

POLITECNICO DI MILANO
Faculty of Industrial Engineering and Informatics
Department of Electronics, Informatics and Bioengineering



Application of Neural Networks to Anger camera imaging

Supervisor: Prof. Carlo Fiorini

Tesi di laurea specialistica di:
Ashfaq Ahmad Janjua
766509
Anno Accademico 2012-2013

Table of Contents

Introduction	7
Chapter 1 Introduction of Neural Networks	9
1.1 The Biological Neuron	10
1.2 The Artificial Neuron	11
1.3 Activation Functions	12
1.3.1 Step function:.....	13
1.3.2 Sigmoid Function:	14
1.3.3 Hyperbolic Tangent.....	14
1.3.4 Linear Function:	15
1.4 Artificial Neural Networks.....	16
1.5 Topology and Complexity	17
1.6 The Perceptron	18
Step function.....	18
1.6.1 Example: Logic Operator.....	19
1.6.2 Decision boundary for the Perceptron	21
1.7 Hebb Learning Rule	22
1.7.1 Hebbian Learning of the AND Operator.....	22
Chapter 2 APPLICATION OF NEURAL NETWORKS IN BIOMEDICAL	26
2.1 Pattern Recognition	27
2.2 Feedforward Neural Network.....	29
2.2.1 Regression:.....	31
2.2.2 Classification:	34
2.3 Supervised Learning in Neural Network	36
2.4 Backpropagation	38
2.4.1 Formulation of the BP Algorithm.....	39
2.5 Optimization	40
2.5.1 Steepest Descent Algorithm	43
2.5.2 Newton's Method	45
2.5.3 Levenberg–Marquardt Algorithm.....	49
Chapter 3 Hicam - High resolution gamma camera	50
3.1 Nuclear Scintigraphy	52
3.2 Hicam - System overview.....	55

3.2.1 Scintillator	56
3.2.2 Collimator.....	62
3.2.3 Silicon Drift Detector (SDD).....	68
3.2.4 Readout electronics	73
3.2.5 Data Acquisition system.....	76
3.2.6 Processing Unit	76
Chapter 4 Chapter 4 Reconstruction Program and Algorithm implemented	78
4.1 Data and zone subdivision	79
4.2 Main NNs GUI.....	80
Chapter 5 Experimental Results.....	88
5.1 First results obtained from Training – Matrices used for the training.....	89
5.2 XY reconstruction from Pattern Net by simulated Data	93
5.3 XY reconstruction from feedforward Net by simulated Data	96
5.4 Z Reconstruction by simulated Data	98
5.5 Energy Reconstruction by simulated data	100
5.6 XY Reconstruction from feedforward Net by Experimental Data.....	101
5.7 Z Reconstruction from feedforward Net by Experimental Data.....	102
5.8 Energy Reconstruction from feedforward Net by Experimental Data.....	103
5.9 XY Reconstruction from Pattern Net by Experimental Data	103
5.10 Z Reconstruction from Pattern Net by Experimental Data	104
Conclusion	105
Bibliography.....	106

Table of Figures

Figure 1.1 Neuron	10
Figure 1.2 Artificial Neuron	11
Figure 1.3 (a) Action potential displayed at oscilloscope (b) The parts of an action potential	12
Figure 1.4 Step Function	13
Figure 1.5 Sigmoid Function	14
Figure 1.6 Hyper Tangent	15
Figure 1.7 Linear Function	15
Figure 1.8 The basic architecture of ANN.	16
Figure 1.9 Different topologies for Neural Networks	17
Figure 1.10 Perceptron	18
Figure 1.11 Linear Discriminator	19
Figure 1.12 Perceptron as a logic AND	20
Figure 1.13 Perceptron as a logic OR	20
Figure 1.14 Decision boundary	21
Figure 1.15 Decision boundary for (a) AND (b) OR	21
Figure 1.16 Decision boundary for AND operator for first Epoch	23
Figure 2.1 Grid of sensors in gamma camera to represent the classes (green circles)	27
Figure 2.2 Input matrix (left) and the desired response matrix (right) for the pattern recognition neural network	28
Figure 2.3 Linear Regression	32
Figure 2.4 Feedforward Net Architecture	33
Figure 2.5 Supervise Learning	36
Figure 2.6 Backpropagation	38
Figure 2.7 Error surface 3D	41
Figure 2.8 Error as a function of weights	42
Figure 2.9 The Method of Steepest Descent Approaches the Local Minimum in a zig-zag path, and the next search direction would be orthogonal to the next	43
Figure 3.1 Gamma camera	51
Figure 3.2 Gamma Camera	53
Figure 3.3 Focusing lens	63
Figure 3.4 Collimators	64
Figure 3.5 Collimator design	66
Figure 3.6 Circular SDD	69
Figure 3.7 SDD basic operation	69
Figure 3.8 Potential distribution in the SDD volume - anode at (0,0)	71
Figure 3.9 Hicam SDD array	72
Figure 3.10 Schematic of a processing channel in Hicam ASIC	74
Figure 4.1 All the zones are shown overlapping one each other	79
Figure 4.2 The first zone is highlighted	80
Figure 4.3 Main GUI	81
Figure 4.4 Network Topology GUI	82
Figure 4.5 Main GUI . Network Selection	83

Figure 4.6 Main GUI . Data partitioning and input matrix dimensions 84
Figure 4.7 Main GUI – Button to select extra variables as input as NN 85
Figure 4.8 Main GUI – Provide to select different sectors 85
Figure 4.9 Main GUI – Display the training results 86
Figure 4.10 Main GUI – Buttons to load input matrix, trained NN and test NN etc. 86
Figure 4.11 GUI to test and visualize experimental data 87
Figure 5.1 Results obtained from Training – X Error 90
Figure 5.2 Results obtained from Training – X Error slope 91
Figure 5.3 Results obtained from Training – E Error 92
Figure 5.4 Results obtained from Training – E Error slope 93
Figure 5.5 Simulated Data XY Reconstruction - Pattern Net 94
Figure 5.6 Simulated Data XY Reconstruction - Pattern Net 95
Figure 5.7 Simulated Data XY Reconstruction - Pattern Net 95
Figure 5.8 Simulated Data XY Reconstruction - Pattern Net 96
Figure 5.9 Simulated Data XY Reconstruction - Feedforward Net 97
Figure 5.10 Simulated Data XY Reconstruction - Feedforward Net 97
Figure 5.11 Simulated Data Z Reconstruction – Pattern Net 98
Figure 5.12 Simulated Data Z Reconstruction – Feedforward NN 99
Figure 5.13 Simulated Data Z Reconstruction with Noise – Feedforward NN 99
Figure 5.14 Simulated Data Energy Reconstruction 100
Figure 5.15 Experimental Data XY Reconstruction with nine Sectors - Feedforward net 101
Figure 5.16 Experimental Data XY Reconstruction with one Sector - Feedforward net 102
Figure 5.17 Experimental Data Z Reconstruction - Feedforward net 102
Figure 5.18 Experimental Data Energy Reconstruction - Feedforward net 103
Figure 5.19 Experimental Data XY Reconstruction - Pattern net 104
Figure 5.20 Experimental Data Z Reconstruction - Pattern net 104

Abstract

Hicam is a research project that is under continuous development by leading European institutes which are Department of Electronic Engineering of Polytechnic institute of Milan, Max-Planck Institute, University of Milan, and University College London.

The objective of this project is to develop high resolution and compact Gamma camera for clinical and research purposes, with overall spatial resolution of less than 1 mm. The most critical part of the gamma camera is the reconstruction of the interaction of events inside crystal based on the signals comes from the array of SSD detectors.

There are different algorithms that could be used for the estimation of the coordinates of interaction of gamma rays inside the scintillator crystal, such as centroid method and Maximum likelihood. But each method has its own advantages and disadvantages in context of field of view, spatial resolution and processing time.

This work is based on the investigation of the Artificial Neural Networks architectures especially Pattern Net and Feed Forward Neural Network to provide better estimation of X, Y and Z co-ordinates of gamma rays with good energy resolution at low energies. Neural Networks are very complicated from training point of view but requires little processing time.

Introduction

The purpose of this work is to investigate Artificial Neural Network architectures for the event reconstruction of gamma ray interaction inside scintillation for gamma camera. The standard architecture of ANN is composed of input layer which takes the data from the sensor grid, hidden layer used of processing and the number of output nodes equal to the number of information required(X,Y,Z,E). In this work, we focus on two types of Neural Networks, one is the Feed Forward Neural Network and the other is the Pattern Recognition Neural Network and compares their results.

The conventional architecture of gamma camera is based on Anger logic, which is the most popular method to estimate the position of gamma ray interaction on a sensor grid with continuous scintillator. But it reduces the useful field of view, the spatial resolution and produces artifacts at the edges of the scintillator crystal because of nonlinearities as in the case of centroid method. On the other hand maximum likelihood produce good results but it takes long time for the estimation of position of gamma ray.

In the first part, we develop a Graphical User Interface for the investigation of different Neural Network topologies in order to produce good estimation of X, Y and Z co-ordinates of gamma ray interaction inside scintillator crystal and energy contained by the gamma ray. The GUI takes two types of matrices, one is the data acquisition matrix which is simulated using GEANT for the training of the ANN and the other matrix contains the experimental data to produce the correct position of gamma ray interaction in 3D way. GEANT was developed by the CERN Laboratories and allows to simulate the possible interactions concerning a gamma ray inside a specific material, locating the interaction points (more than one if there was Compton scattering) and the energy released in each one [16].

In the second part, We deal with the training of ANN based on different network topologies and to improve the field of view and spatial resolution of gamma camera with good estimation of

energy resolution, as in the case of gamma rays with high energy gives better spatial resolution but degrades the energy resolution.

Chapter 1 Introduction of Neural Networks

1.1 The Biological Neuron

The basic inspiration behind Artificial Neural Networks was biological nervous system.

Therefore, it is important to have some knowledge about the working of this system.

