ELSEVIER

Nuclear Instruments and Methods in Physics Research A 411 (1998) 467–474

NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A

A neural network approach to spatial reconstruction in the CTF detector

G. Alimonti*, S. Magni

Dipartimento di Fisica, Universitá di Milano e INFN, Via Celoria 16, 20133 Milano, Italy

Received 25 November 1997; received in revised form 29 December 1997

Abstract

Artificial neural networks may in some cases present a new important approach to information processing. We have investigated whether the accuracy offered by this technique is good enough to extract physical information from the signals coming from an unsegmented large volume liquid scintillator detector.

In particular, we wanted to understand whether this method is well suited to be implemented in the Borexino detector for monitoring or for on-line event selection purposes. The results obtained on data from a smaller scale Borexino-like detector, implementing a neural network algorithm on a sequential scalar computer, have been compared to those of a standard best-fit procedure. © 1998 Elsevier Science B.V. All rights reserved.

PACS: 07.05.Mh

Keywords: Neural network; Spatial reconstruction

1. Introduction

The first phase of the Borexino experiment [1] is the Counting Test Facility, CTF [2], that has been set up in the Hall C of the LNGS. It consists of 5 m^3 of liquid scintillator shielded by 1000 tons of high purity water and viewed by 100 photomultipliers equipped with truncated string cones for light collection. The related read-out electronics is composed of 64 ADC and TDC channels: the relevant pieces of information are the number of collected photoelectrons, giving the total energy released in the detector, and the photon arrival time at each photomultiplier that allows the spatial reconstruction of the event, fundamental in the Borexino experiment to define the fiducial volume. The spatial reconstruction is performed by a standard likelihood method working with the relative arrival times of the hit photomultipliers?

A likelihood function, built in the multidimensional space of the arrival times and spatial coordinates, gives the probability that the arrival time pattern is generated by an event in the given position. For each event this method searches for the spatial point that maximizes this probability.

* Corresponding author. Tel.: 39 2 2392 363; fax: 39 2 2392 617; e-mail: gianluca.alimonti@mi.infn.it.

PII: S0168-9002(98)00295-2

This is a statistical approach to the spatial reconstruction problem, and finds the result with the highest probability to be correct, but has the disadvantage that it requires a long computational time. In this framework, the spatial reconstruction can be seen as a mapping between a set of time inputs and the event coordinates, a typical pattern recognition problem. The neural network approach is well suited for this problem and its fast computational time, a weakness of the standard method, pushed us to investigate the possibility to use this technique.

In this paper we present a brief study of the essential features of an approach based on the simulation of a neural network on a sequential scalar computer. The paper starts with a brief introduction to the neural network approach and a check that the CTF spatial reconstruction problem can be solved with this technique. We then present some peculiarities of the implementation of our net for CTF and finally we compare the net performance with the standard likelihood method on source data taken in several CTF runs.

2. The neural network approach

An artificial neural network can be thought of as a sort of "black box" processing system; its operational capabilities allow the reproduction of an application $X \rightarrow Y$ between two sets of vectors. In our case the input set is the space of possible arrival time configurations while the output is the three-dimensional physical space. In a neural network the basic units are called neurons. A neuron has several input and output connections and the weighted sum of all the signals received by a neuron generates its response through an activation function:

$$y = g\left(\sum_j w_j x_j + \vartheta\right),$$

where x_j and y are the input and output values, ϑ is a thresholding potential and w_j are the weights. The activation function g is very often chosen as a sigmoidal-like function such as

$$g(u) = \frac{1}{1 + \mathrm{e}^{-u/T}}\ .$$

The constant T sets the gain of the activation function. With this simple neuron, various types of networks and architectures can be built; one of the most widely used and well suited for our application is the feed-forward, multilayer neural network with supervised training. In this configuration the neurons are grouped in layers: an input layer, where data are presented, hidden layer(s) and an output layer where results are given. The output of each neuron is connected to the input of the neurons of the following layer; in our configuration there are three neurons in the output layer that give us the x,y,z coordinates of the reconstructed event position.

In principle, one hidden layer is sufficient to fit any continuous function [3] but it may well be more practical to interlayer the network with more than one hidden layer.

