

## A NEW APPROACH TO IMAGE RECONSTRUCTION IN POSITRON EMISSION TOMOGRAPHY USING ARTIFICIAL NEURAL NETWORKS

A. BEVILACQUA, D. BOLLINI, R. CAMPANINI, N. LANCONELLI

*Department of Physics, University of Bologna and INFN Bologna, via Irnerio 46  
40126 Bologna, Italy  
E-mail: bevila@bo.infn.it*

M. GALLI

*ENEA -Bologna, via Don Fiammelli 2  
40129 Bologna, Italy*

Received 23 October 1997

Revised 5 December 1997

This study investigates the possibility of using an Artificial Neural Network (ANN) for reconstructing Positron Emission Tomography (PET) images. The network is trained with simulated data which include physical effects such as attenuation and scattering. Once the training ends, the weights of the network are held constant. The network is able to reconstruct every type of source distribution contained inside the area mapped during the learning. The reconstruction of a simulated brain phantom in a noiseless case shows an improvement if compared with Filtered Back-Projection reconstruction (FBP). In noisy cases there is still an improvement, even if we do not compensate for noise fluctuations. These results show that it is possible to reconstruct PET images using ANNs. Initially we used a Dec Alpha; then, due to the high data parallelism of this reconstruction problem, we ported the learning on a Quadrics (SIMD) machine, suited for the realization of a small medical dedicated system. These results encourage us to continue in further studies that will make possible reconstruction of images of bigger dimension than those used in the present work ( $32 \times 32$  pixels).

*Keywords:* PET; Artificial Neural Networks; Image Reconstruction; Massively Parallel Computers.

### 1. Introduction

Positron Emission Tomography (PET) is a tomographic method that allows imaging of parts of any body through the quantitation of the radiopharmaceutical distribution within the body itself. PET imaging essentially consists of two steps: first of all it is necessary to obtain the number of photons detected by each pair of counters and then by means of these counts we have to reconstruct the related image.

This study gives a solution to the PET image reconstruction problem. Forward photon transport for PET may be represented as:

$$P_j = \sum_{i=1}^{N_s} T_{ij} S_i, \quad (1)$$

where  $P_j$  are the counts of the pair of detectors labeled ‘j’,  $T_{ij}$  is the contribution of the source voxel (volume element) ‘i’ to the counts of pair ‘j’,  $S_i$  is the activity at source voxel ‘i’ and  $N_s$  is the total number of the source elements.  $T_{ij}$  relates projections of every pair of counters with all source elements.  $T_{ij}$  can be found, as we will see, by both analytic and Monte Carlo techniques.<sup>1</sup> Using (1) we can represent the reconstruction problem as:

$$S_i = \sum_{j=1}^{N_p} (T_{ij})^{-1} P_j, \quad (2)$$

where  $N_p$  is the total number of pairs and other symbols have the same meaning they had in (1).

Techniques based on reconstruction from projections are used to recover the tracer distribution. The aim of every reconstruction algorithm is to obtain coefficients of the inverse of  $T_{ij}$ . The most common image reconstruction technique is filtered back-projection (FBP).<sup>2,3</sup> Briefly, the FBP method performs the following sequence of operations: the projections are Fourier transformed, multiplied by a suitable filter in the frequency domain, anti-transformed and back-projected. Other reconstruction techniques (e.g., Maximum Likelihood<sup>4</sup>) find the radioactive distribution giving an iterative estimate of the solution of a set of proper linear equations by means of mathematical methods.<sup>5</sup>

We propose an approach that uses an Artificial Neural Network (ANN) to learn iteratively an approximate inverse of  $T_{ij}$ . The ANN is used in two separate phases: learning, which occurs only once and reconstruction, which occurs each time we want to reconstruct an image. During the learning we know both the activity of the source voxels and the corresponding projection counts. In this way the ANN can reproduce the connections between pairs of counters and source elements. Once these connections are found (i.e., the inverse is determined), the reconstruction of an arbitrary source distribution is a single step process. It has been shown that an ANN can reconstruct Single Photon Emission Computed Tomography (SPECT) images.<sup>6,7</sup> We investigate the possibility of reconstructing PET images by ANNs as well.

First we focus our attention on the reconstruction of ideal images (i.e., without introducing any physical effects like attenuation or scattering). Finally we concentrate on simulating through Monte Carlo techniques a PET process including all physical aspects and we try to reconstruct images so generated.

## 2. Methods

### 2.1. Image Reconstruction Problem

The ANN used is a form of Multiple Adaptive Linear Element (MADALINE) network.<sup>8</sup> It is a feed-forward network which consists of one input and one output layer (i.e., it does not include any hidden layer). The basic structure of the ANN is showed in Fig. 1.

