Applied Mathematical Sciences, Vol. 1, 2007, no. 19, 915 - 929

# Using Neural Nets for Eliminating Noise in Experimental Signals

**George Karatasios**

Physics Department, University of Thessaloniki
University Campus, 54124 Thessaloniki, Greece
gkaratasios@auth.gr

**Ioannis P. Antoniades**

Department of Informatics, University of Thessaloniki
University Campus, 54124 Thessaloniki, Greece
ipa2@auth.gr

**Eleni Fournou**

Department of Applied Sciences
Technological Education Institution (TEI) of Thessaloniki
57400 Sindos, Greece
efournou@gen.teithe.gr

**Panos Argyrakis**

Physics Department, University of Thessaloniki
University Campus, 54124 Thessaloniki, Greece
panos@physics.auth.gr

**Abstract**

In this paper we present a technique using feedforward neural nets to treat some types of noise frequently encountered in experimental measurements. We show that these neural nets act as inherent filters in lineshapes contaminated with noise, and propose solutions that could be implemented online to smooth the curves of interest. This method works just as well or better than conventional methods for treating random noise. We first show how such a net can learn to recognize a function, and then show how we can extract parameters in that function, such as, for example, the temperature in a temperature-dependent energy distribution. We then show that this operation can sustain a certain amount of noise without losing its ability to perform equally well. Finally, we treat the problem of deconvolution of measured lineshapes from the instrument response function. We discuss the general principles of this process, and particularly the inherent difficulties present that do not allow a direct solution by standard techniques. Then, we show how a feedforward net can be used successfully to perform this operation. The examples used here are distributions similar to time decays found in optical spectroscopy of crystals, but the principles of this work hold for any such deconvolution situation.

**Mathematics Subject Classifications**: 82C32, 44A35, 94A12

**Keywords**: Neural nets; signal processing; convolution; back-propagation algorithm.

# 1 Introduction

Neural nets have been employed recently in a variety of applications, too numerous to present here an exhaustive list. They range from pattern and speech recognition to banking (loan decisions), chess playing and many more. By general consent they encompass a growing and highly promising field of scientific research. Several different models and mathematical techniques have been employed in this effort depending on the problem at hand.

In a previous paper [1] we examined the correspondence between conventional methods of function approximation (Fourier series, etc.) and the mappings that are achieved by feedforward neural networks. A number of examples were investigated ranging from performing real value reciprocal arithmetic to approximating particle identifier functions that can identify masses and charges of energetic nuclear isotopes. In the present work we extend the methodology to problems that are frequently met in experimental measurements. We investigate whether we can use neural nets for eliminating the noise that often corrupts such measurements, making it difficult to recover the pure signal that was intended to be measured.

The general structure and mechanism of the feedforward net [2] were described in detail in our previous paper [1]. Briefly, a network can be "trained" to implement a mapping of a set of inputs to a set of outputs. These exemplars may represent selected arguments of a function (inputs) and the function's value (outputs), in which case the neural network is effectively used to perform function approximation. The concept of what constitutes a function can be very broad, e.g. image identification can be considered a functional mapping of a set of pixels (arguments), to some output which can be used to classify the content of the picture.

Recently, a new method has been introduced [3] for accelerating the learning rate in the Back Propagation Algorithm by a factor of 30%. This is done by introducing a new "single-direction" coefficient in the change of the coefficients when calculating the new values. This shows the increasing interest in this method, which is the most commonly used for such problems.

The training of a network is an iterative process whereby we attempt to minimize the error of the mapping by successively refining the parameters that govern its operation. For example, a three-layer network is represented by an input layer, a "hidden" layer, and an output layer. Each layer may have a variable number of neurons (nodes). All nodes at a given layer are connected with the nodes of the next layer only, and there are no reverse or sideways connections. Each connection is characterized by a unique strength $w$. Initially all $w$ values are chosen at random. For each pattern to be learned the net is presented with the pattern and the corresponding known target. The signal of a particular pattern is presented at the input level. Then it is propagated in the forward direction only, one layer at a time. The input to each node is computed as the weighted sum, $x$, of outputs from all nodes connected to it from the previous layer:

$$x = \sum_i w_i I_i.$$

$w_i$ represents a particular connection weight from a given lower level neuron, and $I_i$ the output from that node. The output from the current node is then computed using an "activation" function that is nonlinear, differentiable, and lower and upper bounded. A popular choice for such a function is the logistic function, $y = 1/(1 + \exp(-x))$.