All organisms have a controlling unit, which is able to learn and adapt it according to the environmental changes. Animals with developed brain such as humans have very complex networks of highly specialized neurons in order to perform tasks

The brain can be divided in different functional and anatomic sub-units, each one is responsible for certain activity like hearing, vision, sensor and motor control. It is connected to the sensors through nerves and the actors are in the rest of the body.

The brain consists of millions of neurons, approximately 10^{11} in average. Neurons could be seen as the basic building block of the central nervous system. They are interconnected at points called synapses. The brain is complex due to the large number of densely connected simple units working together. Each neuron receives an input from millions of other neurons[1].

The structure of a neuron is similar to animal cell. The neuron can be divided in three major parts: Soma(the cell body), Dendrites, and the axon. The structure of a neuron is shown in figure 1.1.

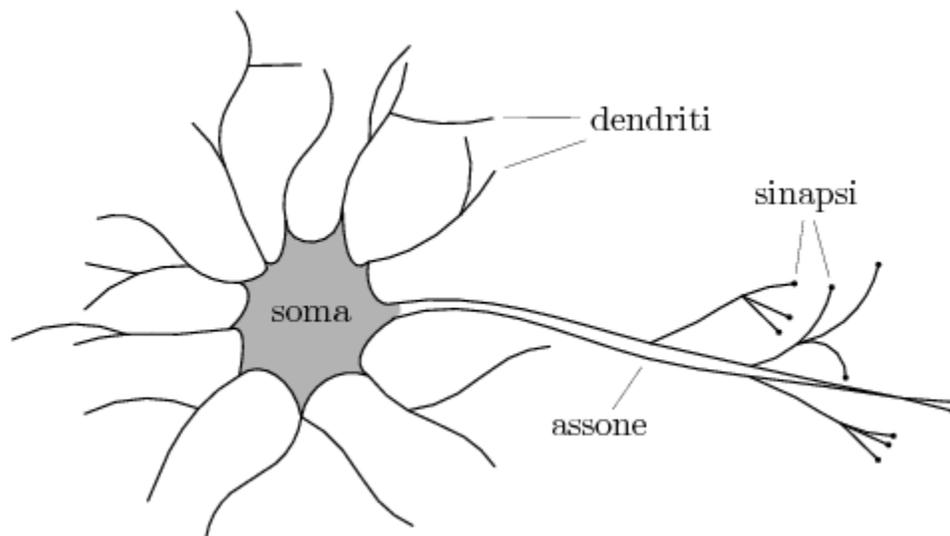


Figure 1.1 Neuron

The signal is passed from a neuron to another one chemically and/or electrically through the synapses and dendrites. To generate a new signal, the neuron internal voltage has to surpass a determined threshold called action potential. The transmission speed of the signal is low (ten of ms). The fast response is due to the large number of connections between neurons. If a link is used frequently, it will generate a bigger signal in less time, on the other hand, if it is not often used, it will shrink till disappear [2].

1.2 The Artificial Neuron

When modeling artificial neuron from real neuron, the complexity of is highly abstracted. It is basically considering inputs like synapses, which are multiplied by weights considered as strength of the respected signal. The final output is computed by a mathematical function called activation function $G(Z_j)$ and the threshold function b_0 .

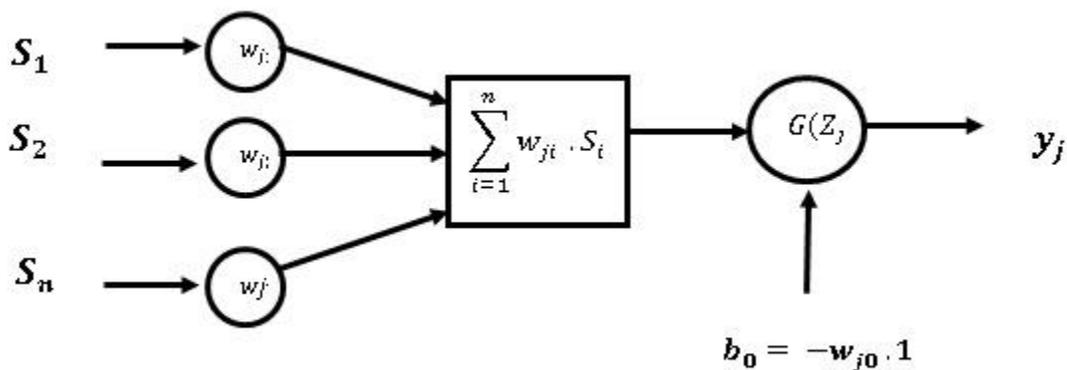


Figure 1.2 Artificial Neuron

The strength of the input depends upon the weight of the artificial neuron which is multiplied by. The stronger input corresponds to the higher weight. The weights may have negative values, so in this case the signal is considered as inhibited by negative weight. The computation of the neuron depends on weight, different weights produce different output. We can produce the desire output by adjusting the weights of an artificial neuron for specific inputs [3].

For this simple model of neuron j it can be identify that:

S_i = Input Signal

w_{ji} = Synaptic weights or simply weights

$Z_j = \sum_{i=1}^n w_{ji} \cdot S_i$ = Activation value

$G(Z_j)$ = Activation function

b_0 = Activation threshold or bias

y_j = Output of J^{th} neuron

As in human behavior to remember a thing you have to think about it again and again, it is the same with the NNs: you have to present different data at the network inputs many times before obtaining the desired output. This operation is called TRAINING.

1.3 Activation Functions

When we study brain, we come to know that the activation function is the abstract representation of the firing rate of the action potential in which the potential of a cell membrane changes rapidly rises and fall, follow a consistent trajectory as shown in Fig 1.3. As the neuron is firing we can represent it with binary 1 on the other hand it is zero[4].

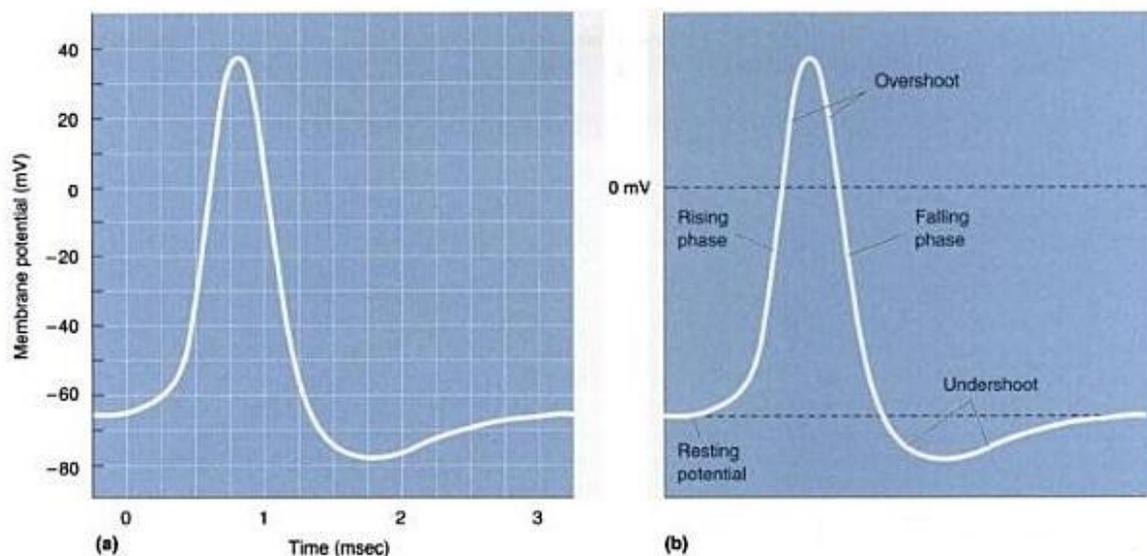


Figure 1.3 (a) Action potential displayed at oscilloscope (b) The parts of an action potential

As in the case of computational networks, the activation function of a specific node calculates the output of that node based on the given set of inputs or a single input. We can consider any standard digital circuit as the network of activation functions that could be 1 as ON or 0 as OFF, depending on input. The linear perceptron in Artificial Neural Networks exhibits the same behavior of digital circuits. However, in order to compute nontrivial problems, a *nonlinear* activation function requires only a small number of nodes[5].

Activation functions are of following types:

1.3.1 Step function:

The original Perceptron uses a step function as an activation function. The output is represented by a certain value, V_1 , if the weighed sum of the inputs is above a certain threshold and A_0 if the weighed sum of the inputs is below a certain threshold. The values A_1 and A_0 could represent 1 or 0.

$$\begin{aligned} G(Z_j) &= 1 && \text{if } Z_j > 0 \\ G(Z_j) &= 0 && \text{if } Z_j = 0 \\ G(Z_j) &= -1 && \text{if } Z_j < 0 \end{aligned}$$

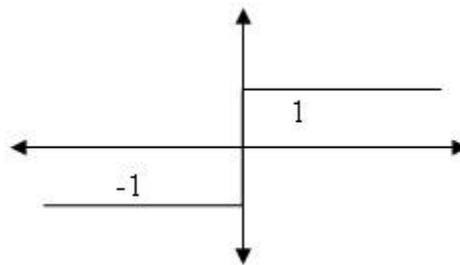


Figure 1.4 Step Function

The step activation function is useful for binary classification problems. When we want to classify an input pattern into one or more classes, a binary classifier with a step activation function is very useful. Another useful application that can use step function could be to create a set of small feature identifiers. Each identifier would be a small network that would output a 1 if a particular input feature is present, and a 0 otherwise.

1.3.2 Sigmoid Function:

The nonlinear curved S-shape function is called the sigmoid function. It is common type of activation used to design the neural networks. Mathematically it is well mannered, strictly increasing and differentiable function.

$$G(Z_j) = \frac{1}{1 + e^{\beta \cdot Z_j}}$$

The sigmoid function is derived from exponential function. Different shapes of the function can be achieved by varying β , the slope parameter. It adjusts the abruptness of the function as it changes between the two asymptotic values.