Once the weights are set in the training process, the network architecture is completely determined. There is a variety of training algorithms: in the present case we have followed the method known as back-propagation. A set of Monte Carlo simulated input data is presented to the net with the corresponding output; for each input pattern the energy function is evaluated:

$$E = \frac{1}{2}\sum (O_{\mathrm{NN}} - O_{\mathrm{M_c}})^2$$

where O_{NN} is the computed network output for a given pattern and $O_{\mathrm{M_c}}$ is the Monte Carlo generated output for the same pattern. The energy function is computed by the net and its value is propagated backwards to modify each weight in order to be minimized. The updating equation is

$$\Delta w_{ij}^{(k+1)} = -\frac{\partial E}{\partial w_{ij}}\eta + p\Delta w_{ij}^{(k)},$$

where $\Delta w_{ij}^{(k+1)}$ is the $(k+1)$st update of the weight w_{ij} and η is the learning parameter, representing the step length on the error surface that is defined by the value of the error as a function of all weights. The last term is a momentum term which stabilizes the training by avoiding oscillations. The strength of the damping is set by the parameter p which should be between 0 and 1. The updating of the weights can be done for every presented pattern, by grouping patterns together or just once after the complete training set, called one epoch. The smaller the number of patterns the energy function is averaged over, the faster, but more unstable too, the training process can be.

The training starts with randomly selected weights and relatively high values of the learning rate and momentum; the factors are changed during the training over several thousands of epochs, until the error function reaches a minimum on a validation set, a Monte Carlo generated set different from the learning set.

It is important that the input variables are of the same order of magnitude and range in order to obtain stable convergence towards a minimum: the simplest method to obtain this is to renormalize the input variables to an interval [0,1]. Furthermore, in order to avoid local minima, it is recommended to pick the input event at random from the simulated set and, to prevent overfitting, to have the number of simulated learning patterns about one order of magnitude higher than the number of weights in the net. Overfitting is the problem that arises when there are so many weights in the net with respect to the number of learning patterns that the net becomes very specialized in fitting "too well" only those patterns, losing in such a way its generalization capability.

3. Neural network for CTF

3.1. Are NN good for CTF?

Before approaching the reconstruction of "real data" taken during CTF runs, we wanted to make sure that the NN method is well suited for the CTF spatial reconstruction problem and to understand what performance we could expect. We then built a feed-forward NN keeping in mind what has been said in the previous paragraph, and applied it to CTF Monte Carlo generated data.

A set of 20 000 point-like, 1 MeV Monte Carlo events was generated uniformly in the sensitive region of the detector: data from each event are the generated position and the photon arrival time at each photomultiplier.

We decided on a three-layer NN (36-36-3 neurons per layer) with a sigmoidal activation function having the output range in [0,1]: we then needed to fix the space where we allowed the NN to reconstruct. This implies some limitations that will be shown and resolved in the optimizations of the NN for the CTF data presented in the next paragraph.

The choice of 36 inputs was taken in view of the final application: in the real detector there are 36 "double" channels, in which two photomultiplier analog outputs are summed, and 28 "single" channels. We coupled each single photomultiplier with the closest pair to have a reasonable and limited learning time that would allow us to try different solutions (the four pairs on the top of the detector and the four pairs on the bottom do not have any close single photomultiplier).

To fulfill the input normalization requirement, we applied a time cut common to all the events and the training was performed using the back-propagation algorithm. We started the learning session by first grouping one tenth of the training patterns; after reaching a good degree of convergence, we performed the updating only after presenting the complete training set. We stopped the training when the error function behaviour became flat, that means the net was no longer learning, as shown in Fig. 1.

The Fortran package JetNet 3.4 [4] was used for the NN computations and a graphical interface was developed that allowed us to interactively monitor and modify the relevant NN parameters [5].

The results of the NN algorithm are shown in Fig. 2: the difference between the reconstructed and the generated coordinates is plotted. It can be noticed that the resolution on the z coordinate is different from the resolution on x and y: this is true also for the likelihood method reconstruction code and comes from the cylindrical (and not spherical) symmetry given to the photomultiplier distribution.

We remind that the purpose of this Monte Carlo study was to verify the possibility to use NN for the spatial reconstruction in CTF and not to build the best possible net to reconstruct Monte Carlo data: in this frame we conclude that the answer is definitely encouraging.