This synchronous operation is continued until the last layer generates the output of the net. There are various ways of defining the mapping error to be minimized. Typically it is chosen as a least square measure [2]: $E = (t \arg et - output)$.

Based on this error, the error gradient with respect to each weight is computed and the weights adjusted by projecting a fixed amount in the direction of the steepest descent. The projection procedure may use various methods to hasten convergence, i.e. Newton methods [4], conjugate gradients [5], second order methods [4], or Metropolis methods [6] to escape from local minima. The simplest adaptation of this method to the multilayer networks is called the backpropagation algorithm [2]. One such signal propagation together with the corresponding weights adjustment constitutes one cycle, and it usually takes several hundred or thousand cycles for the error $E$ to become minimal. At this point the net has learned the set of presented signals, as the output does not differ significantly from the target.

In the first part of this work we show that a neural net can learn to calculate parameters from statistical distributions; in particular temperatures from temperature-dependent energy distributions. An important property of the net is that it is able to do so whether the presented lineshapes are smooth or contaminated with noise. This is a manifestation of the nature of neural nets, which are not necessarily good for exact calculations, but are fault-tolerant (up to a certain limit), and thus can operate successfully in noisy environments.

The second part of the paper deals with a similar problem, that of deconvolution of spectral distributions. This problem in its general form has been a very old one, frequently appearing when retrieving and analysing experimental signals. These signals may be due to a variety of different phenomena in physical, chemical, biological or other disciplines. Regardless of the origin, the problem is always posed in the same way: The finally measured and recorded signal is the combination of the "pure" signal and the "instrument response" function. The problem at hand is to deconvolve these two signals in a meaningful way, so that we "subtract" the instrument function from the total signal and recover the pure signal for which the experiment was designed. The peculiarity and constant interest of this operation lies with the fact that generally speaking the direct deconvolution of any two functions is not possible (for reasons explained below), and, therefore, one has to resort to iterative methods to retrieve the signal of interest. Our technique presented here is to perform this operation utilizing neural networks. In this study we draw examples from the time decay functions in optical spectroscopy experiments, but everything included in this paper is valid and of general use wherever deconvolution situations arise.

## 2 Parameter estimation

In this application our network is effectively taught to approximate single valued functions of many variables. In particular, we examine whether neural nets may be used for parameter estimation of several types of statistical distributions that may be corrupted by various levels of noise. We take, as examples, distributions of the form:

$$f(\varepsilon) = e^{-\varepsilon/T} \text{ and } f(\varepsilon) = \varepsilon e^{-\varepsilon/T} \tag{1}$$

They represent energy ($\varepsilon$) distributions for various systems observed at thermodynamic equilibrium at some temperature $T$. Thus the first distribution is readily identified as a Maxwell-Boltzmann distribution. The second distribution is that observed, for example, for energies of neutrons evaporated from a highly excited compound nucleus in statistical equilibrium with all output channels [7]. It also may be adapted to describe energy spectra of observed light charged ions ( e.g. protons, alpha particles ).

One useful function that we could ask of our neural network is to act as a thermometer, in that when presented with some distribution of the form in Equation (1) to be able to identify the temperature $T$. We would like our thermometer to correctly identify temperatures it has not been taught before (interpolate and extrapolate), as well as respond correctly in the presence of noise imbedded within an energy spectrum.

We show that both these capabilities are attainable with the simple network employed here. Typical energy spectra of one hundred or so channels (with intensities per channel scaled to the range 0.0 to 1.0), applied as inputs to the input layer, may be accurately mapped to the correct temperature, with or without a significant amount of statistical noise (noise of zero mean and with various variances) imbedded within the spectra.

