this 10% S/N case the minimum value of $E^V(\rho)$ is reached after 136,000 training epochs, we preferred to train the ANN up to 196,000 epochs. Since these two sources of uncertainty may be considered as independent then the total uncertainty will be given by the square root of sum of squares of both values, which amounts to ± 0.039. The result of fitting a straight line to $ln\left[E^V(\rho_{min})\right]$ as a function of S/N, Fig. 7b, may be expressed as:

$$E^V_{min}(n) = \alpha exp(\beta n), \tag{5}$$

where α = 0.00037514, β = 0.80284, and n represents the corresponding S/N value. This curve describes how the noise present in the signals or patterns to be identified, n, contributes to set limits to the learning capability of an ANN.

One thing that at first sight appears baffling is the fact that, for small values of S/N, the ANN takes longer to learn its task, *i.e.*, the $E^V(\rho)$ error curve reaches its minimum value slowly, requiring a large number of training epochs, see 1% and 2% cases in Table II. In these cases, it is undeniable that it takes longer to learn their tasks, but the point is that the different tasks are not equivalent, and this is a result of to the limiting learning effect due to the noise present in the signals. In other words, at low values of S/N, there are subtle features that are learnable, but, due to the noise masking effect, at high S/N values, they cannot be learnt. In fact, this masking effect grows smoothly as a function of n or S/N value, as suggested by Eq. 5, Fig. 7b. These subtle features, although learnable, are not easy to be learnt by the ANN, in deed, they demand

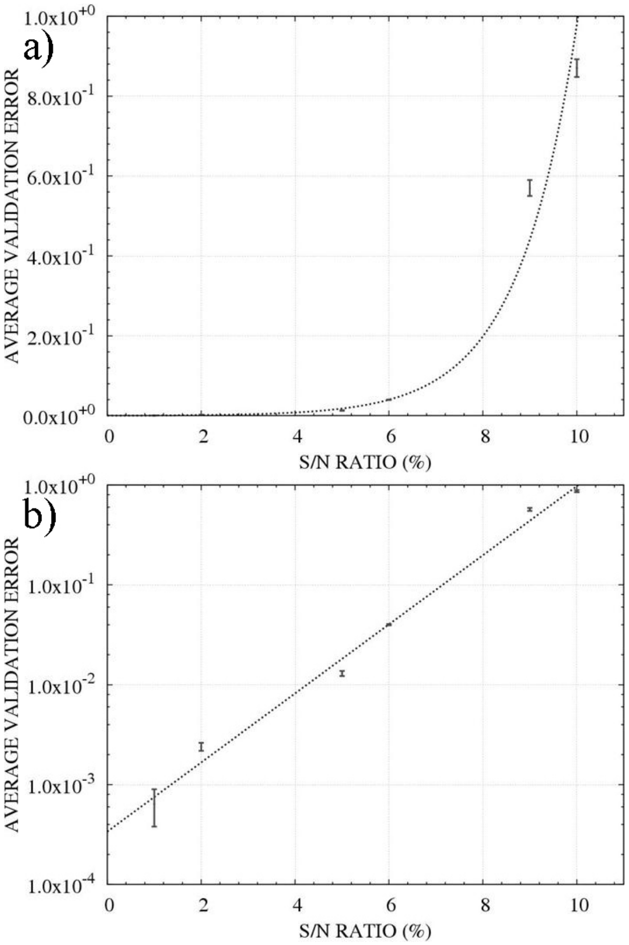

FIGURE 7. Plot of $E^V(\rho_{min})$ vs. n = S/N in linear and logarithmic scales. The dotted lines were obtained fitting a straight line to $ln\left[E^V(\rho_{min})\right]$.

TABLE II. Summary of the number of training epochs that it took each one of the validation error curves $E^V(\rho)$ to reach their minimum values. In the 1% case, the total number of training epochs is split in 100,000 epochs using a learning factor equal to 0.3 and 800,000 training epochs using a learning factor equal to 5.0 what amounts to an equivalent of 13,433,333 learning epochs with a 0.3 learning factor.