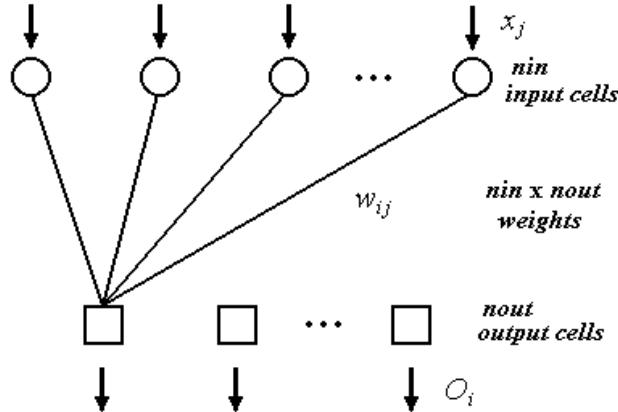


Fig. 1. The basic structure of the PET reconstruction by ANN.

Each output is connected to all the input cells. The outputs of the ANN are calculated by:

$$O_i = \sum_{j=1}^{nin} x_j w_{ij}, \quad (3)$$

where  $O_i$  is the value of each output neuron ( $i = 1, \dots, nout$ ),  $x_j$  is the value of the neuron 'j' of the input layer,  $w_{ij}$  is the value of the weight connecting input neuron 'j' and output neuron 'i' and  $nin$  and  $nout$  are the number of neurons of input and output layer respectively.

The ANN has a linear output because the reconstruction of a PET image, as we have seen in (2), is a linear process. By Eqs. (2) and (3) we can see that there is a strong connection between the ANN topology and the reconstruction process. Indeed if we associate the output layer with the images to be reconstructed and the input layer with the projection counts, we can say that each output neuron represents one pixel (or voxel) of the reconstructed image and each input cell represents one pair of counters. In this way  $nout$  is equal to the number of the pixels (in our case for  $32 \times 32$  squared images there are 1024 pixels) and  $nin$  is equal to the number of the pairs of detectors (for  $Nriv$  counters total possible pairs are  $Nriv(Nriv - 1)/2$ ); we really see that  $nin$  is usually less than  $Nriv(Nriv - 1)/2$ , because some pairs of counters do not give any contribution to the reconstruction

of the image and we can therefore neglect them. To reconstruct  $32 \times 32$  images we use  $N_{riv} = 156$  detectors and  $nin = 5331$  pairs. The value of the weights  $w_{ij}$  are the coefficients of the inverse matrix (i.e., the contribution of the pair ‘j’ to the reconstruction of pixel ‘i’).

There are two separate phases to this approach: training (or learning) and reconstruction. During the training, examples, where both images and corresponding projection counts are known, are presented to the network (projections are presented to the inputs and images to the outputs). In this way the ANN iteratively learns to approximate the inverse matrix. The training examples (patterns) consist of a set of ideal images like in Fig. 2 and their corresponding projections.

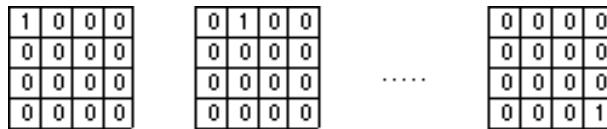


Fig. 2. Ideal images of the training set for  $4 \times 4$  images (total of 16 images). This paradigm is valid also for  $N \times N$  images (with a total of  $N^2$  training images).

With those images, we show to the ANN the behavior it must have to reconstruct each single pixel of the image. To do this we can choose this kind of image because PET is a linear process and so we can use the “superposition principle”: we can so represent every source distribution as a set of elementary point sources such as in Fig. 2. With this set of learning patterns the ANN is able to generalize and once trained it can reconstruct every type of images we want.

As a training paradigm we use the generalized delta rule for supervised learning. Application of the delta rule to our problem gives the formulae:

$$\Delta w_{ij}(n+1) = \sum_k \epsilon \delta_{ik} P_{jk} + \alpha \Delta w_{ij}(n), \quad (4)$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n+1), \quad (5)$$

where  $w_{ij}(n)$  is the value of the weight before adjustment (at iteration ‘n’), while  $w_{ij}(n+1)$  is its value after adjustment (at iteration ‘n+1’). The values of adjustments for each weight are calculated by (4). The first term in (4) is the sum over the number of patterns of  $\delta_{ik}$  (error in source element  $S_i$  in pattern ‘k’) multiplied by corresponding projection of pair ‘j’ and by  $\epsilon$  (learning rate).  $\epsilon$  allows control of the average weight change and it is adjusted to obtain the fastest stable convergence. The value for  $\delta_{ik}$  is found by determining the difference between ideal and actual outputs. The second term in (4) is the momentum term,<sup>8</sup> which reduces oscillations during convergence ( $\alpha$  is fixed to 0.95).