In a typical calculation we employ a network with a topology of 100:5:1 (number of nodes in the net, input:hidden-layer:output). Choosing, as a representative case, the identification of spectra with temperatures in the range of $10 > T > 1$, we teach the network to correctly identify 20 temperatures over this interval. In these examples, as well as for the deconvolution examples presented later, the weights are adjusted using Newton's method to calculate the extent of the steepest descent projections over the error surface. Sample spectra for equilibrium neutron evaporation are shown in Figure 1 for temperatures of 1.0, 5.0 and 10.0 $MeV$. The temperatures taught were actually their reciprocals since the output of the nodes at the output level is bounded between 0 and 1. The neural thermometer was then presented with 100 untaught spectra uniformly covering this range; with or without statistical noise. Sample spectra with a signal to noise ratio of 20:1 are indicated in Figure 2 for the same temperatures as above. For the noise-free spectra, the temperatures were identified with less than 0.1 % error. The noisiest spectra had their temperatures correctly identified within 2 % error on the average. This may be seen in Figure 3, where we plot the temperature error versus temperature (the difference between the true temperature and that estimated by our neural thermometer) for untaught energy spectra at three levels of progressively increasing statistical noise (indicated by signal to noise ratios of 1000:1 to 10:1). These temperature estimates are as good as could be developed by a smoothing least square fitting procedure.

Similar accuracies were obtained in estimating temperatures from the Boltzmann distributions for spectra over the same temperature range. While we have restricted our estimating capabilities to these temperature ranges and distributions, the method extends to other cases. The choice of parameter range to be taught should correspond to that expected to be met, given the physics of the process under study. Thus, the range of temperature for the neutron energy spectra considered here corresponds to that observed in nuclear reactions for excited emitting nucleus with excitation energies in the

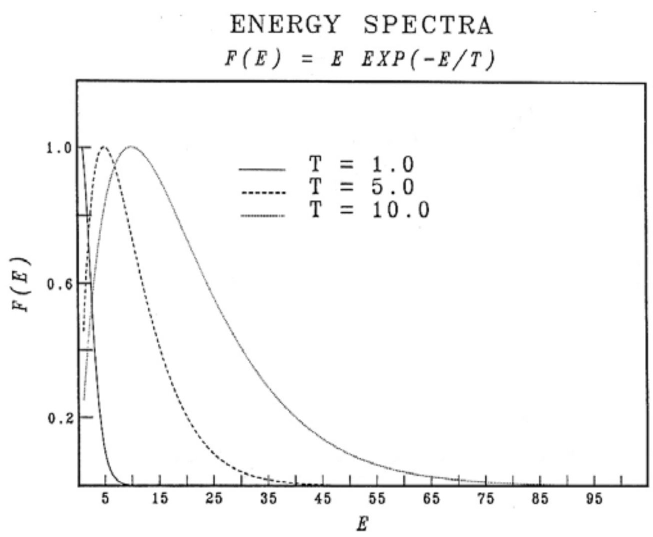


Figure 1: Sample energy spectra for equilibrium neutron emission as a function of excited emitter's temperature.

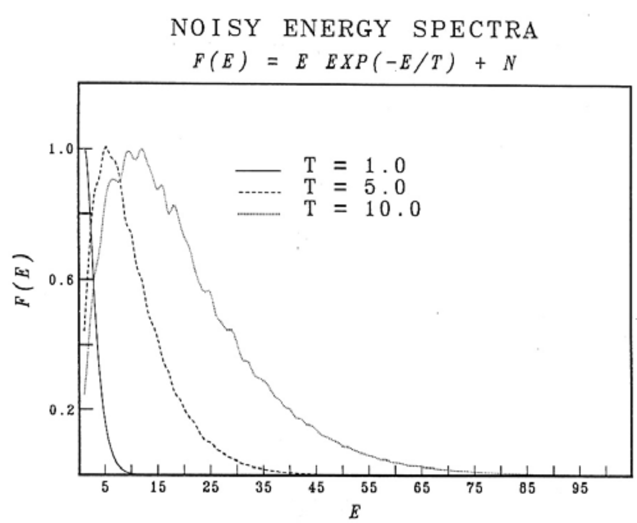


Figure 2: As in Figure 1 but with inclusion of statistical noise.

range of several $MeV$ to around 100 $MeV$. These are characteristic of typical accelerator experiments.