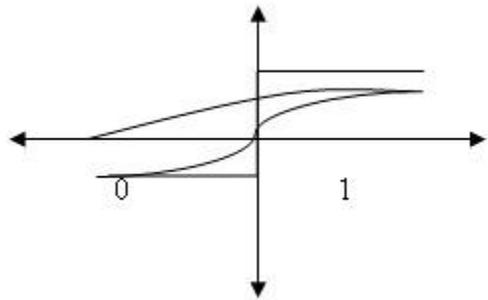


Figure 1.5 Sigmoid Function

1.3.3 Hyperbolic Tangent

This function is easily defined as the ratio between the hyperbolic sine and the cosine functions (or expanded, as the ratio of the half-difference and half-sum of two exponential functions in the points z and $-z$):

$$G(Z_j) = \frac{e^{Z_j} - e^{-Z_j}}{e^{Z_j} + e^{-Z_j}}$$

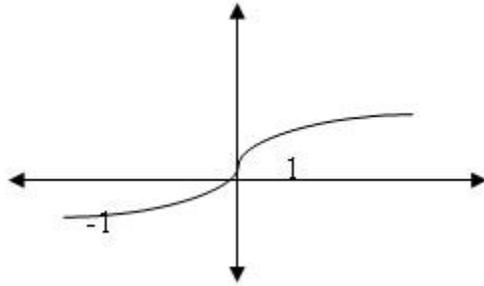


Figure 1.6 Hyper Tangent

1.3.4 Linear Function:

The common activation function for neural networks is the linear activation function. It can produce only positive numbers over the entire real number range. It is most useful for the output layer of a predictive neural network. The linear activation function is only suitable for the output layer if a neural network trained with gradient descent based training methods because the sigmoid activation function has a constant derivative of one.

$$G(Z_j) = C \cdot Z_j$$

where C is the constant of proportionality.

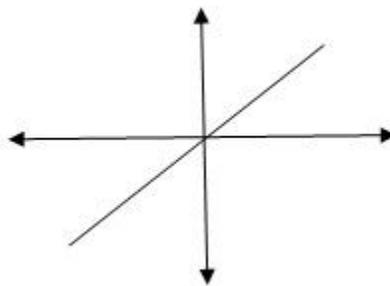


Figure 1.7 Linear Function

1.4 Artificial Neural Networks

Artificial Neural Networks (ANNs), belongs to the field of artificial intelligence which is based on the models proposed by cognitive theory. Cognitive theory is a learning theory of psychology that describes the human behavior by understanding the thought processes. The working methodology in ANNs in a computer system is very similar to that of cognitive theory. Artificial Neural Network composed of several neurons and divided into different layers.

Layer: Neurons at the same distance from the input neurons

Input Layer: The input layer receive the initial “stimulus” .

Output Layer: The output Layer of neurons gives the final result to the network

Hidden Layer: One or more layers of neurons that takes data from previous layer and process it and provide it to the next layer of neurons. There can be any number of hidden layers between the input layer and the output layer, but typically the number of 5 hidden layers in any particular ANN is limited.

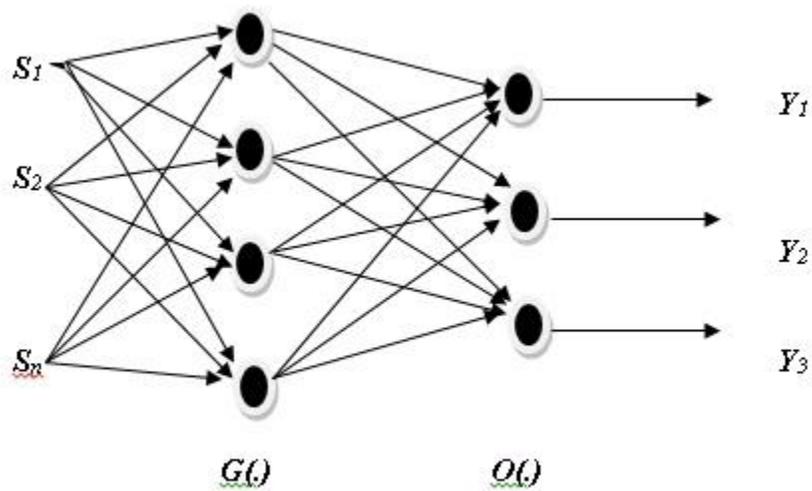


Figure 1.8 The basic architecture of ANN.

Figure 1.4 represents the basic architecture of Neural Networks. Each connection that connects one layer to another is connected through associated unique weight and each connection is treated separately. When the network learns, the weights are adjusted in order represent the

strength of connections between neurons. Thus, the network might determine that a particular connection is of no value, and give it a weight of zero.

An artificial neural network represents a non-linear model characterized by the number of neurons, its network topology, activation functions, and the values of synaptic weights and biases.

1.5 Topology and Complexity

Artificial neural networks can utilize many different topologies to learn the same set of data because of their robustness. The topology can impact the accuracy of the network to classify new data and the amount of time required to learn the desired response. Thus, choosing a good topology becomes a crucial task to the success of the neural network. The figures given below represent different topologies according to their classification [6].

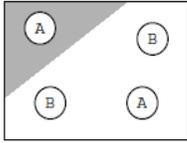
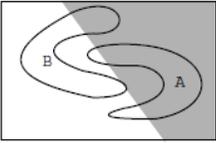
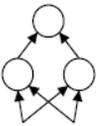
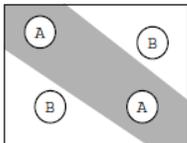
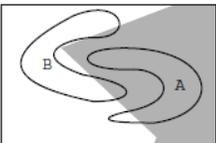
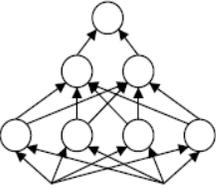
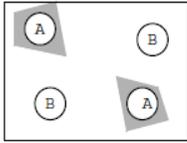
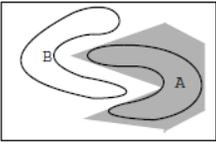
Topology	Type of Decision Region	XOR Problem	Classes with Meshed Regions	Most General Region Shapes
	Half bounded by hyperplanes			
	Convex Open or Closed Regions			
	Arbitrary Regions (Complexity limited by the number of nodes)			

Figure 1.9 Different topologies for Neural Networks

1.6 The Perceptron

The Perceptron is a simple structure consist of a single-layer artificial network with only one neuron. The input of a perceptron is real valued or logical 1 or 0 that is multiplied with the corresponding weights to form the linear combination and passes through the activation function to produce the output Y [7].

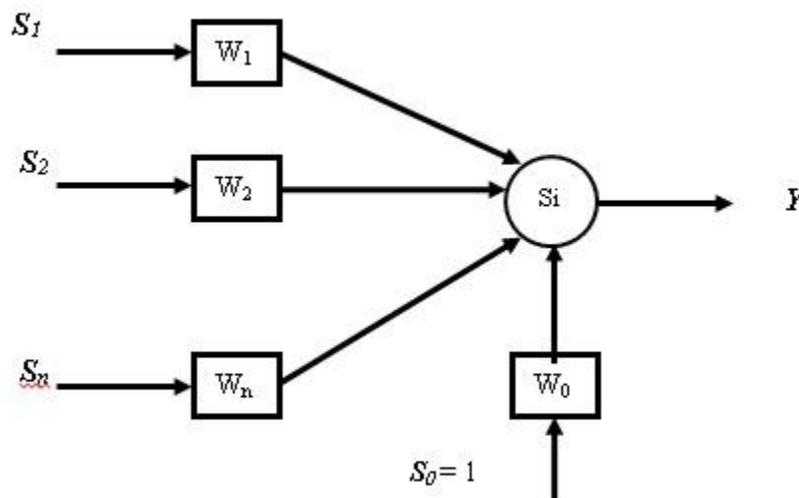


Figure 1.10 Perceptron

where S_i are the components of the input $\mathbf{S} = (S_1, S_2, \dots, S_n)$ from the set $\{(S, y)\}$

Activation function is the step function defined as follows.

Step function:

$$G(Z_j) = 1 \quad \text{if } Z_j > 0$$

$$G(Z_j) = 0 \quad \text{if } Z_j = 0$$

$$G(Z_j) = -1 \quad \text{if } Z_j < 0$$

Where

$$Z_j = \sum_{i=1}^n w_{ji} \cdot S_i \quad \text{Activation value}$$

$$\theta = S_0 \cdot W_0 \quad \text{Threshold}$$

The Perceptron classify the data depending upon the sum which is greater than the threshold value $Z_j > -w_0$ or it is less than the threshold value $Z_j < -w_0$. In the above formulation, the w_0 defines the threshold value which is the weight of an additional input that is held constant to $S_0 = 1$. Basically it bias the perceptron towards the respected value which is may be 1 or 0. For these reasons we can refer perceptron as threshold logic unit (TLU).

A linear discriminant is a device that decides whether an input pattern belongs to one more classes which is equivalent to the perceptron.

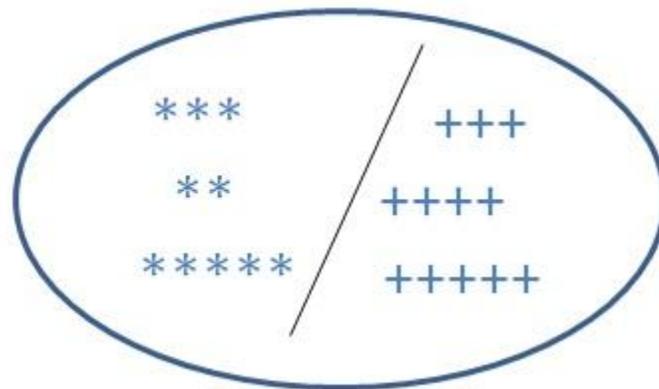


Figure 1.11 Linear Discriminator

A Network of Linear discriminat can represent a variety of functions, on the other hand a single unit is not sufficient to do so. In the case of digital circuits, every boolean function can be presented by the network of interconnected units. Here are the examples that represent the perceptron to perform different logical functions.

1.6.1 Example: Logic Operator

A Perceptron can calculate most of the primitive boolean functions including AND, OR, NAND and NOR but cannot calculate XOR because it requires a plane for classification not single line.