We use a gradient steepest descent mechanism (codified in (4) and (5)) in order to minimize the Mean Squared Error (MSE) between ANN-reconstructed and ideal

images. MSE can be written as:

$$\text{MSE} = \frac{\sum_{k=1}^{npatt} \sum_{i=1}^{nout} \sum_{j=1}^{nin} (x_j w_{ij} - I_{ik})^2}{nout \ npatt}, \quad (6)$$

where  $npatt$  is the number of learning patterns ( $N^2$  for  $N \times N$  images) and  $I_{ik}$  is the value of the element ‘i’ of the ideal image of the pattern ‘k’. Training steps are illustrated in Fig. 3.

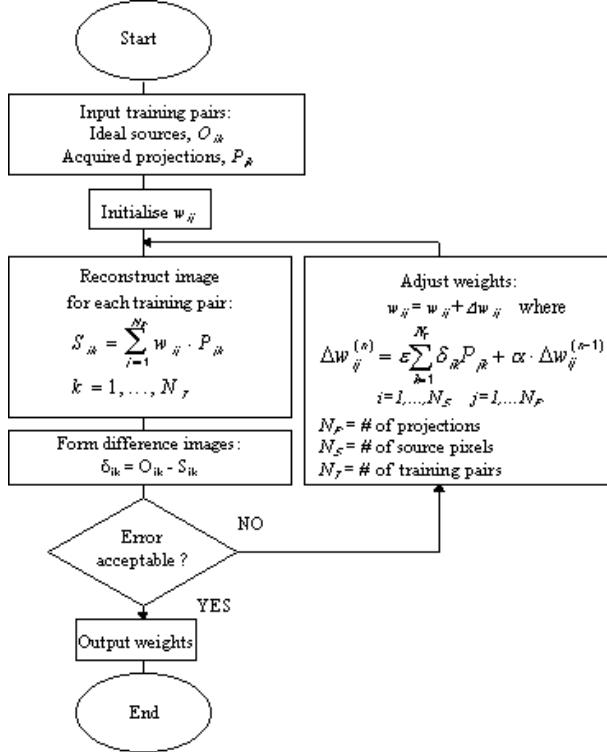


Fig. 3. The training phase of an iteration step PET reconstruction by ANN.

First of all we present to the ANN the training pairs which consist of a set of  $npatt$  ideal images and their corresponding projections. Then we initialize the network weights and we calculate the reconstructed ANN images through (3) and the pixel-by-pixel difference error  $\delta_{ik}$  for all the images of the set. This done, we find the MSE between reconstructed and ideal images. If this value is acceptable we stop the learning, otherwise we will update the weights through (4) and (5) and then we shall perform other iteration steps until the MSE has converged to an acceptable value. In the first step we initialized the weights with random values included from 0 to 1.

MSE is a real function that depends both on ANN weights and how training patterns are made. With the delta rule algorithm, in each learning iteration, we move in MSE surface of a step equal to  $\epsilon$  following the gradient of MSE itself. The aim of the training phase is to find the minimum of MSE. The value of this minimum depends on the number of detectors used. If the minimum value reached during training is not sufficient to perform an acceptable reconstruction, it is necessary to increase the number of detectors. Actually, for each kind of the whole system geometry (area of mapped images, dimension of detectors ring), we must have at least a certain number of detectors, if we want a good reconstruction. This fact is correlated with the discretization due to finite sampling by the acquisition system.

Once training is ended, weights are kept constant and they are accepted as elements of the inverse matrix. Now we are ready to reconstruct every type of image we want. Reconstruction, as we can see in Fig. 4, is a single step process.

The only thing we have to do is to present the projections of the image to be reconstructed to the ANN inputs and perform a forward step like in (3). Images are formed by calculating the dot product between inverse matrix and projection data. Obviously geometries to be reconstructed must be contained inside the area mapped during the learning (area of the images of training set).

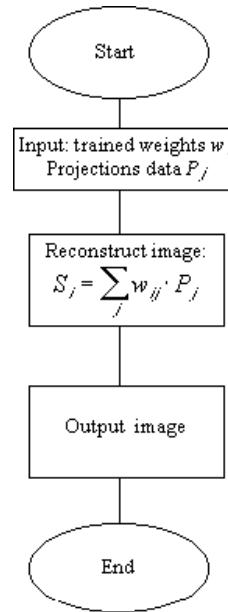


Fig. 4. The reconstruction phase of the PET reconstruction by ANN.