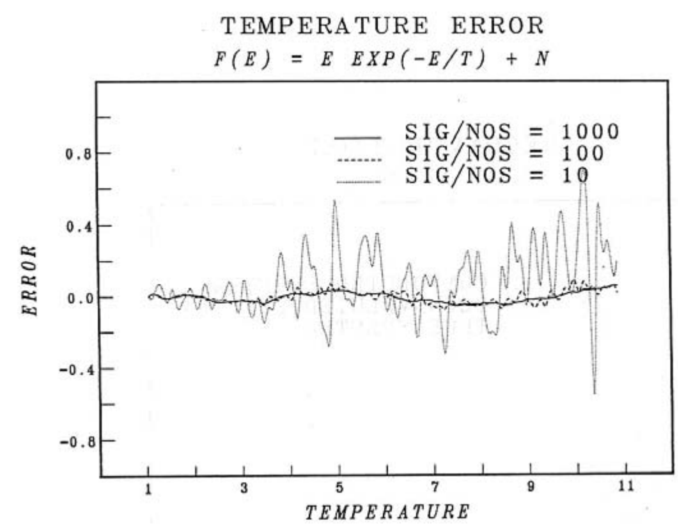


Figure 3: Error of computed temperature for various levels of signal to noise.

# 3 Convolution and deconvolution

Given two functions, $x(t)$ and $h(t)$, then the convolution of $x$ and $h$ is given by $y(t)$ as:

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) h(t-\tau)\, d\tau \tag{2}$$

The symbol $*$ describes the convolution operation. When we do not have the functional form of $x(t)$ and $h(t)$, as it is usually the case, but only a discrete spectrum of each made of a number of points, then the discrete convolution is given by the summation:

$$y(kt) = x(kt) * h(kt) = \sum_{i=0}^{N-1} x(iT)\, h[(k-i)T] \tag{3}$$

where both $x(kt)$ and $h(kt)$ are periodic functions with period N,

$$x(kt) = x[(k+rN)T] \qquad r = 0, \;\; \pm 1, \;\; \pm 2, \;\; ... \tag{4}$$

$$h(kt) = h[(k+rN)T] \qquad r = 0, \;\; \pm 1, \;\; \pm 2, \;\; ... \tag{5}$$

In continuous or in discrete form this operation physically corresponds to making one of two functions into a "moving window" which samples the first function in its entire range. Thus the $y(t)$ function is not simply a point-by-point product of $x(t)$ and $h(t)$, but each point depends on the entire length (all points) of the convolved functions.

The convolution of two functions can be succinctly related to their Fourier transforms [8, 9]. This is because of the Convolution Theorem, which states that if $x(t)$, $h(t)$, and $y(t)$ are the functions of Equation (2) , and if $X(f)$, $H(f)$, and $Y(f)$ are the corresponding Fourier transforms of these three functions, then it holds that:

$$Y(f) = X(f)H(f) \tag{6}$$

which means that the Fourier transform of the convolution is just the product of the individual Fourier transforms. This is valid for both continuous and discrete functions. Thus, in order to find $y(t)$ in Equation (2). we simply calculate $X(f)$ and $H(f)$, take the product $Y(f)$, and finally take the inverse Fourier transform of $Y(f)$, which gives directly $y(t)$.

The obvious way to deconvolve two functions would be to perform the inverse operation, i.e. first measure $h(t)$ and $y(t)$, then calculate the Fourier transforms $H(f)$ and $Y(f)$, and finally take:

$$X(f) = Y(f)/H(f)$$

and the inverse Fourier transform of $X(f)$ in order to recover $x(t)$. This series of operations works well when the measured signal is not corrupted with an additional noise component. But in experimental situations as discussed here, when even a seemingly trivial degree of noise is added to the convolution process, rote application of the Fourier transform method is usually hopelessly inadequate [10]. Typically, high frequencies in the noise are amplified out of all proportion when divided by the small frequency response of the instrument at these frequencies. Recovering the spectrum thus requires a certain amount of empirical smoothing of the experimental spectra and/or the use of frequency filters such as the optimal (least square Weiner) filter. Using such filters often requires some ingenuity from the experimentalist in formulating the underlying noise model. Other techniques have also been employed with the use of a "regularizer", which is some function that performs successive estimates of the function sought. This method works iteratively, and the successive estimates are convolved in each cycle and compared to the observed signal. Attempts are made to minimize the difference of the observed from the regularizer function. The only problem is that there is no universal regularizer, and it must be chosen separately for each problem. To find what is the best choice is no easy task. We thus see that the obvious path of functional smoothing before deconvolution is difficult or sometimes impossible, and there is still a need for a general technique that would work adequately for any noise form or level (up to the inherent quality of the signal). Thus, our main objective in attempting to use neural networks to perform deconvolution is to see if they can effectively devise a universal noise filter and recover the original spectra in any deconvolution situation.