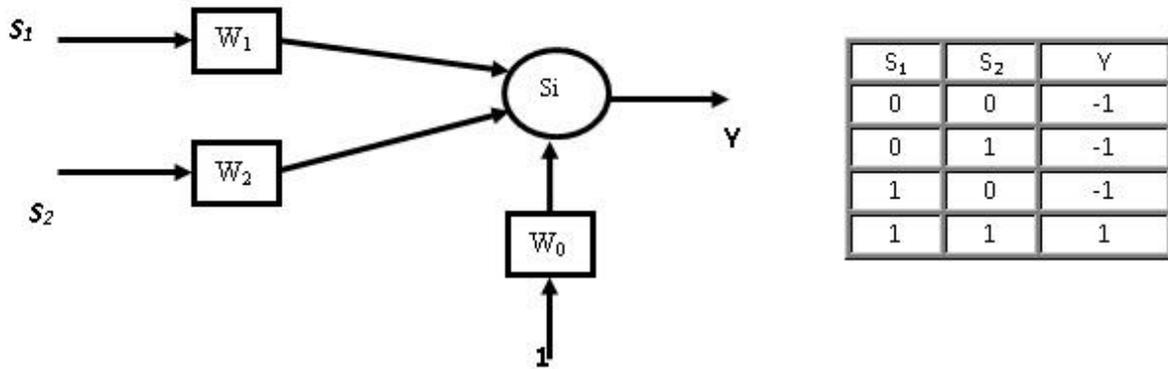


Figure 1.12 Perceptron as a logic AND

Where $w_1 = 3/2$ $w_2 = 1$ and $w_0 = -2$

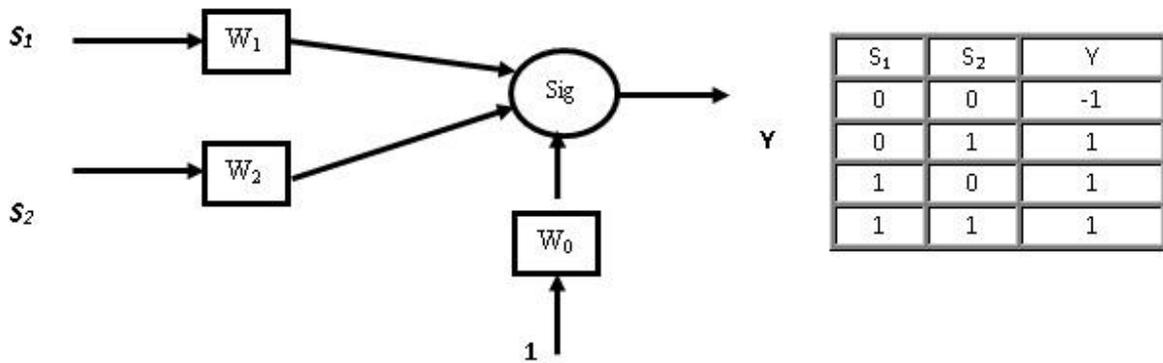


Figure 1.13 Perceptron as a logic OR

Where $w_1 = 3/2$ $w_2 = 1$ and $w_0 = -2$

The network output is

$$y = S_1 \cdot w_1 + S_2 \cdot w_2 + w_0$$

1.6.2 Decision boundary for the Perceptron

Decision boundary defines the boundary at which the output of the perceptron is precisely equal to the threshold θ , i.e. $\sum S_i W_i = \theta$

In the case of 1-D, the surface is just a point:

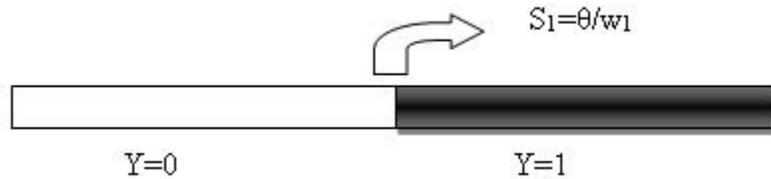


Figure 1.14 Decision boundary

In the case of 2-D, the decision boundary (between $y = -1$ and $y = 1$) is at

$$0 = S_1 \cdot w_1 + S_2 \cdot w_2 + w_0$$

$$S_2 \cdot w_2 = -S_1 \cdot w_1 - w_0$$

$$S_2 = -\frac{w_1}{w_2} S_1 - \frac{w_0}{w_2} \quad I$$

$$y = mx + b \quad II$$

So, in 2-D the decision boundaries are always straight lines by comparing Equation I with equation of line (Equation II). Where m is the slope of the line and b is the Y intercept.

We can now plot the decision boundaries of logic gates

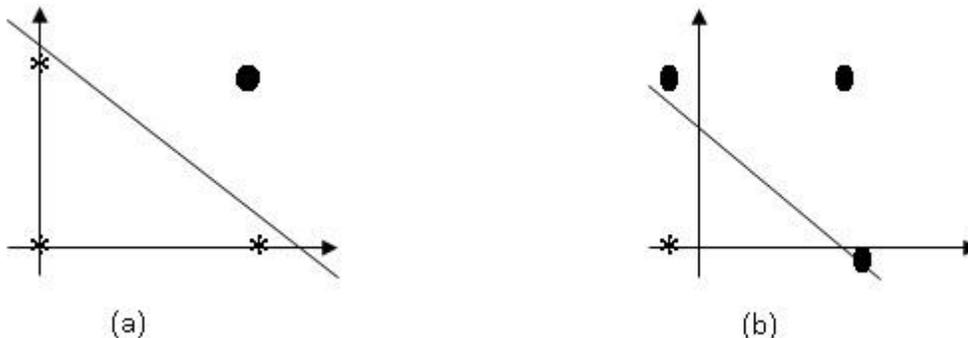


Figure 1.15 Decision boundary for (a) AND (b) OR

Legend

0 = 1

X = -1

1.7 Hebb Learning Rule

In 1949 neuropsychologist Donald Hebb postulated how biological neurons learn:

“When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place on one or both cells such that A’s efficiency as one of the cells firing B, is increased.”

In other words, If two neurons on either side of a synapse (connection) are activated simultaneously (i.e. synchronously), then the strength of that synapse is selectively increased.

Hebbian learning can be summarized as follow[8]

$$w_i^{n+1} = w_i^n + \Delta w_i$$
$$\Delta w_i = \delta \cdot t \cdot s_i$$

Where

- δ : learning rate
- t: the desired output
- S_i : the ith perceptron input

An obvious problem with this rule is that it is unstable – chance coincidences will build up the connection strengths, and all the weights will tend to increase indefinitely. Consequently, the basic learning rule is often supplemented by:

If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.

Another way to stop the weights increasing indefinitely involves normalizing them so they are constrained to lie between 0 and 1.

1.7.1 Hebbian Learning of the AND Operator

Suppose we start from a random initialization of $w_1 = 0, w_2 = 1$ and $w_0 = 0$ using a learning rate $\delta = \frac{1}{2}$

$$x_2 = -\frac{w_1}{w_2} - \frac{w_0}{w_2}$$

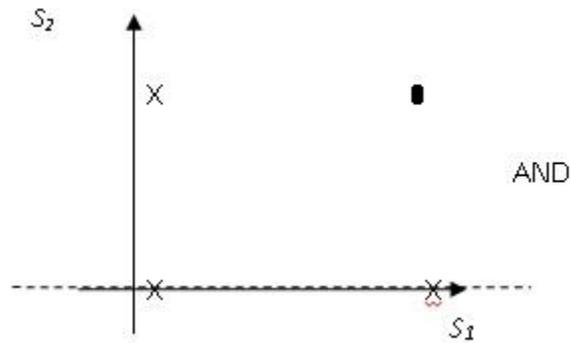


Figure 1.16 Decision boundary for AND operator for first Epoch

Epoch 1

$$w_i^{n+1} = w_i^n + \Delta w_i$$

$$\Delta w_i = \delta \cdot t \cdot s_i$$

Record 1:

$$w_1 = 0 + 0 = 0$$

$$w_2 = 1 + 0 = 1$$

$$w_0 = 0 - 1/2 = -1/2$$

Record 2:

$$w_1 = 0 + 0 = 0$$

$$w_2 = 1 - 1/2 = 1/2$$

$$w_0 = -1/2 - 1/2 = -1$$

Record 3:

Ok

Record 4:

$$w_1 = 0 + 1/2 = 1/2$$

$$w_2 = 1/2 + 1/2 = 1$$

S ₁	S ₂	S ₀	y	
0	0	1	-1	Record 1
0	1	1	-1	Record 2
1	0	1	-1	Record 3
1	1	1	1	Record 4

$$w_0 = -1 + 1/2 = -1/2$$

Epoch 2

Record 1:

OK

Record 2:

$$w_1 = 1/2 + 0 = 1/2$$

$$w_2 = 1 - 1/2 = 1/2$$

$$w_0 = -1/2 - 1/2 = -1$$

Record 3:

Ok

Record 4:

$$w_1 = 1/2 + 1/2 = 1$$

$$w_2 = 1/2 + 1/2 = 1$$

$$w_0 = -1 + 1/2 = -1/2$$

S ₁	S ₂	S ₀	y	
0	0	1	-1	Record 1
0	1	1	-1	Record 2
1	0	1	-1	Record 3
1	1	1	1	Record 4

Epoch 3

Record 1

Ok

Record 2:

$$w_1 = 1 + 0 = 1$$

$$w_2 = 1 - 1/2 = 1/2$$

$$w_0 = -1/2 - 1/2 = -1$$

Record 3:

$$w_1 = 1 - 1/2 = 1/2$$

$$w_2 = 1/2 - 0 = 1/2$$

$$w_0 = -1 - 1/2 = -3/2$$

Record 4:

$$w_1 = 1/2 + 1/2 = 1$$

$$w_2 = 1/2 + 1/2 = 1$$

S ₁	S ₂	S ₀	y	
0	0	1	-1	Record 1
0	1	1	-1	Record 2
1	0	1	-1	Record 3
1	1	1	1	Record 4

$$w_0 = -3/2 + 1/2 = -1$$

.
. .
. .
. .
. .
. .