In this section we have illustrated how an ANN can learn to reconstruct a PET image from projection counts. In next section we will see how to simulate the projections of a PET process.

## 2.2. Types of Simulation Techniques Used

Projection counts are the only data we really know when we have to reconstruct a PET image. Indeed the tomograph provides us with the counts of the photons detected by each pair of counters; from these counts we have to reconstruct the density distribution of the radioactive source. Therefore it is important to set up a simulation system that gives us projections of every kind of image we want to reconstruct. We need projection counts both during training phase (each training pair consists of  $\langle$ image ' $k$ ', projections of image ' $k'$  $\rangle$ ) and when we want to reconstruct a known image to test the goodness of the ANN reconstruction.

We use two types of simulations: in the first one we calculate projections through geometric methods neglecting all physical effects, while in the second one we use Monte Carlo techniques which includes all aspects such as attenuation, scattering and so on. This way we are able to see the quality of the reconstruction in the absence or presence of physical effects.

In a geometric simulation we calculate projections of each pair by means of formula (1).

Pixels that intersect the line joining two detectors give a contribution to counts of that pair proportional to the length of the intersecting line included in the pixels themselves ( $l_{34}$  in Fig. 5).  $l_{ij}$  are elements of the matrix  $T$  ( $l_{ij} = T_{ij}$ ). Other pixels

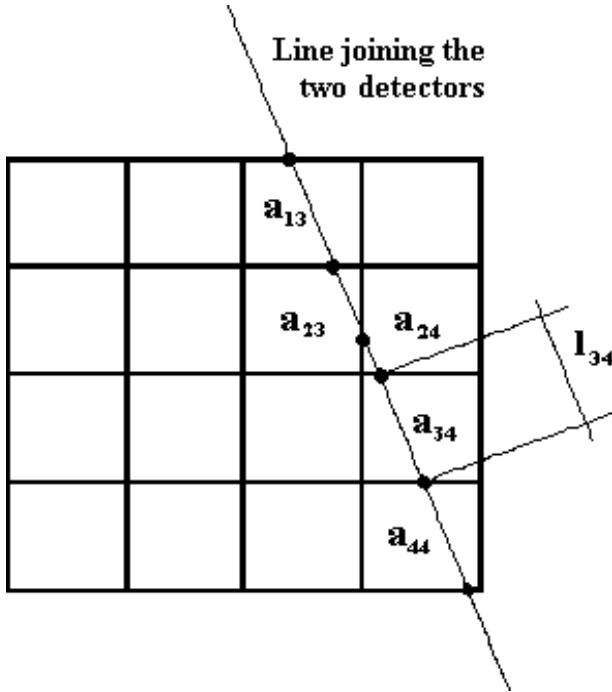


Fig. 5. Acquisition process: Projections for a pair of counters are calculated through line joining the centers of the two detectors.

(not intersecting the joining line) do not give any contribution to that pair. This way we can find elements  $T_{ij}$  by the analytic (or geometric) technique. Projections for each pair are calculated by summing contributions of each intersecting pixel multiplied by the activity inside the pixel itself.

For the second type of simulation we adopt a Monte Carlo simulator based on the EGS4 code (Electron-Gamma-Shower).<sup>1</sup> This code allows the transport of particles in every material and in every kind of geometry by means of Monte Carlo techniques. We set up a system such as the one shown in Fig. 6.

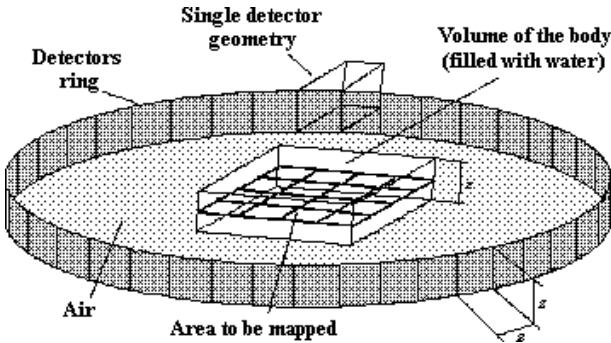


Fig. 6. PET system simulated.