## 4 Deconvolution with neural networks

The model problem we first consider is to teach the neural network to deconvolve a small set of exponentially decaying functions that have been convolved with an instrument function that is roughly Normal or Gaussian in shape. This problem has a realistic counterpart in a laser induced experiment [11, 12] wherein a crystal is excited by a finite duration pulse and the exciton decay from the process is measured. Due to the complicated physics present de-excitation is not a simple exponential. A typical halflife for the signal decay is of the order of $10 - 100$ $ns$, when excited by a laser pulse of about 5 $ns$. The decay intensities are measured using a time to amplitude converter, and the measured spectra extend over 256 channels corresponding to 256 time intervals. We thus require 256 input and output nodes. The time duration of the laser signal is approximately 50 time units in length and thus we choose the number of hidden nodes to be 50. The network was first trained to map 5 non-noisy convolved to deconvolved spectra, where the deconvolved spectra were of the normalized form,

$$s(x) = e^{-x/T} \tag{7}$$

for 5 values of $T$ over the range $65 > T > 1$. These taught spectra were correctly deconvolved to within less than 1% error. The ability of the network to deconvolve untaught spectra over this range of $T$ was found to be successful within 1-2% error. Indeed, it was found capable of deconvolving spectra that were combined weighted sums of exponentially decaying spectra. Indicated in Figure 4 is the accuracy with which deconvolution of a combined spectra, composed of the equally weighted sum of two untaught spectra with decay constants equal to $T = 23.76$ and $T = 37.7698$, was achieved. The deconvolved spectrum calculated by the neural net is indistinguishable from the true spectrum. Shown in Figures 5 and 6 are the neural network's deconvolved spectra and true spectra for the same combined spectra as above, but with significant levels of white noise added to the convolved spectra (signal to noise ratios of 10:1 and 5:1 respectively). The noisy convolved spectra are also indicated. The neural network was taught the same examples as above and one additional mapping. This latter exemplar was a small constant spectrum as input and the output was specified to be zero. This was intended to be the uniform white noise spectra that is to be filtered from the real spectrum. It may be appreciated that the neural network has done a superior job at recovering the undistorted spectra. Inclusion of the noise exemplar improves the recovered spectrum by approximately a factor of 2, compared to not including it in the set of examples. This indicates that the network has indeed learned to act as a filter to some degree.

For comparison, we may consider the standard Fourier transform based sig-

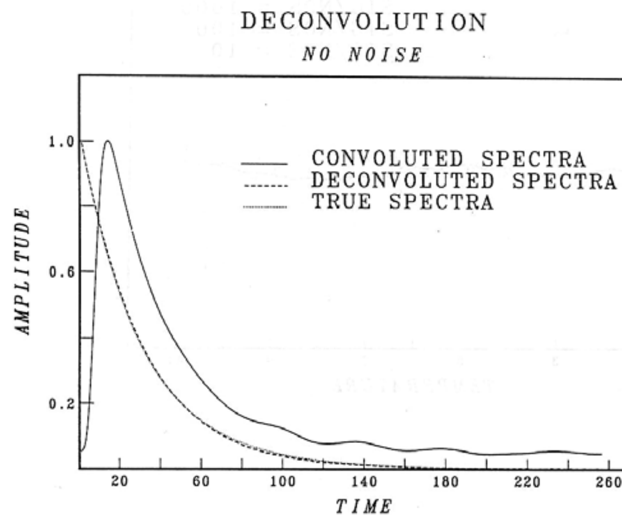


Figure 4: Convolution and deconvolution spectra for sum of two spectra with $T = 23.76$ and $T = 37.7698$.