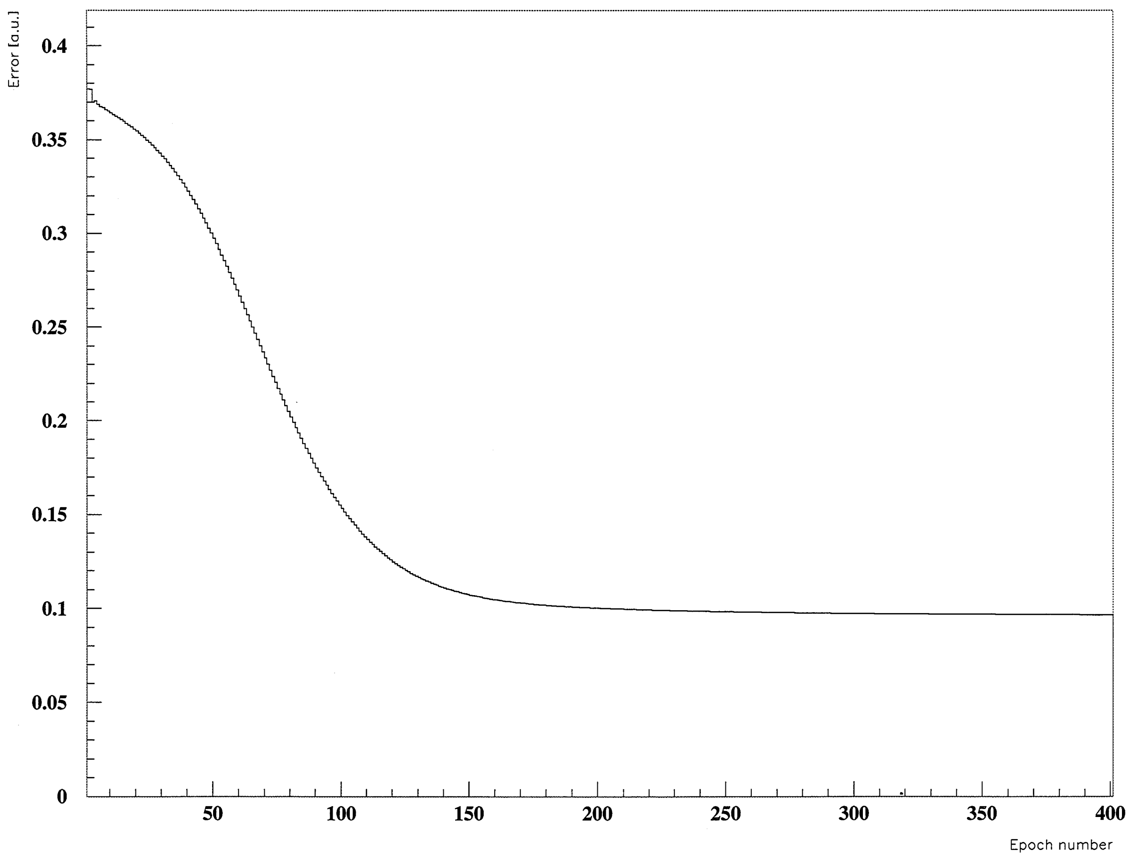

Fig. 1. Training error as a function of epoch number.