Epoch 8

Record 1:

Ok

Record 2:

$$w_1 = 3/2 - 0 = 3/2$$

$$w_2 = 3/2 - 1/2 = 1$$

$$w_0 = -3/2 - 1/2 = -2$$

Record 3: OK

Record 4: OK

X1	X2	X0	y	
0	0	1	-1	Record 1
0	1	1	-1	Record 2
1	0	1	-1	Record 3
1	1	1	1	Record 4

**Chapter 2 APPLICATION OF NEURAL NETWORKS
IN BIOMEDICAL**

2.1 Pattern Recognition

Pattern recognition is the scientific area that defines the methods for classification and provides the object description. It is a challenging and fascinating task to design and implement algorithms that emulates the human ability to classify the objects and describe different features of an object since the early times of computing. Therefore, Pattern recognition is a wide area of research, with many applications to other disciplines of engineering and sciences including biomedical.

Basically, Pattern recognition describes a methodology to perform a task of placing the objects to their correct class based on the measurements about the object. There are many algorithms that has been developed to perform such tasks automatically with the help of computer. Objects to be recognized, measurements about the objects, and possible classes can be almost anything in the world. For this reason, there are very different pattern recognition tasks. We can define a pattern recognition system as a system that can make measurements about certain objects and on the bases of these measurement it can classify the object to its desire class [9]. For example, In the case of gamma camera the grid of sensors consist of five sensors in each row and there are five row of sensors as shown in Figure 2.1. The architecture of gamma camera is discussed more detail in chapter 3. Each small circle depicted in figure 2.1 represent the number of classes along X axes and Y axes. There are 57 classes along X-axes and 49 classes along Y-axes. We may utilize the algorithms of pattern recognition in each direction X and Y.

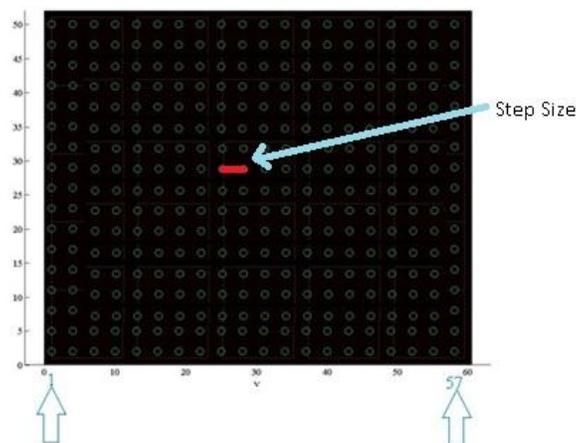


Figure 2.1 Grid of sensors in gamma camera to represent the classes (green circles)

2.2 Feedforward Neural Network

The idea of Neural networks with backpropagation learning algorithm was first presented by D.E. Rumelhart and J.L McClelland. They proposed multi-layer architecture of neural networks with feedforward flow of data through nodes that requires fully connected network between consecutive layers. This architecture leads towards the computing model of "universal approximator" which is based on Kolmogorov's Theorem. There are two main ideas behind the feedforward neural network. The first idea says that the output of each neuron from the previous layer is connected with the corresponding inputs of all neurons of the next layer that is the network is fully connected. The second idea is that the errors of the neurons from the output layer are sequentially propagated backward through all the layers from output layer to input layer in order to calculate the error of all neurons which is depicted in figure. Another important property of the feedforward neural networks is the use of sigmoid functions as an activation function.

As we can say that feedforward neural network is a universal approximator. We can approximate any continuous real valued function but depends on several factors which are as follows:

1. A feed-forward neural network is a non-parametric statistical model designed to provide nonlinear relations in the data. In general, the mapping supposes that there is an existence of unknown relationship between the input and the output variables. FNN as an approximator built that unknown relationship with the aid of learning data.
2. The objective is to find the smallest architecture that can accurately fits the function based on the number of hidden layers and the number of neurons on each layer.
3. The third factor is the control over the learning process which is usually implemented by learning rate.

We can make the network more flexible in terms of mapping by increasing the number of hidden layers and neurons in each layer. In the case of pattern classification, the problem is more likely to be linearly separable in a high dimensional feature space than in a low dimensional one, while projected into a high dimensional space nonlinearly. But the

computations involve in higher dimensional space require much more time for training and needed lot of resources. On the other hand, increasing the number of hidden neurons leads towards the overfitting mappings [10]. In order to minimize the implementation of complex mappings, several supporting algorithms were proposed. For example, In the case of gamma camera the grid of sensors consist of five sensors in each row and there are five rows of sensors as shown in Figure 2.1. The neural network utilize two matrices in order to train the Neural Network for X axis's, first is the reading of the 25 sensors and the 2nd matrix is the desired response as shown figure 2.3. The number of rows in each matrix depends on the number of repetitions at each point of interaction and the step size. The following formula gives the total number of rows in the matrices used by the neural network for X-axis's.

$$\text{No. of Rows} = \text{Length of grid} * \text{No. of repetitions at each point} / \text{Step Size.}$$

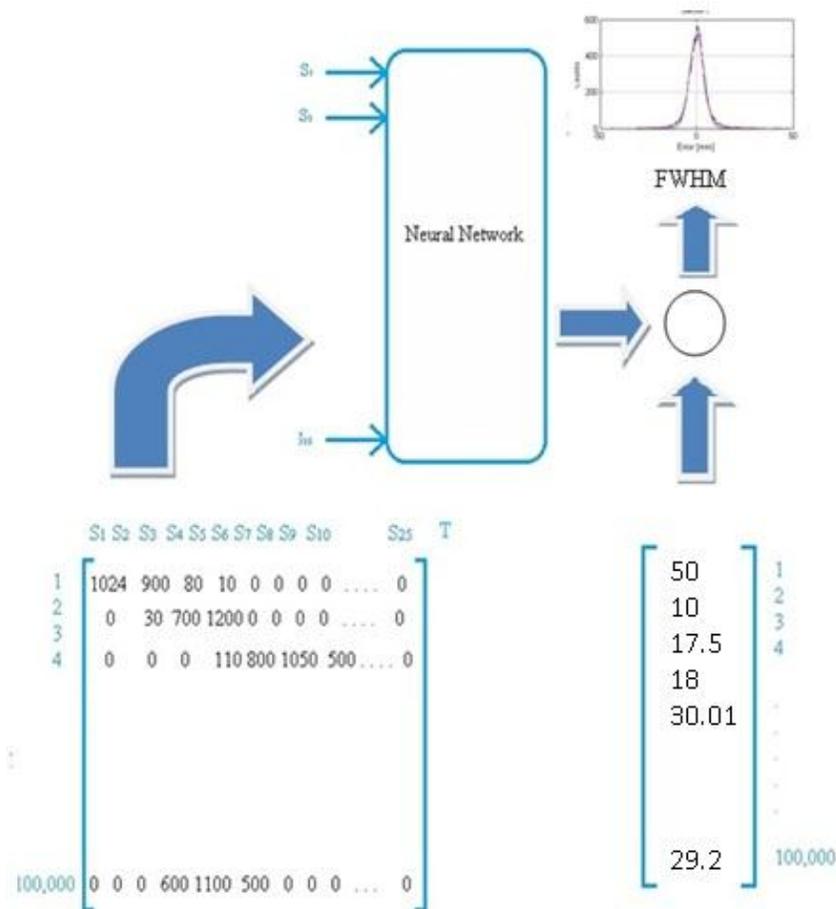


Figure 2.3 Input matrix (left) and the desired response matrix (right) for Feedforward Neural Network

Applications of feed forward neural network.

- Regression
- Classification

2.2.1 Regression:

Suppose there is a variable Y that we want to calculate based on some other variable X. So we can say that Y is a dependent variable and X is independent variable because Y depends on X. Moreover, suppose that there is a relationship between X and Y that could be linear or nonlinear. Consider a linear relationship between X and Y. There is another variable e which is distributed randomly i.e, with uniform distribution, we call it 'disturbance' or 'error'[11].

Let I be the index of observations on the data pairs X and Y. The simple linear model could be stated as:

$$Y_i = a_0 + a_1 X_i + e_i$$

The parameters a_0 and a_1 represent the Y intercept and the slope of the relationship, respectively.

We may assume the following assumption about the error e of this simple model.

1. The mean of the error is zero i.e., $E(e_i) = 0$
2. It has same variance for all i. i.e. $E(e_i^2) = \sigma^2$
3. There is no correlation between the observation i.e. $E(e_i e_j) = 0$

The task of the estimator is to calculate the parameter a_0 and a_1 from the available data represented by \hat{a}_0 , \hat{a}_1 . So the estimated output becomes

$$\hat{Y}_i = \hat{a}_0 + \hat{a}_1 X_i$$

We can define the estimated becomes

$$\hat{e}_i = Y_i - \hat{Y}_i$$

In a scatter diagram of Y against X shown in figure 19 depicts the vertical distance between observed Y_i and the estimated value \hat{Y}_i .

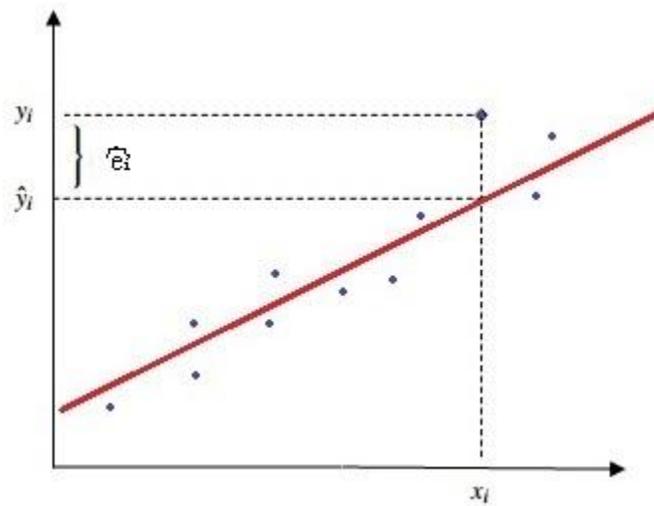


Figure 2.4 Linear Regression

We may utilize FNN in order to estimate the optimum parameters a_0 and a_1 .