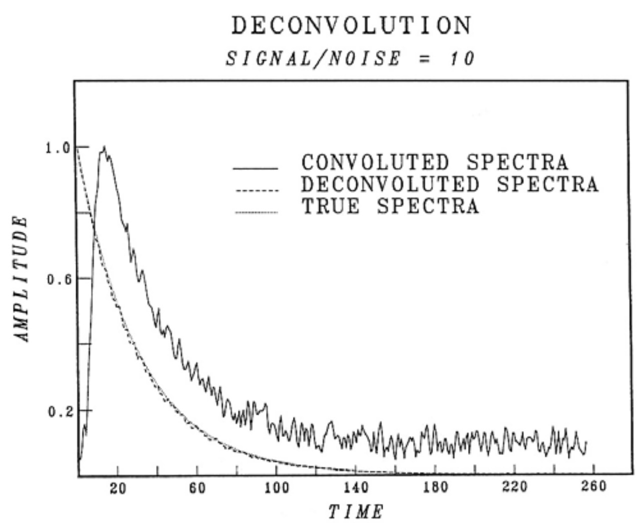


Figure 5: Same as in Figure 4 with statistical noise. Signal to noise ratio of 10:1.

nal processing techniques discussed above. Shown in Figure 7 are the normalized power spectra (plotted on a logarithmic scale) for the instrument function (the convoluting kernel), the noise free combined exponentially decaying spectra, and the power spectra of the resulting convolved spectra. In the absence of noise, the standard algorithm (Equation (2)) recovers the original spectra

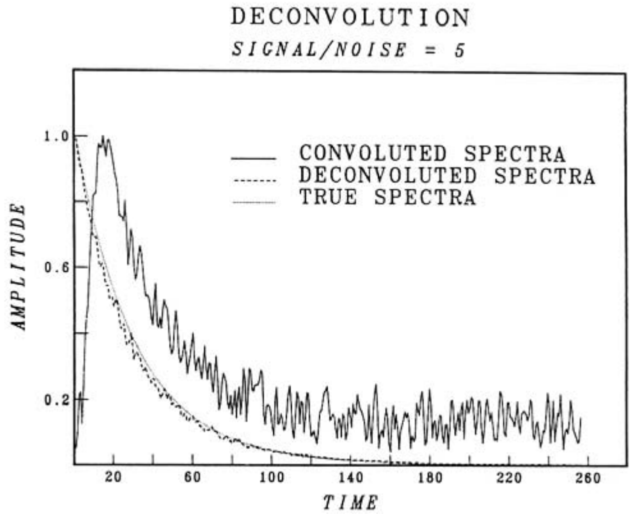


Figure 6: Same as in Figure 5 with signal to noise ratio of 5:1.

exactly, as required by the Fourier transform convolution theorem. However, when white noise is added to the convolved spectra, the deconvolution is much worse. Figure 8 shows the power spectra and the convolved spectrum when white noise has been added with a signal to noise ratio is 10:1. Rote application of the Fourier transform method, without any pre-smoothing or frequency filtering, results in the completely meaningless deconvolved spectra shown in Figure 9. If smoothing of the convolved spectra is performed (averaging over a three channel wide window) and an optimal Weiner filter is used, then the deconvolution can be performed with somewhat better results, as shown in Figure 10. However even here, where significant effort has been made to remove the white noise, it is evident that the neural network deconvolution has performed in a much superior fashion, as is evident from Figure 5, where the same noise corrupted convolved spectrum is treated.

# 5 Discussion and Conclusions

Our intention in this work has been to examine whether neural nets can provide an alternate means for filtering noise in experimental signals. We have provided two examples of such cases: parameter estimation and deconvolution of signals from response functions. This method may be superior to conventional methods because it does not depend on the details of the signals studied. In particular, the performance of the method does not significantly degrade in the presence of high levels of noise, and thus neural nets are particularly amenable to problems such as signal deconvolution, where minute amounts