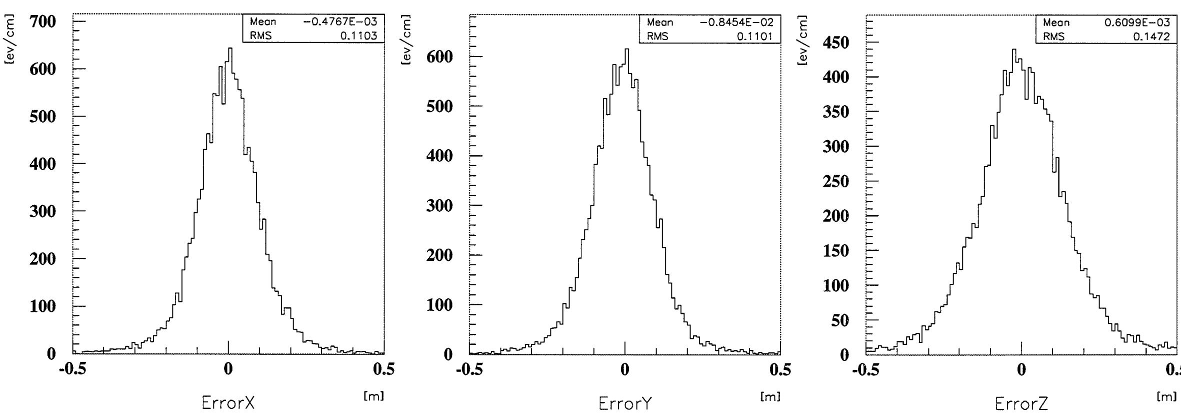

Fig. 2. MC events reconstructed by a 36 input, three-layer NN with a sigmoidal activation function.

3.2. NN for the CTF detector

The next step was to preprocess the data, selecting and organizing the significant information, and to optimize the NN for the "real" reconstruction.

3.2.1. Time cut

Due to the timing properties of the liquid scintillator used in CTF, part of the emitted light is absorbed and re-emitted by the scintillator itself before reaching the photomultipliers. This has the effect to delay the arrival time of the photons that undergo this process and, moreover, since the re-emitted light is isotropic, the information valid for the spatial reconstruction is completely lost. Therefore, a time cut selecting the prompt information improves the reconstruction.

A constant time cut common to all the events would not be appropriate because the arrival time spread of the photons is event position dependent. A dynamical time cut has been defined:

$$t_{\text{cut}} = \bar{t} + 2.5\sigma_t,$$

where $\bar{t}$ is the mean value of the event arrival time distribution and σ_t is the width of the same distribution. The normalization value was chosen so that most of the inputs are in the range [0,1], but not all; we could have chosen the highest possible time value, thus strongly compressing most of the inputs, but this would fail to exploit the full NN input dynamic range.

A drawback of this time cut is that the convergence towards the minimum is more critical: we have to be careful during the training process, monitoring and modifying the learning parameters via the graphical interface developed.

3.2.2. Linear activation function

Using the sigmoidal activation function with the output range [0,1], the space where an event can be localized has to be selected; for the Monte Carlo results we have shown, we allowed the reconstructed event to be in a 2 m radius sphere with the center common with the Inner Vessel (I.V.). The Monte Carlo generated events were in a 1 m sphere, in the I.V.; by reconstructing in the same limited region, due to the finite resolution of the detector, the events would be statistically pushed inside. Furthermore, limiting in advance the space where events can be localized, would force any spurious arrival time combination to be inside the predefined region.

To overcome this limitation we implemented a linear activation function: the outputs are completely free running and predefinition of the possible space is no longer required.

We noticed that much more care needs to be taken during the training session since it is more critical now that the energy function can diverge. The resolution performance is equivalent to that obtained with the sigmoidal function and a slight improvement is noticed in the computational time. The results will be shown in the next sub-section after the implementation of a further simplification.

3.2.3. Two-layer NN

As previously stated [3], one hidden layer is enough to fit any continuous function so perhaps the CTF spatial reconstruction problem can be solved, or well approximated, with a two-layer NN.

In Fig. 3 Monte Carlo events reconstructed by a two-layer NN with a linear activation function are shown: comparing the plots with Fig. 2, no degradation in the resolution is evident. On the other hand, with this simpler NN configuration, the training process is much faster having about 100 weights to be optimized instead of about 3600. Advantages are evident both in the learning phase, allowing several tests to be performed in the same day, and in the computational time (a table summarizing the time performance will be shown later).

A partial justification of this simplification comes from the resolution of a similar two-dimensional problem. Consider a "light source" in the plane and three photo-multipliers of known positions. If the distances of the light source, or the arrival time of the photons, from the photomultipliers are given, then it is possible to show that the coordinates of the light source can be written as a linear combination of the given distances with weights independent of the light source position. This solution has the same form given by a two-layer NN, with a linear activation function.