In general the goal of the FNN is to estimate a target function t given the finite set of N observation as shown in figure 20.

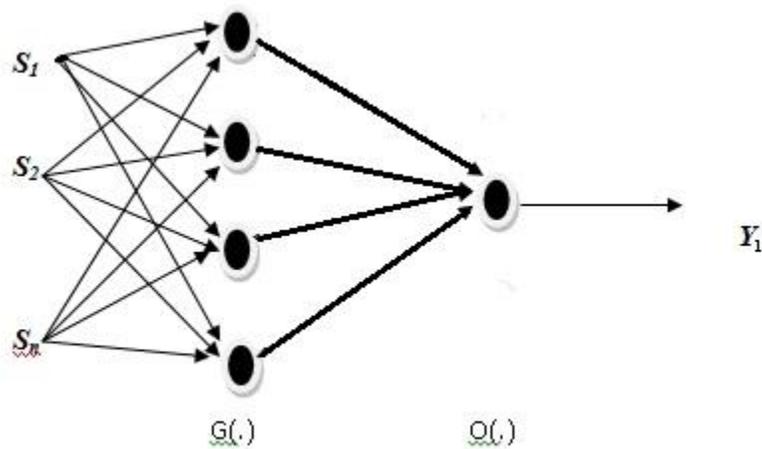


Figure 2.5 Feedforward Net Architecture

$Y_1 = G(\sum_j W_j \cdot O(\sum_i w_{ji} \cdot S_i))$ represents the output of FNN

$$t_n = \hat{Y}_t + e_n \quad \text{where } e_n \text{ belongs to } N(0, \sigma^2)$$

Where t_1, t_2, \dots, t_N are the samples of i.i.d observation from $N(0, \sigma^2)$

In the case of regression problems the learning behind FNN is the Maximum likelihood estimation. Let us suppose that we have n observation i.i.d (Identically independent distribution) process t_1, t_2, \dots, t_N . In probability theory, a sequence of random variables is independent and identically distributed (i.i.d.) if each random variable has the same probability distribution as the others and all are mutually independent. Let define the likelihood L of the samples as its probability.

$$L(W) = \prod_n^N \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2} (t_n - Y_n)^2} \dots\dots\dots(2.1)$$

Where W is the set of weights and the purpose is that to find optimum weights that maximize L(W). Taking log both sides.

$$\underset{W}{\operatorname{argmax}} L(W) = \underset{W}{\operatorname{argmax}} \sum_n^N [\log(\frac{1}{\sqrt{2\pi\sigma}} - \frac{1}{2\sigma^2} (t_n - Y_n)^2)]$$

$$\operatorname{argmax}_W L(W) = \operatorname{argmax}_W \sum_n^N \left[\log \frac{1}{\sqrt{2\pi}\sigma} - \log \frac{1}{2\sigma^2} (t_n - Y_n)^2 \right]$$

$$\operatorname{argmax}_W L(W) = \operatorname{argmax}_W \sum_n^N \left[0 - \log \frac{1}{2\sigma^2} (t_n - Y_n)^2 \right]$$

$$\operatorname{argmax}_W L(W) = \operatorname{argmax}_W - \sum_n^N \frac{1}{2\sigma^2} (t_n - Y_n)^2$$

$$\operatorname{argmin}_W L(W) = \operatorname{argmin}_W \sum_n^N (t_n - Y_n)^2$$

$$\operatorname{argmin}_W L(W) = \operatorname{argmin}_W \sum_n^N (e_n)^2$$

$$\operatorname{argmin}_W L(W) = \operatorname{argmin}_W E (e_n)^2 \quad \dots\dots\dots (2.2)$$

The equation ii show that we can get the optimum weights if the mean of the square of the error function e_n comes to at the global minimum. So it is a recursive process and called training of the network. Another important point that depicts this equation is that there is the square of the error because the error function is both positive and negative.

2.2.2 Classification:

In statistics and machine learning , classification is the problem to identify a new observation belongs to which of a set of categories, on the basis of its attributes and the available training data. Each observation is analyzed on the basis of a set of quantifiable properties, known as various *features or attributes*. These properties may be define as a dependent or independent variables based on their probabilistic distribution and could be belong to following types as,

1. ordinal (e.g. "large", "medium" or "small"),
2. categorical (e.g. "A", "B", "AB" or "O", for blood type),
3. integer-valued (e.g. the number of occurrences of a part word in an email)
4. real-valued (e.g. a measurement of blood pressure).

An algorithm that implements classification, especially implementation in programming language, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, that maps input data to a category. Consider the general architecture of ANN with output y as shown in figure 2.3

$$y = G(\sum_j W_j \cdot O(\sum_i w_{ji} \cdot S_i))$$

In statistics, the goal of classification is to separate two or more classes according to the posterior probability. Let C_0 and C_1 be the two classes and the t ($t = 0$ if $t \in C_0$ and $t = 1$ if $t \in C_1$) is the variable that we want to find probability. The posterior probability is the probability of the parameters t given the evidence X as

$$p(t|X) = y^t (1 - y)^{1-t} \quad t \sim \text{Be}(y)$$

Let we have samples $t_1, t_2, t_3, \dots, t_n$ are of i.i.d observation from bernulli distribution. We may define L as likelihood probability of samples

$$L(W) = \prod_n y_n^{t_n} (1 - y_n)^{1-t_n} \quad \dots\dots\dots(2.4)$$

Where W is the set of weights and the purpose is that to find optimum weights that maximize $L(W)$. Taking log both sides.

$$\log(L(W)) = \log\left(\prod_n y_n^{t_n} (1 - y_n)^{1-t_n}\right)$$

$$\text{argmax}_W L(W) = \text{argmax}_W \sum_n [\log(y_n^{t_n} (1 - y_n)^{1-t_n})]$$

$$\text{argmax}_W L(W) = \text{argmax}_W \sum_n [\log(y_n^{t_n}) + \log((1 - y_n)^{1-t_n})]$$

$$\text{argmax}_W L(W) = \text{argmax}_W \sum_n [t_n \log(y_n) + (1 - t_n) \log(1 - y_n)] \quad \dots\dots\dots (2.5)$$

The equation 2.5 represent the cross entropy and it minimization is equivalent to the minimization of the kullback-leibler divergence of the network output and the target distribution.

2.3 Supervised Learning in Neural Network

During a training process, the neural network fits to the data by learning algorithms. These learning algorithms are characterized to utilize the given output which is the target function t that is compared to the predicted output which is the actual output of the neural network Y and by the adaptation of all parameters according to this comparison. The parameters of a neural network are its weights. The figure 21 represents the general methodology of supervised learning.

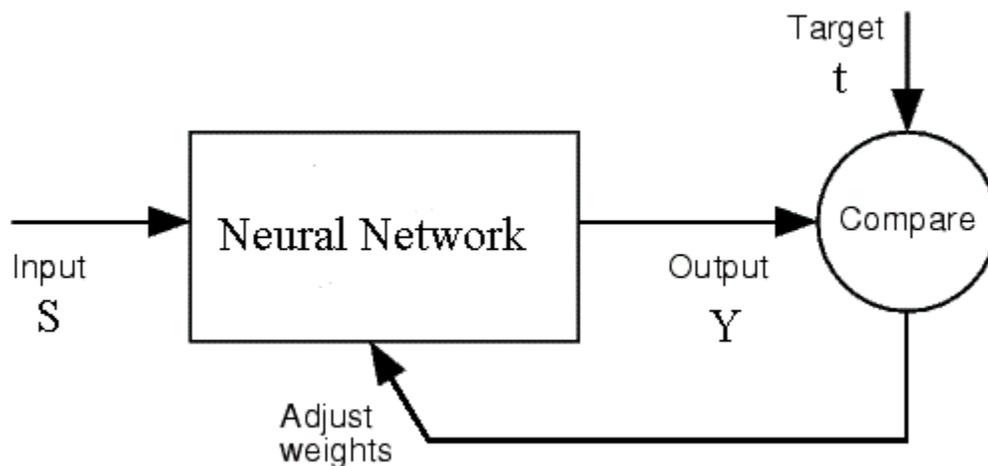


Figure 2.6 Supervise Learning

At the start of the training, all the weights are usually initialized with random values formulated from a standard normal distribution. Initialization is very critical for some application because different initialization methods have significant impact on trained network. During an iterative training process, the following steps are repeated:

- The neural network calculates an output $Y(s)$ for given inputs S and current weights. If the training process is not yet completed, the predicted output Y will differ from the target t .
- An error function E , like the sum of squared errors (SSE) or mean square error

$$\underset{W}{\operatorname{argmin}} L(W) = \underset{W}{\operatorname{argmin}} E (e_n)^2$$

or the cross-entropy

$$\underset{W}{\operatorname{argmax}} L(W) = \underset{W}{\operatorname{argmax}} \sum_n^N [t_n \log(y_n) + (1 - t_n) \log(1 - y_n)]$$

Measures the difference between predicted and observed output, where $n = 1, \dots, N$ indexes the observations, i.e. given input-output pairs.

- All weights are adapted according to the rule of a learning algorithm.

The training will stop if a pre-specified training criterion is fulfilled ,e.g. if all absolute partial derivatives of the error function with respect to the weights are smaller than a given threshold
There are different methods to minimize the error iteratively which are very similar to each other including.

- Delta Rule
- Widrow Hoff Rule
- Adaline Rule
- Backpropagation

2.4 Backpropagation

In the previous section we saw that neural networks with a single layer of computing units are capable of computing wider range of Boolean function. However the computation required to find the correct combination of weights increases substantially when more parameters and more complicated topologies are considered. In this section we discuss a popular learning method capable of handling such large learning problems — the backpropagation algorithm. This numerical method was used by different research communities in different contexts, was discovered and rediscovered, until 1985. It has been one of the most studied and used algorithms for neural networks learning ever since.

Backpropagation is not only more general than the usual analytical derivations, which handle only the case of special network topologies, but also much easier to follow. It also shows how the algorithm can be efficiently implemented in computing systems in which only local information can be transported through the network.