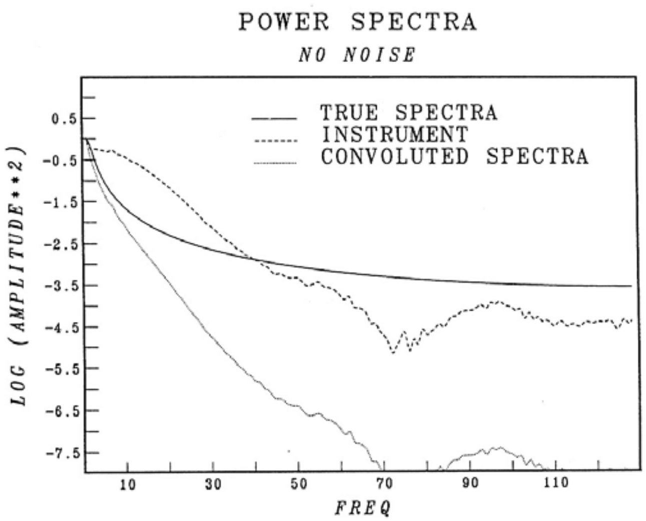


Figure 7: Normalized power spectra of convoluted and deconvoluted spectra without noise.

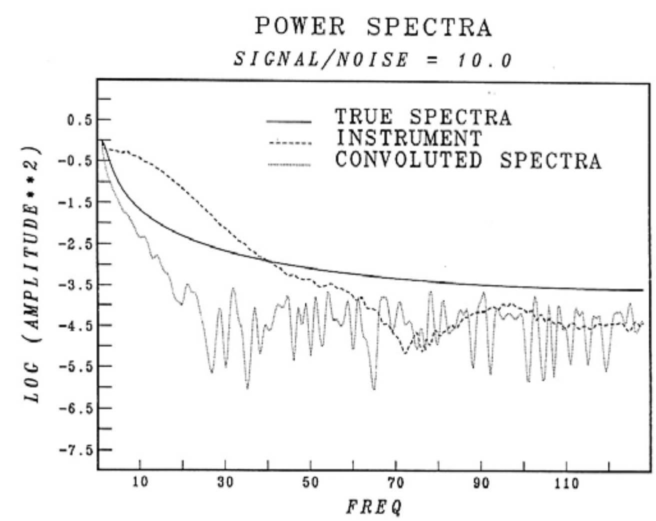


Figure 8: As in Figure 7 but with white noise with signal to noise ratio of 10:1 added after convolution.

of noise may pose serious difficulties for traditional methods. The principal reason behind this is that the summing of signals that feed a neuron provides an internal averaging of the total input and tends to cancel individual fluctuations. Similarly, the aggregate input in turn has fluctuations damped out by the nature of the "squashing" property of the activation function. The com-

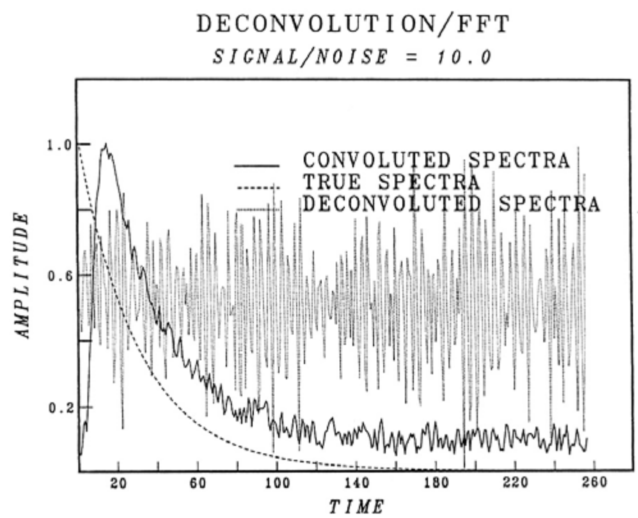


Figure 9: Fourier transformed deconvoluted spectra of noisy spectra of Figure 8.

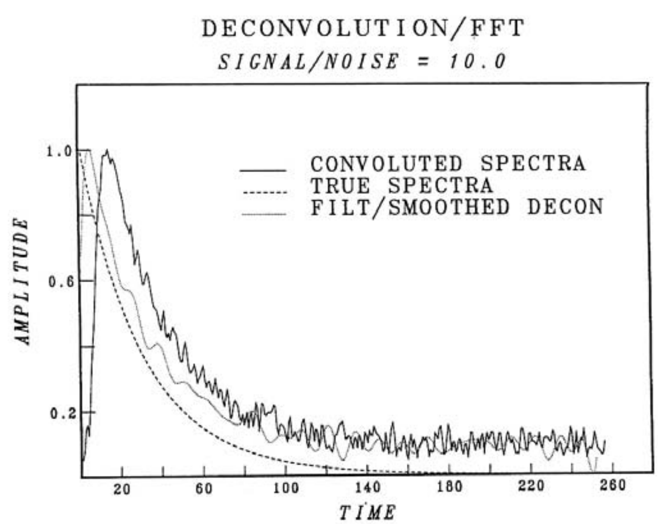


Figure 10: Fourier tranform based deconvolution of noisy spectra of Figure 8 with window smoothing and Weiner Filter.

bined effect of these operations through several layers of neurons provides for a good operation in the presence of noise.