To conclude this part where the NN for the data reconstruction has been defined, we have to take into consideration that unfortunately, at the time the source data were collected, the CTF had been

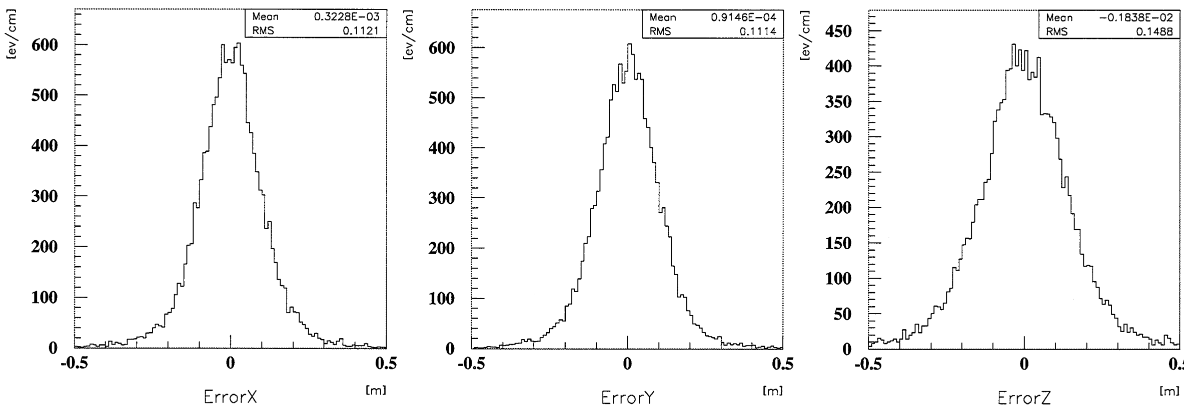

Fig. 3. MC events reconstructed by a 36 input, two-layer NN with a linear activation function.

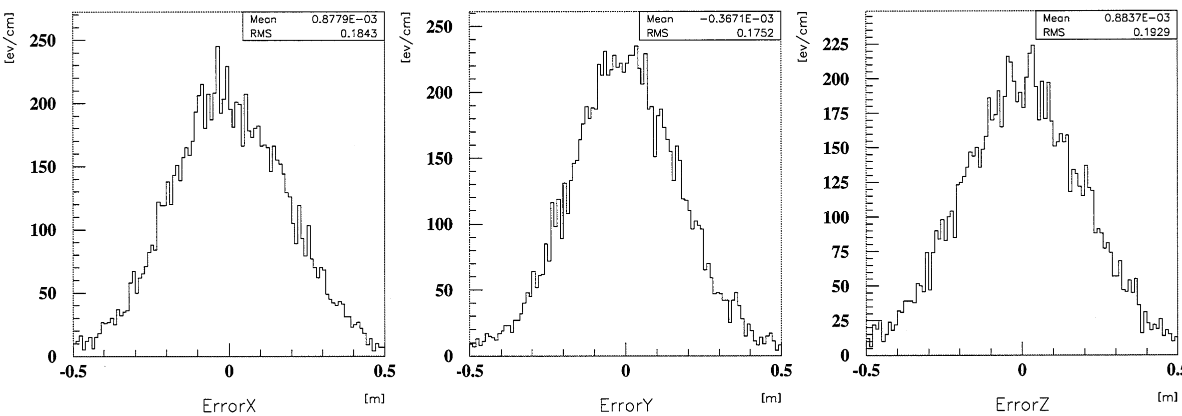

Fig. 4. MC events reconstructed by a NN optimized on a real working condition of the detector at the time the source data were collected.

running for a while and the detector itself had been damaged by some unexpected accidents. So the results we show could only exploit the information coming from about half of the photomultipliers, 50% of the installed ones being dead.

The Monte Carlo was run in such a damaged condition. The NN was adjusted to cope with this degraded situation where several channels and even several inputs of the original 36 input NN were dead. The best condition was found giving the NN all the electronic channels that were alive as independent inputs; the final configuration has then 41 inputs in the first layer.

These new Monte Carlo events have also been generated with an 820 KeV energy to be in the same condition as the source data that will be presented in the next section; the NN reconstruction is shown in Fig. 4. The worsening of the spatial resolution is evident. Furthermore, due to the scattered position of the dead photomultipliers, the symmetry of the detector was not preserved.