The basic methodology behind backpropagation is that It is based on supervised learning. It requires a dataset of the desired output for every inputs making up the training set, and is a generalization of the delta rule. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). Before moving to the mathematical derivation of the backpropagation algorithm, it is convenient to develop some intuitions about the relationship between the actual output of a neuron and the desired output for a particular training case as show in figure below. The flow of information is from the input to output and the flow of error is from output to input used to calculate new weights[12].

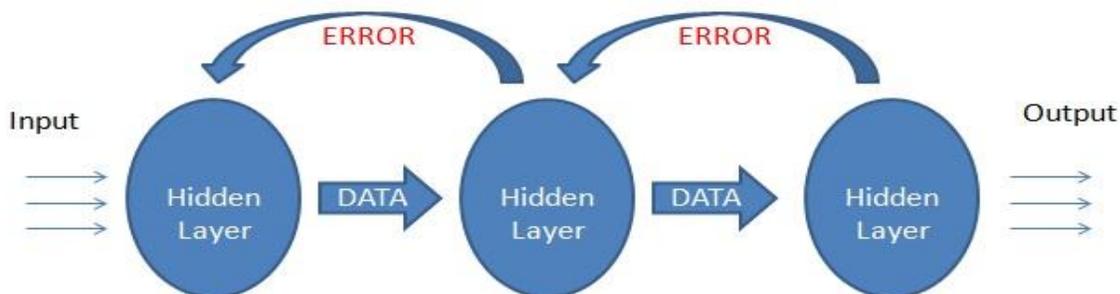


Figure 2.7 Backpropagation

2.4.1 Formulation of the BP Algorithm

Assuming that the discrete-time desired output y_n is given,

$$y_n = g\left(\sum_j W_j h\left(\sum_i W_{ji} \cdot S_i\right)\right)$$

$$A = \sum_j W_j h\left(\sum_i W_{ji} \cdot S_i\right)$$

$$a_j = \sum_i W_{ji} \cdot S_i$$

$$b_j = h(a_j)$$

We can motivate the backpropagation learning algorithm as gradient descent on sum-squared error (we square the error because we are interested in its magnitude, not its sign). The total error defined as

$$e(w) = \sum_n (t_n - y_n)^2 = \sum_n (t_n - g(A))^2$$

Take partial derivative on both sides with respect to W_j in order to find weights update rule for W_j

$$\frac{\partial e(w)}{\partial W_j} = 2 \sum_n (t_n - g(A)) \cdot \frac{\partial (t_n - g(A))}{\partial W_j}$$

$$\frac{\partial e(w)}{\partial W_j} = 2 \sum_n (t_n - g(A)) \cdot (-g(A)') \frac{\partial A}{\partial W_j}$$

$$\frac{\partial e(w)}{\partial W_j} = 2 \sum_n (t_n - g(A)) \cdot (-g(A)') \cdot b_j$$

So the update rule for W_j becomes

$$W_j^{k+1} = W_j^k + 2\mu \sum_n (t_n - g(A)) \cdot (-g(A)') \cdot b_j$$

Take partial derivative on both sides with respect to w_{ji} in order to find weights update rule for w_{ji}

$$\frac{\partial e(w)}{\partial w_{ji}} = 2 \sum_n^N (t_n - g(A)) \cdot \frac{\partial (t_n - g(A))}{\partial w_{ji}}$$

$$\frac{\partial e(w)}{\partial w_{ji}} = 2 \sum_n^N (t_n - g(A)) \cdot (-g(A)') \frac{\partial A}{\partial w_{ji}}$$

$$\frac{\partial e(w)}{\partial w_{ji}} = 2 \sum_n^N (t_n - g(A)) \cdot (-g(A)') \cdot W_j \frac{\partial}{\partial w_{ji}} b_j$$

$$\frac{\partial e(w)}{\partial w_{ji}} = 2 \sum_n^N (t_n - g(A)) \cdot (-g(A)') \cdot W_j \cdot h'(a_j) \frac{\partial}{\partial w_{ji}} a_j$$

$$\frac{\partial e(w)}{\partial w_{ji}} = 2 \sum_n^N (t_n - g(A)) \cdot (-g(A)') \cdot W_j \cdot h'(a_j) S_i$$

So the update rule for w_{ji} becomes

$$w_{ji}^{k+1} = w_{ji}^k + 2\mu \sum_n^N (t_n - g(A)) \cdot (-g(A)') \cdot W_j \cdot h'(a_j) \cdot S_i$$

2.5 Optimization

In previous section, the problem of learning in neural networks has been formulated in terms of the minimization of an error function E . This error is a function of the weights and biases in the network, because y_n is a function of weights and biases. we may represent these weights (W_j and w_{ji}) into a single vector W . The problem addressed in this section is to find a weight vector W which minimizes an error function $E(w)$. It is useful to have a simple geometrical picture of the error minimization process, which can be obtained by viewing $E(w)$ as the error surface sitting above weight space, as shown in Figure.

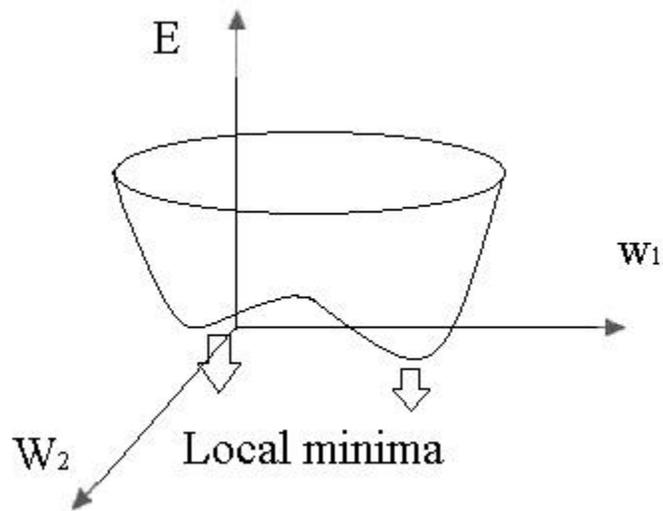


Figure 2.8 Error surface 3D

For networks with a single layer of weights having linear activation functions in the output layer and a cost function as sum-of-squares error, the error function will become a quadratic function of the weights and the error surface will have multidimensional parabolic form. In this case the error function has a single minimum which can be located by solution of linear equations[9].

However, for more general networks, in particular those with more than one hidden layers, the error function will be a highly nonlinear function of the weights, and there may exist many minima all of which satisfy the following condition

$$\nabla e(w) = 0 \quad \dots\dots\dots (2.6)$$

where ∇E denotes the gradient of error function in weight space. The minimum point at which the value of the error function is smallest one is called the global minimum while other minima are called local minima. There may be some other points which satisfy the condition of equation 2.6, such points are called local maxima or saddle points. Any vector W which satisfy the condition of equation (1) is called a stationary point, and there are different kinds of stationary point are illustrated in Figure below.

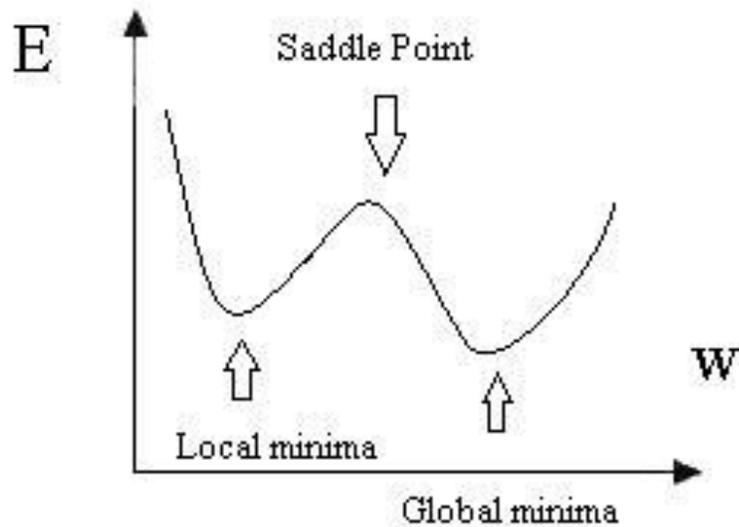


Figure 2.9 Error as a function of weights

As a result of the non-linearity of the error function, it is not in general possible to find closed-form solutions for the minima. Instead, we consider methods which involve a search through weight space consisting of a succession of steps of the form

$$W^{n+1} = W^n + \Delta W^n$$

Where n labels the iteration step. Different algorithms involve different choices for the weight vector increment ΔW^n . Recently several adaptive learning algorithms has been discovered for feed-forward neural networks. In optimization theory, many of these algorithms are based on the gradient descent algorithm. These methods usually have a poor convergence rate and depend on parameters which have to be specified by the user such as step rate. There is no theoretical method exists to optimize these parameters. The optimum values of these parameters are very crucial for the training of neural network. As in the case of backpropagation algorithm which often behaves very badly on large-scale problems and which success depends of the user dependent parameters learning rate and momentum constant.

2.5.1 Steepest Descent Algorithm

The steepest descent method is the simplest of all the gradient methods. Consider a differentiable mean square error function $e(w)$, which is defined within a given boundary. The gradient of such function $e(w)$ defines the slope of the error function $e(w)$. If the gradient is negative, the function decreases and on the other hand if gradient is positive the error function is increasing. It is useful to employ the method of steepest descent in order to find the local minimum of $e(w)$, where it uses a zig-zag like path start from arbitrary point X_0 and gradually slide down the gradient, until it converges to the actual point of minimum [13].