N/S (%)	ρ_{min} (epochs)	$E^V(\rho_{min})$	$stdE^V(\rho_{min})$
1	$100,000 + 800,000$	0.0006	0.00026
	$\approx 13,433,333$		
2	3,050,000	0.0022	0.00022
5	960,000	0.012	0.00076
6	1,308,000	0.040	0.00062
9	2,515.000	0.55	0.020
10	196,000	0.87	0.022

a large number of training epochs. The decreasing ability to learn subtle features as the parameter n increases is reflected in the diminishing quality of the corresponding *BP* vs. E_{Tot} scatter plots as the parameter n increases; see Figs. 3a-f and 6a-c. One may assume that the extent of the spreading of the 100 (*BP*, E_{Tot}) values predicted by the ANN around the corresponding tutorial value for a fixed class, is determined by the amount of features that the ANN may be able to extract during the learning or training stage. Of course, subtle features takes longer to be learnt, but, as far as the S/N value is small enough as not to mask that particular feature, it will eventually be learnt by the ANN. This means that, for small S/N values, the error surface is quite smooth once the ANN has learnt the gross features of the patterns. This is why we were successful when, in the 1% S/N case, we replaced the original 0.3 learning factor by a much larger value equal to 5.0 and the ANN was still able to learn with no problem. On the other side, a large noise, large n, implies a bumpy error surface, preventing the learning rule to be able to detect shallow minima corresponding to subtle features.

Visualizing the learning process as adaptation or evolution of the ANN configuration point in link space, which, when using a back-propagation learning law, is equivalent to following the steepest descendent path [35], then overtraining may be accounted for by those regions in link space where the steepest descendent path is due to structures in the error surface that have nothing to do with the relevant pattern features of interest, but rather they are produced by the concomitant signal noise. Unlikely, ANN feature learning occurs when the configuration point in link space traverses regions where the steepest descendent path is pronounced and produced by a particular feature of interest. Once this feature is learnt by the ANN, the shape of the error surface is again dominated by noise, and overtraining will be observed for a while until the ANN configuration point gets to a new region, where, once more, the shape of the error surface is dominated by the influence of the new feature ready to be learned. In the $E^V(\rho)$ error curves of Figs. 2b and 4 one can clearly observe reflected this behavior. Eventually, the ANN will get to a point where the remaining features not yet extracted or learnt by it, are not able to dominate any more the shape of the error surface in link space, and noise will take over for that point on as the error surface shape defining factor, *i.e.*, overtraining will dominate the learning process for that point on. The idea is to stop the training process at that point.

In the particular case of a 1% noise, one can see from the qualities of Figs. 3a, 6a and the smoothness of Fig. 5, that the ANN was able to learn the gross patter features after only 100,000 training epoch, a small number in comparison to the other S/N cases. But in order to extract the rest of the pattern features, presumably subtle features, it took the ANN a very large number of epochs to learn them. In deed, the S/N is so small that overtraining never shown up. It might be that the ANN is still extracting pattern features after a training of 100,000 training epochs with a 0.3 learning factor plus 1,320,000 training epochs with a 5.0 learning factor, totaling an equivalent of 100,000 + (5.0/0.3)x1,320,000 $\approx$ 22,100,000 training epochs with a 0.3 learning factor. Of course, for practical classification applications, it might not be necessary to let the ANN to try to extract all the pattern features, since all one needs to do is to perform a pattern identification, which may be achieved once the ANN has learnt a certain number of pattern features and not necessary all of them. This is why we decided, from a practical point of view that, after a training of 100,000 (0.3) + 800,000 (5.0) epochs, the ANN is suited for our pattern identification purposes. In fact, the ANN trained 2,000,000 (0.3) epochs or even 100,000 (0.3) might work as well also. We decided to keep on training the ANN because we were not only interested in developing the ANN classification ability, but rather in observing how it was going to learn or extract the remaining unlearned features, *i.e.*, how its $E^V(\rho)$ error curve was going to evolve.

Finally, is good to remind that the search for the minimum of $E^V(\rho)$ using the steepest descent path depends a lot on the random initialization point, $\vec{w}(0)$, in link space. Depending on the circumstances, a particular initial value, $\vec{w}(0)$, may lead the evolution of the ANN through regions corresponding to a slow adaptation path or vice versa. The 9% case corresponds to the first case, requiring 2,515,000 epochs to get to the minimum, and the 10% case to the second, requiring only 136,000 epochs. The point is that, although the selected path to get to the minimum may happen to be shorter or longer, there are limits set to the amount of feature extraction that the ANN may learn from a training data set, and those limits are defined by the S/N but not by the path followed to get to the minimum.

6. Conclusions

We have presented evidences supporting an explanation and parameterization, Eq. 5, of how the concomitant noise that distorts the signals or patterns to be identified by an ANN set limits to its learning capability. Also, we have presented evidences that explain overtraining as a competition between the patterns relevant features, on the one side, against the signal noise, on the other side, as the main cause defining the shape of the error surface in link space and, consequently, determining the steepest descent path that controls the ANN adaptation process.