4. Results

The results given in this section were obtained by reconstructing source runs where ^{222}Rn was inserted in a few centimeter sphere quartz vial filled with the same scintillator used in CTF, acting as a point-like light emission. The source was moved around in the CTF scintillator: located in the center of the detector, close to the nylon surface of the I.V. and in several points spread out through the whole sensitive volume, for a total of about 30 different positions. The average number of hit channels was about 80% of the working ones, depending on the source position.

The ^{222}Rn decay chain has a distinctive signature in the coincidence ^{214}Bi–^{214}Po: first a β decay with end point at 3.23 MeV, followed by a 7.688 MeV α decay with half life 164 μs. Due to the α quenching measured in our scintillator [6] the 7.688 MeV ^{214}Po decay has an equivalent light production to a β energy release of 820 KeV. Due to the strong tag of the energy requirement of the α decay together with the time correlation of a previous β decay, it is very easy and efficient to discriminate this event from the background.

The NN performance was tested by reconstructing the 820 KeV energy equivalent α decay. Results are shown in Fig. 5, where the mean values of the source positions reconstructed by the NN are compared with the positions reconstructed by the standard likelihood method algorithm. The mean values of the two methods differ by only a few centimeters, without any significant systematic shift.

The NN resolution in reconstructing each single source event is of the same order as the NN resolution expected from the Monte Carlo simulations, as shown for one particular source in Fig. 6 and summarized for the whole source set in Fig. 7.

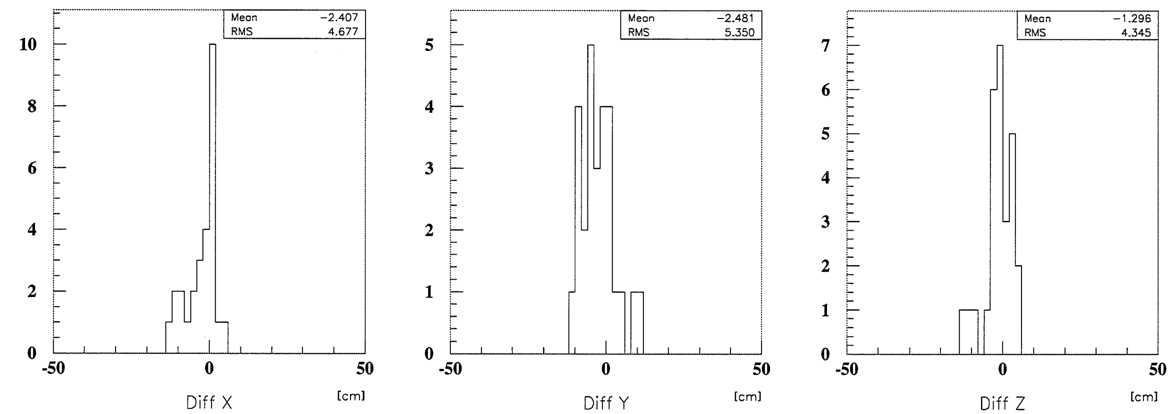

Fig. 5. Difference between the source mean values reconstructed by the NN and by the likelihood-method code.

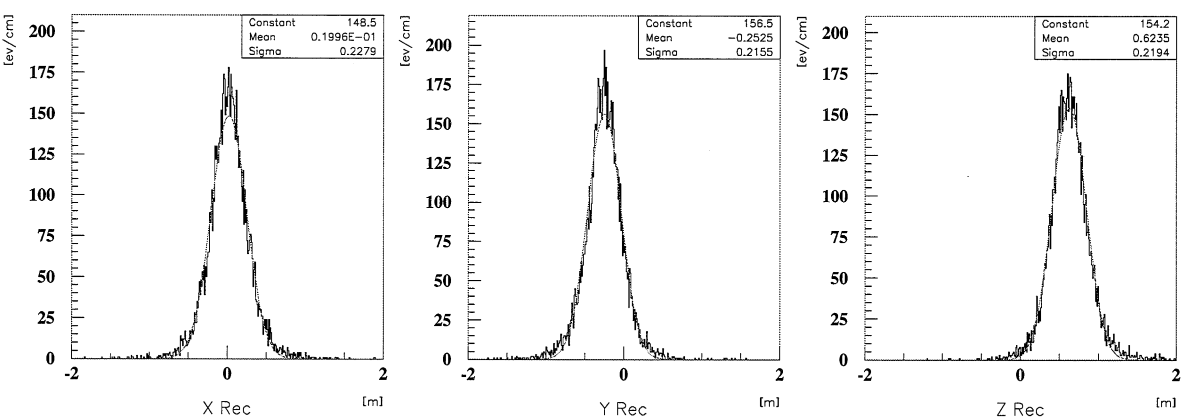

Fig. 6. A source position reconstructed by the NN.