The area to be mapped (i.e., the image to be reconstructed) is contained in a body of  $3.2 \times 3.2 \text{ cm}^2$  and height  $Z = 1.3 \text{ mm}$ . The volume of the body is filled with water to simulate in a reasonable way the human body tissues. Detectors lie in a ring of radius equal to  $3.2 \text{ cm}$ ; each one of the 156 detectors consists of a  $0.15 \times 0.15 \text{ cm}^2$  squared base parallelepiped made of Yttrium Aluminium Perovskite (YAP).<sup>9</sup> The remaining volume of the space between the body and the detectors ring is filled with air.

Starting from emissions of positrons of 700 keV, we simulate the whole PET process (thermalization, annihilation, photons transport in mediums and finally detection of photons). The number of positrons emitted from each pixel is equal to the desired activity inside that pixel. In order to reduce statistical fluctuations we use training patterns such as in Fig. 2, but we simulate 100 000 emissions from each pixel. Then we properly scale values of projection counts. These scaled counts, corresponding to each pattern of the training set, represent the elements of the matrix  $T_{ij}$ . Therefore we can see that these coefficients can be found also by Monte Carlo methods.

### 3. Results and Discussion

The first type of simulation technique used is, as we have seen, the geometric simulation. We perform it by an analytic projector primarily to investigate the possibility for an ANN to reconstruct a PET image and secondly to choose the value of

parameters to be used. We thus empirically find that to reconstruct a  $32 \times 32$  squared image with the side equal to the radius of the detectors ring, we need a number of sampled angles (equal to the number of detectors) greater than 110. If we use a number of detectors much greater than this value, reconstruction is still good, but the resources required for training increase. We also fixed the momentum term to 0.95: this value gives a more rapid convergence during the training. The choice of the optimal learning rate is strongly connected with the projection values of the training set.

Once the minimum number of detectors, required for a good reconstruction, and the momentum term are fixed, we perform the simulation by the Monte Carlo projector. The geometry of the simulated system is the same as in the previous case.

### 3.1. Training

Training is the most critical phase of the ANN algorithm, indeed it requires a lot of resources (both CPU and memory). With a serial architecture it is not possible to perform training for images bigger than  $32 \times 32$  in an acceptable time, so we investigate the improvement obtained with learning on a parallel architecture machine. We accomplish the training with two different machines: a DEC Alpha machine with 200 MHz, 380 MFlops single processor and a Quadrics SIMD machine<sup>10</sup> with 128 nodes, an  $8 \times 4 \times 4$  topology and a peak performance of 6.4 GFlops.

Learning for  $32 \times 32$  images (using an ANN with 5331 input neurons and 1024 output neurons) needs 55 MB memory resources and each iteration step requires about 30 minutes on the Alpha machine. The minimum during training is reached after a few hundred of iteration steps, as we can see in Fig. 7 for the training of  $16 \times 16$  and  $32 \times 32$  images.

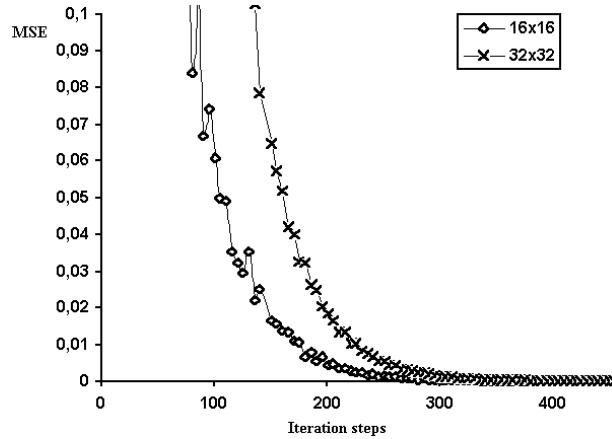


Fig. 7. Convergence of MSE during the training of the  $16 \times 16$  and  $32 \times 32$  images.

Neural networks, because of their structure, are fit to be realized in a natural way on non-sequential architectures, like the MIMD (Multiple Instruction Multiple Data) or SIMD (Single Instruction Multiple Data) ones. Here, the problem has a high data parallelism, given by both pattern presentation and output nodes computation, therefore the latter (SIMD) is the optimal architecture for us. We used the massively parallel Supercomputer Quadrics (QH1) consisting of 128 floating point units, with a total of 512 MB RAM. We cannot implement the whole network on each physical node (PN), because each PN has poor memory resources available (4 MB). Therefore we must keep sequential pattern presentation and hence we parallelize on the output nodes (ON). Obviously, when the number of pixels (equal to ON) is greater than number of PN, we have to implement more than one ON on each PN. This way the degree of parallelism decreases: for  $16 \times 16$  images, ON/PN is equal to 2, while for  $32 \times 32$  images ON/PN is equal to 8. Obviously we perform learning tests with the same data used on the Alpha machine (380 MFlops) with  $16 \times 16$  and  $32 \times 32$  images. Due to the limited size of the RAM of each PN, we have to fetch the training patterns from a disk device at each iteration step. We consider two such CPU-time measures as meaningful ones: the first one represents the total time (TT), and it also includes the read time (RT) necessary to fetch the patterns; the second one points out only compute time (CT = TT - RT). In training with an Alpha machine we can neglect the read time, because the patterns are fetched only once at the beginning of learning and they are then kept in memory.