Due to the nature of neural nets the results they give are not exact, but we consider this method to be successful in these problems if they can provide a solution within a few % error. Developing networks with greater accuracy most often corresponds to simply carrying out the training over longer periods (more iterations) and/or increasing the number of hidden nodes. However, the asymptotic best results achievable are difficult to predict and the expense in computer time may become prohibitive.

Neural network training is equivalent to expanding a function in terms of basis functions, where each hidden node is a basis function described, for example, by an optimally parameterized logistic function (or other such lower and upper bounded function). The backpropagation algorithm is an effective means of achieving the parameterization. However, it is not clear if it is the best algorithm to achieve this end. The patterns that must be taught to a net must usually be of the same nature as the patterns for which we will require the net to perform the deconvolution. In other words there must be some rough prior knowledge of the type of signal expected.

Other investigations are similarly finding the neural network approach useful for these types of problems. Attempts [13, 14] have also been made to deconvolve experimental functions using neural network methods. In a preliminary study [13] a CMAC (Cerebellar Model Arithmetic Computer) was successfully used to deconvolve a decaying sinusoid impulse function convolved with low level white noise. However, it was less successful at the lower signal to noise ratios considered here. A similar type of net [15] was also used for the deconvolution (separation) of two overlapping chromatographic peaks, but these signals usually contain no noise. Finally, an algorithm [14] was devised that is suited to implementation in a neural net type electronic circuit, and also performs deconvolution satisfactorily. It utilizes a steady-state feedback type of network and it is able to perform well also in the presence of noise.

**Acknowledgments.** This project was supported by the Research Committee of Technological Education Institution (TEI) of Thessaloniki, and also by a grant of the GGET (Greek Secretariat of Research and Technology) in conjunction with ESF in the frame of international organizations.

# References

[1] D. Logan and P. Argyrakis, P. Function Evaluation with Feedforward Neural Networks, International Journal of Computer Mathematics, **67** (1997), 201.

[2] D.E. Rumelhart and J.L. McClelland, Parallel Distributed Processing, Vol. 1 and 2, MIT Press, Cambridge (Mass), 1986.

[3] S. Sotirov, A method of accelerating neural network learning, Neural Processing Letters **22** (2005), 163-169.

[4] R. Hecht-Nielsen, in Proceedings of the International Joint Conference on Neural Networks, **1** (1989), 593.

[5] A.H. Kramer and A. Sangiovanni-Vincentelli, Efficient Parallel Learning Algorithms for Neural Networks, in: Advances in Neural Information Processing Systems 1, California, 1989, pp 40-48, Morgan Kaufmann.

[6] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by Simulated Annealing, Science **220** (1983), 671.

[7] J.M. Blatt and V.F. Weisskopf, Theoretical Nuclear Physics, Wiley, New York, 1966.

[8] E.O. Brigham, The Fast Fourier Transform, Prentice Hall, Englewood Cliffs, NJ, 1974.

[9] G. Horlick, Fourier Transform Approaches to Spectroscopy, Anal. Chem. **43**, 61A (1971).

[10] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Cambridge UP, 1986.

[11] P. Argyrakis, PhD Thesis, University of Michigan, 1979.

[12] P. Argyrakis and R. Kopelman, Chem. Phys. **78** (1983), 251.

[13] F.H. Glanz and W.T. Miller, in Proceedings of the IEEE 1989 International Conference on Accoustics, speech, and signal processing, **4** (1989), 2349.

[14] S. P. Fraleigh and S.M. Cramer, Abstracts of Papers of the American Chemical Society, **199** (1990), 158.

[15] C.R.K. Marrian and M.C. Peckerar, in Proceedings of the IEEE First International Conference on neural networks, **3** (1987), 749.

**Received: September 11, 2006**