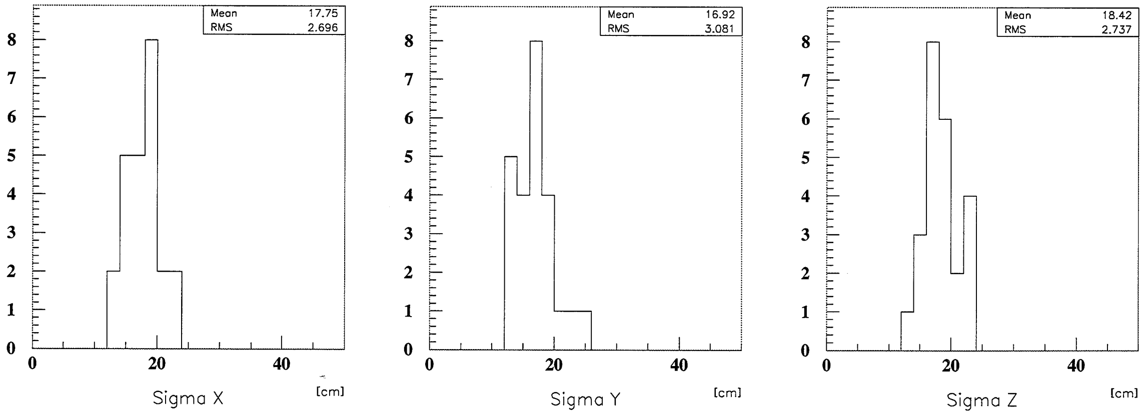

Fig. 7. Resolution in reconstructing different source positions with NN.

Table 1
Time performance

	I/O (ms/eV)	Reconstruction time (ms/eV)
NN	2	0.21
Likelihood	2	180

It can be noticed that the resolution averaged over the source set is about 18 cm, not far from the standard likelihood method algorithm resolution that is about 14 cm.

Finally, the time performance is summarized in Table 1 for both the NN implementation and the standard likelihood method algorithm. This time performance is evaluated on a 500 Mhz Digital Alpha CPU. It can be noted that the I/O time, reading the data from disk, is now by far the dominant contribution in the NN approach that, for the reconstruction, is almost three orders of magnitude faster than the standard likelihood method code. For a monitoring or on-line application purpose, the I/O time will not be such a limitation and it will be possible to fully exploit the time performance of the NN.

5. Conclusions

Given the good compromise shown by NN on CTF data between spatial resolution and processing time, we think that this method is well suited to be implemented for the Borexino detector for monitoring or on-line event selection purposes.

This analysis has been performed on 820 KeV energy equivalent events. It should be extended to the whole energy spectrum or at least to the range of ν interest in Borexino that is from 250 to 800 KeV [1]. Due to the damaged conditions of the CTF detector at the time the source data were collected, we do not think it is worthwhile to push further this test. It should also be noted that, grossly speaking, having half of the photomultipliers dead it is almost like working at half the source energy and that this is not far from the lower end of the energy interval of interest. This gives us confidence that the spatial reconstruction is feasible with NN in the whole energy spectrum of interest, with good resolution and time performance.

Acknowledgements

We thank Prof. G. Bellini for giving us the opportunity of approaching and testing this new technique within the Borexino project.

References

[1] G. Ranucci, Nucl. Phys. B (Proc. Suppl.) 32 (1993) 149.
[2] G. Alimonti et al., Nucl. Instr. and Meth. A, 406 (1998) 411.
[3] G. Cybenko, Meth. Control Signals Systems 2 (1992) 303.
[4] JetNet – A versatile artificial Neural Network package CERN-TH.7135/94.
[5] Stefano.Magni@Mi.Infn.It.
[6] F. Elisei et al., Nucl. Instr. and Meth. A, 400 (1997) 53.