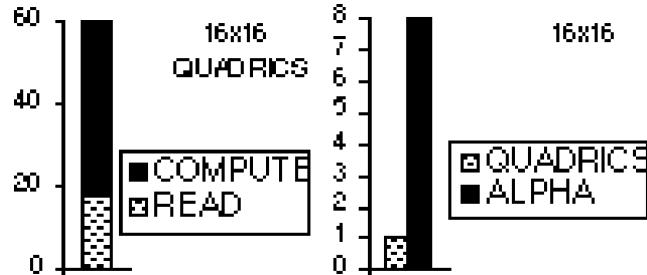


Fig. 8. For a  $16 \times 16$  sized image: Comparison between CT and RT on Quadrics (on the left) and between Quadrics and Alpha TT (on the right); units are minutes (left) and hours (right).

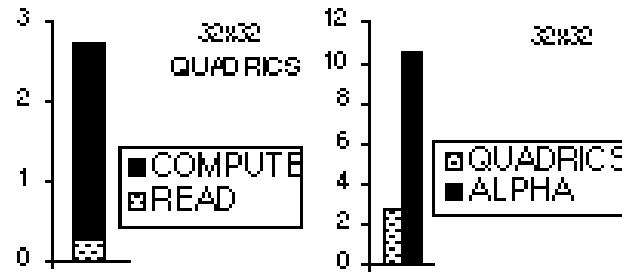


Fig. 9. For a  $32 \times 32$  sized image: Comparison between CT and RT on Quadrics (on the left) and between Quadrics and Alpha TT (on the right). Values are in day units.

In Figs. 8 and 9 we compare CPU time used in the two different architectures: as we can see,  $TT(Alpha)/CT(QH1) \sim 13$  in the first case and it is  $\sim 4$  in the second one; a remarkable improvement, mostly if we consider that in peak performance terms Alpha is equal to about 8 Quadrics nodes. This improvement is more evident in the first image ( $16 \times 16$ ), indeed the  $16 \times 16$  image is the most meaningful one, because in this case there is a higher degree of parallelism ( $ON/PN = 2$ ). If we consider the total time  $TT(QH1)$ , the improvement in the  $16 \times 16$  image is much less remarkable, because the read time is roughly equal to  $1/3$  of the total time: this is due to very slow communication between the QH1 and the storage device (approximately 1 MB/s). In the  $32 \times 32$  image, where read time is less significant and the parallelism has a lower degree ( $ON/PN = 8$ ) the decrease of improvement between parallel and serial architectures is smaller. Instead, analysing the CT in the  $16 \times 16$  image, that is the most meaningful one for Quadrics machine with 128 nodes, we can say that this training algorithm is particularly suitable for massively parallel architectures, because the very high data parallelism of the ANN reconstruction problem allows us to obtain a remarkable improvement. Moreover, the Quadrics machine could be used in a small, powerful dedicated system, suitable for medical applications. When we have a Quadrics machine with a faster communication bus and a greater number of nodes (the next generation of Quadrics machines,<sup>11</sup> expected for the beginning of 1999, will consist of 2048 nodes, each one with a peak performance sixteen times greater (800 MFlops) than now (50 MFlops)) we will perform training of images bigger than  $32 \times 32$  in an acceptable time (it should be even possible for a  $128 \times 128$  images training in a quite reasonable time). These results offer us an algorithm-architecture pair that shows that this seems to be the right way. Thanks to a reduction of the learning time obtained by using a Quadrics machine, we can execute useful studies on the optimal choice of training parameters (learning rate, momentum term and so on).

A more rapid convergence (independently from the type of architecture used) should be obtained using another type of training paradigm (e.g., conjugate gradient method<sup>12</sup>) and a  $\pi/2$  geometric symmetry should reduce computational needs.