1. J.J. Vega and R. Reynoso, *Nucl. Instr. and Meth. B***243** (2006) 232.
2. J.J. Vega and R. Reynoso, *Rev. Mex. Fís. S* **53(3)** (2007) 118.
3. T. Kihm, V.F. Bobrakov and H.V. Klapdor-Kleingrothaus, *Nucl. Instr. and Meth. A* **498** (2003) 334.
4. T. N. Ginter, Ph. D. Dissertation, Vanderbilt University, Nashville, Tennessee, (1999).
5. C.J. Gross *et al.*, *Nucl. Instr. and Meth. A* **450** (2000) 12.
6. J. Hellmig and H.V. Klapdor-Kleingrothaus, *Nucl. Instr. and Meth. A* **455** (2000) 638.
7. J. Hellmig, F. Petry and H.V. Klapdor-Kleingrothaus, Patent DE19721323A.
8. B. Majorovits and H.V. Klapdor-Kleingrothaus, *Eur. Phys. J. A* **6** (1999) 463.
9. L. Andronenko *et al.*, Preprint PNPI NP-3-1998 Nr. 2217.
10. N.J. Shenhav and H. Stelzer, *Nucl. Instr. and Meth.* **228** (1985) 359-364.
11. M.N. Andronenko and W. Neubert, *Annual report 1998-1999* FZR-271 (1999) 67.
12. A. Moroni *et al.*, *Nucl. Instr. and Meth.* **225** (1984) 57.
13. M.F. Vinyard *et al.*, *Nucl. Instr. and Meth. A* **255** (1987) 507.
14. K.E. Rhem and F.L. Wolfs, *Nucl. Instr. and Meth. A* **273** (1988) 262.
15. J.J. Vega, J.J. Kolata, W. Chung, D.J. Henderson and C.N. Davids, Proc. *XIV Symposium on Nuclear Physics, Cuernavaca, Mexico, 1991*, M. Brandan, (ed., World Scientific, Singapore, 1991) 221.
16. C.M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press-Oxford, (1995).
17. C.M. Bishop, *IEEE Transactions of Neural Networks* **4** (1993) 882.
18. C. Wang, S.S. Venkatesh and J.S. Judd, *NIPS'1993* **6** eds. J. Cowan, G. Tesauro, J. Alspector, Morgan-Kaufmann, (1994) 303.
19. W.S. Sarle, in Proc. *of the 27 th Symposium on the Interface of Computing Science and Statistics* (1995) 352-360.
20. A.S. Weigend, M. Mangeas and A.N. Srivastava, *International Journal of Neural Systems* **6** (1995) 373.
21. C.M. Bishop, *Neural Computation* **7** (1995) 108.
22. S. Bös, *ICANN'1995* **2**, ed. by EC2 & Cie, 111.
23. S. Bös, *NIPS'1995* **8**, eds. G. D. Touretzky, M. Mozer, M. Hasselmo, MIT Press, (1996) 218.
24. P. Sollich and A. Krogh, in *Advances in Neural Information Processing Systems* **8**, eds. D. S. Touretzky, M. C. Mozer and M. E. Hasselmo, MIT Press, (1996) 190.
25. S. Amari, N. Murata, K.-R. Finke, M. Finker and H. Yang, *IEEE Transaction on Neural Networks* **8** (1997) 985; and *NIPS'1995* **8**, eds. G. D. Touretzky, M. Mozer, M. Hasselmo, MIT Press, (1996) 190.
26. S. Amari and N. Murata, *IWANN'1997*, eds. J. Mira, R. Moreno-Diaz, J. Cabestany, Springer, (1997) 284.
27. S. Lawrence and C. L. Giles, *IJCNN'00* Vol. 1, eds. Shun-Ichi Amari, C. Lee Giles, Marco Gori, and Vincenzo Piuri, IEEE Press, (2000) 114.
28. P. Domingos, *ICML'2000*, Morgan Kaufman, (2000) 223.
29. G. N. Karystinos and D.A. Pados, *IEEE Transactions on Neural Networks* **11** (2000) 1050.
30. R. Caruana, S. Lawrence and C.L. Giles, *NIPS'2000* Vol. **13**, eds. T. Leen, T. Dietterich, V. Tresp, MIT Press, (2001) 28.
31. A. Zell *et al.*, *SNNS - Stuttgart Neural Network Simulator, version 4.2*, University of Stuttgart, Institute for Parallel and Distributed High Performance Systems; and University of Tübingen, Wilhem-Schickard-Institute for Computer Science, (1998).
32. J. Damgov and L. Litov, *Nucl. Instr. and Meth. A* **482** (2002) 776.
33. M. Ambrosio *et al.*, (The MACRO Collaboration), *Nucl. Instr. and Meth. A* **492** (2002) 376.
34. E. Yoshida, K. Shizuma, S. Endo and T. Oka, *Nucl. Instr. and Meth. A* **484** (2002) 557.
35. R. Hecht-Nielsen, Neurocomputing, Addison-Wesley Publishing Company, (1990).