### 3.2. Reconstruction

To test the reconstruction method, we use an artificial image obtained from the projections of a simple brain phantom. This phantom represents a single brain slice with 55 gray levels. We used the methods described in Sec. 2.2 to obtain three images: an *ideal image*, from a simple geometrical projection; a *noiseless image*, from a Monte Carlo simulation with  $20 \cdot 10^6$  positron emissions; a *noisy image*, from a Monte Carlo simulation with  $1 \cdot 10^6$  positron emissions. Once the projection counts related to this phantom are obtained, we reconstruct the image by the ANN and FBP methods. To perform the FBP reconstruction, we use software available in Ref. 13 and we include the attenuation correction in reconstruction of Monte Carlo simulated images by means of multiplicative techniques of Chang.<sup>14</sup> The training

phase of the ANN reconstruction requires 400 iteration steps; the reconstruction phase, as we have seen, is a single step process. It needs, for a  $32 \times 32$  image, 24 MB memory resources and, once the reconstruction matrix is fetched, we can perform the reconstruction on an Alpha machine in a few seconds (the ANN reconstruction time is comparable with the FBP one).

We measure the accuracy of the two reconstructions (FBP and ANN) by the Normalized MSE (NMSE) formula defined as:

$$\text{NMSE} = \frac{\sum_{i=1}^{N_s} (O_i - I_i)^2}{N_s \sum_{i=1}^{N_s} I_i^2}, \quad (7)$$

where  $O_i$ ,  $I_i$  are the values of the  $N_s$  reconstructed and original pixels. The improvement between the two reconstructions can be written as:

$$\text{IMP} = \frac{\text{NMSE}_{FBP} - \text{NMSE}_{ANN}}{\text{NMSE}_{FBP}}. \quad (8)$$

The ANN reconstruction for the ideal image (Fig. 10 and Table 1) is quite perfect because, due to the absence of physical effects, there are no fluctuations between examples proposed to the ANN during training and images presented in the reconstruction phase. We will see that the ANN reconstruction is worse when we introduce physical effects. In Fig. 11 we show the reconstruction of the noiseless image: the introduction of physical effects leads to a worse reconstruction, both for the FBP and the ANN (see Table 1). The ANN method is much better, although it is more sensitive to physical effects.

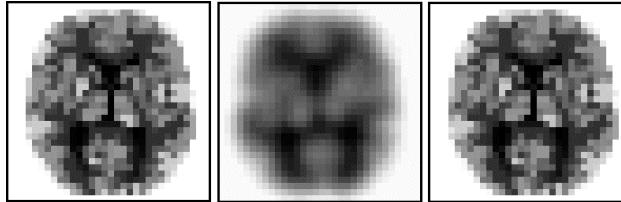


Fig. 10. Reconstruction of the ideal image. Original image (left), FBP reconstruction (middle) and ANN reconstruction (right).

Table 1. NMSE for ANN and FBP reconstruction of the ideal image, the noiseless and the noisy one. Last row is the relative improvement IMP.

	Ideal Image	Noiseless Image	Noisy Image
FBP	$292 \cdot 10^{-6}$	$970 \cdot 10^{-6}$	$970 \cdot 10^{-6}$
ANN	$3 \cdot 10^{-6}$	$83 \cdot 10^{-6}$	$333 \cdot 10^{-6}$
IMP (%)	98	91	66

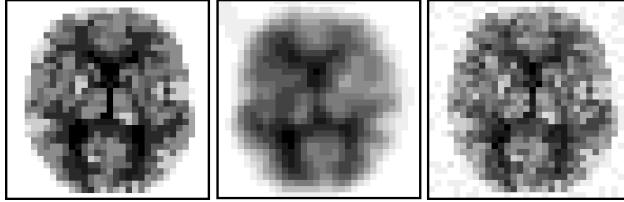


Fig. 11. Reconstruction of brain phantom with  $20 \cdot 10^6$  counts. Original image (left), FBP reconstruction (middle) and ANN reconstruction (right).

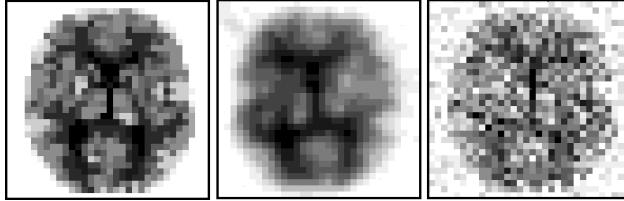


Fig. 12. Reconstruction of the noisy image. Original image (left), FBP reconstruction (middle) and ANN reconstruction (right).

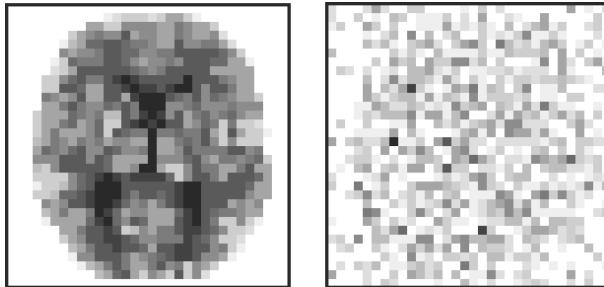


Fig. 13. Noiseless case. Difference between the original image and the reconstructed one: FBP (left) and ANN (right).

In Fig. 12 we show the reconstruction of the noisy image. We can see that now the ANN reconstruction is much worse than in the noiseless case seen above: the improvement now is equal to 66% , but there is still a substantial improvement over the FBP (see Table 1). The density gradients are better reproduced with the ANN than with the FBP: we see that the FBP is not sensitive to the increase of the statistical noise. For images bigger than those used in this work this effect would assume more importance. It is therefore necessary to study some methods that reduce the noise fluctuations. In particular Wiener filtering techniques<sup>6,15,16</sup> should yield an important improvement in reconstruction in cases with noisy data and result in a lower minimum of the NMSE.

In Fig. 13 we can see the pixel-by-pixel difference between the original image and the reconstructed one for the noiseless case.

We can see that the ANN difference image does not show any brain structure. We can therefore say that the ANN is able to reconstruct any kind of distribution contained inside the area mapped during training. Moreover, while training requires a lot of resources and it must be therefore executed on a parallel machine, the reconstruction phase can be obtained easily on a single processor machine in a reasonable time.

#### 4. Conclusion

This study has shown that it is possible to reconstruct the radioactive distribution and hence an image of a PET system using artificial neural networks. To do this we must train the network with simulated data that most closely match the parameters of the real acquisition system. The training phase requires a lot of computational resources and it must be executed on a parallel machine. However it is sufficient to perform the training only once; the reconstruction phase is a single step process and it is possible to perform it on a single processor machine.

The results obtained show that the ANN reconstruction is much better than the FBP one for noiseless data (i.e., with more counts), while for the noisy case it is still better, although the improvement is quite low. Further studies must be done in order to reduce the effects of noise fluctuations and to make the training less heavy, if we want to reconstruct real PET images. Wiener filtering techniques should be especially useful in noisy data cases; moreover conjugate gradient methods and the reduction of the training set by means of geometric symmetry should reduce the computational needs in the training phase.

#### References

1. W. R. Nelson, H. Hirayama, and D. W. O. Rogers, "The EGS4 code system," SLAC-Report-265, 1985.
2. D. W. Townsend and M. Defrise, "Image reconstruction methods in Positron Tomography," Lectures given in the academic training program of CERN, 1992/93, CERN Report 93-02.
3. R. A. Brooks and Di Chiro, *Phys. Med. Biol.* **21**, 689 (1976).
4. L. A. Shepp and V. Vardi, *IEEE Trans. Med. Imag.* **1**, 113 (1982).
5. G. T. Herman, *Image reconstruction from projections: The fundamentals of Computerized Tomography* (Academic Press, New York, 1980).
6. M. T. Munley, C. E. Floyd Jr., J. E. Bowsher, and R. E. Coleman, *Med. Phys.* **21**, 1889 (1994).
7. C. E. Floyd Jr., *IEEE Trans. Med. Imag.* **10**, 485 (1991).
8. B. J. Kroöse and P. P. Van der Smagt, *An Introduction to Neural Networks* (University of Amsterdam, 1993).
9. R. Pani et al., *Nucl. Instr. and Meth.* **A348**, 551 (1994).
10. C. Battista et al., *Int. J. High Speed Comp.* **5**, 637 (1993).
11. A. Bartoloni et al., *APEmille: A Parallel Processor in the Teraflops Range*, Internal Note, Department of Physics, University of Rome La Sapienza (University of Rome, 1994).
12. M. R. Hestenes and E. Stiefel, *Nat. Bur. Standards J. Res.* **49**, 409 (1952).

13. R. H. Huesman, G. T. Gullberg, W. L. Greenberg, and T. F. Budinger, "RECLBL Library users manual," Lawrence Berkeley Laboratory, University of California, 1977.
14. L. T. Chang, *IEEE Trans. Nucl. Sci.* **25**, 638 (1978).
15. M. A. King, P. W. Doherty, and R. B. Shwinger, *Med. Phys.* **10**, 876 (1983).
16. B. C. Penney, S. J. Glick, and M. A. King, *IEEE Trans. Med. Imag.* **9**, 60 (1990).