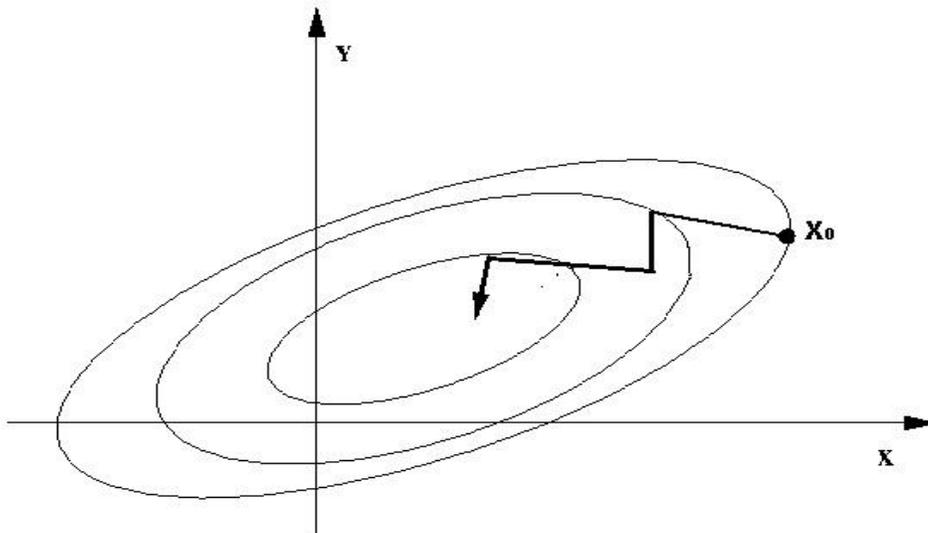


Figure 2.10 The Method of Steepest Descent Approaches the Local Minimum in a zig-zag path, and the next search direction would be orthogonal to the next

Let us define some commonly used indices before moving to the derivation:

- p is the index of patterns, from 1 to P , where P is the number of patterns.
- m is the index of outputs, from 1 to M , where M is the number of outputs.
- i and j are the indices of weights, from 1 to N , where N is the number of weights.
- k is the index of iterations.

The mean square error (MSE) is defined for the optimization of the training process. For all training patterns and network outputs, it is calculated by

$$\mathbf{E}(\mathbf{S}, \mathbf{W}) = \sum_{p=1}^P \sum_{m=1}^M (\mathbf{t}_{p,m} - \mathbf{y}_{p,m})^2 \quad \dots (2.7)$$

Where

S is the input vector

W is the weight vector

t is the desired output vector

y is the actual output vector

The steepest descent algorithm is a first-order algorithm. It uses the first-order derivative of total error function to find the minima in error space. Normally, gradient **g** is defined as the first-order derivative of total error function given in Equation (2.6)

$$\mathbf{g} = \left[\frac{\partial \mathbf{E}(\mathbf{S}, \mathbf{W})}{\partial w_1} \quad \frac{\partial \mathbf{E}(\mathbf{S}, \mathbf{W})}{\partial w_2} \quad \frac{\partial \mathbf{E}(\mathbf{S}, \mathbf{W})}{\partial w_3} \quad \dots \quad \frac{\partial \mathbf{E}(\mathbf{S}, \mathbf{W})}{\partial w_N} \right] \quad \dots (2.8)$$

We may define the update rule of the steepest descent algorithm by using the definition of gradient vector **g** as

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha \mathbf{g}_k \quad \dots (2.9)$$

Where α is the step size.

The training process of the steepest descent algorithm is asymptotic convergence. All the elements of the gradient vector **g** could be very small around the solution and there may be a very small change in weights. Because of the fast convergence and easy implementation of this method, it is popular among many areas of research. The biggest advantage of this method is that it guarantees to find the minimum through numerous times of iterations as long as it exists. It produces the optimum results if the step size is chosen properly; it is the compromise between the convergence speed and the optimum results. However a larger step size will increase the convergence speed, but it could also result in an estimate with large error.

2.5.2 Newton's Method

The next method which is the alternative to Steepest Descent Algorithm for fast optimization of neural networks is the Newton's method. It assumes that all weights are linearly independent and all the gradient components g_1, g_2, \dots, g_N are functions of weights as shown below:

$$g_1 = F_1 (w_1, w_2, w_3, \dots w_N)$$

$$g_2 = F_2 (w_1, w_2, w_3, \dots w_N)$$

.

.

.

$$g_N = F_N (w_1, w_2, w_3, \dots w_N)$$

Where F_1, F_2, \dots, F_N are nonlinear relationships between weights and related gradient components.

Each g_i ($i = 1, 2, \dots, N$) can be expanded by first-order approximation of Taylor series as shown below:

$$\begin{bmatrix} g_1 \approx \frac{\partial E}{\partial w_1} + \frac{\partial g_1}{\partial w_1} \Delta w_1 + \frac{\partial g_1}{\partial w_2} \Delta w_2 + \dots + \frac{\partial g_1}{\partial w_N} \Delta w_N \\ g_2 \approx \frac{\partial E}{\partial w_2} + \frac{\partial g_2}{\partial w_1} \Delta w_1 + \frac{\partial g_2}{\partial w_2} \Delta w_2 + \dots + \frac{\partial g_2}{\partial w_N} \Delta w_N \\ \vdots \\ g_N \approx \frac{\partial E}{\partial w_N} + \frac{\partial g_N}{\partial w_1} \Delta w_1 + \frac{\partial g_N}{\partial w_2} \Delta w_2 + \dots + \frac{\partial g_N}{\partial w_N} \Delta w_N \end{bmatrix} \dots (2.10)$$

We may define the gradient vector \mathbf{g} as

$$\mathbf{g} = [g_1 \quad g_2 \quad g_3 \quad \dots \quad g_N]$$

Where g_i is defined below according to equation 2.8

$$g_i = \frac{\partial E}{\partial w_i}$$

By inserting g_i in the equation 2.10 to get:

$$\begin{bmatrix} g_1 \approx \frac{\partial E}{\partial w_1} + \frac{\partial^2 E}{\partial w_1^2} \Delta w_1 + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_1 \partial w_N} \Delta w_N \\ g_2 \approx \frac{\partial E}{\partial w_2} + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_1 + \frac{\partial^2 E}{\partial w_2^2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_2 \partial w_N} \Delta w_N \\ \vdots \\ g_N \approx \frac{\partial E}{\partial w_N} + \frac{\partial^2 E}{\partial w_N \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_N \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_N^2} \Delta w_N \end{bmatrix} \quad (2.11)$$

By comparing the gradient components of steepest descent method given in Equation 2.1 with the gradient components given in Equation 2.5 we come to know that the second-order derivatives of the total error function need to be calculated for each component of gradient vector [14].

In order to minimize the total error function E , each element of the gradient vector given in Equation 2.11 should be zero (The minimum error possible).

$$\frac{\partial E}{\partial w_1} + \frac{\partial^2 E}{\partial w_1^2} \Delta w_1 + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_1 \partial w_N} \Delta w_N \approx 0$$

$$\frac{\partial E}{\partial w_2} + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_1 + \frac{\partial^2 E}{\partial w_2^2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_2 \partial w_N} \Delta w_N \approx 0$$

.

.

.

$$\frac{\partial E}{\partial w_N} + \frac{\partial^2 E}{\partial w_N \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_N \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_N^2} \Delta w_N \approx 0$$

Solving above equation to get

$$\begin{aligned}
 -\frac{\partial E}{\partial w_1} &\approx \frac{\partial^2 E}{\partial w_1^2} \Delta w_1 + \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_1 \partial w_N} \Delta w_N \\
 -\frac{\partial E}{\partial w_2} &\approx \frac{\partial^2 E}{\partial w_1 \partial w_2} \Delta w_1 + \frac{\partial^2 E}{\partial w_2^2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_2 \partial w_N} \Delta w_N \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 -\frac{\partial E}{\partial w_N} &\approx \frac{\partial^2 E}{\partial w_N \partial w_1} \Delta w_1 + \frac{\partial^2 E}{\partial w_N \partial w_2} \Delta w_2 + \dots + \frac{\partial^2 E}{\partial w_N^2} \Delta w_N
 \end{aligned}$$

Above Equations can be also written in the form of matrix

$$\begin{bmatrix} -\frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \\ \vdots \\ -\frac{\partial E}{\partial w_N} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_1 \partial w_2} & \frac{\partial^2 E}{\partial w_2^2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} X \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_N \end{bmatrix} \dots \quad (2.12)$$

There are N equations for N parameters so that all Δw_i can be calculated. With the solutions, the weight space can be updated iteratively.

$$\mathbf{g} = \begin{bmatrix} \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \\ \vdots \\ \frac{\partial E}{\partial w_N} \end{bmatrix} \dots (2.13)$$

$$\Delta \mathbf{W} = \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_N \end{bmatrix} \dots (2.14)$$

where the square matrix is Hessian matrix:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_1 \partial w_N} \\ \frac{\partial^2 E}{\partial w_1 \partial w_2} & \frac{\partial^2 E}{\partial w_2^2} & \dots & \frac{\partial^2 E}{\partial w_2 \partial w_N} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial^2 E}{\partial w_N \partial w_1} & \frac{\partial^2 E}{\partial w_N \partial w_2} & \dots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \dots (2.15)$$

By combining Equations 2.13, 2.14 and 2.15 by using Equation 2.12 to get

$$-\mathbf{g} = \mathbf{H}\Delta \mathbf{W}$$

Therefore, the update rule for Newton's method is

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mathbf{H}_k^{-1} \mathbf{g}_k \dots (2.16)$$

As the second-order derivatives of total error function, Hessian matrix \mathbf{H} gives the proper evaluation on the change of gradient vector. By comparing Equations 2.9 and 2.16, one may notice that well matched step sizes are given by the inverted Hessian matrix.

2.5.3 Levenberg–Marquardt Algorithm

In order to make sure that the approximated Hessian matrix $J^T J$ is invertible, Levenberg–Marquardt algorithm introduces another approximation to Hessian matrix:

$$H = J^T J + \mu I \quad \dots \quad (2.17)$$

where

μ is always positive, called combination coefficient

I is the identity matrix

The Jacobian is a matrix of all first-order partial derivatives of a vector-valued function. In the neural network case, it is a **N**-by-**W** matrix, where **N** is the number of entries in our training set and **W** is the total number of parameters (weights + biases) of our network. It can be created by taking the partial derivatives of each output in respect to each weight, and has the form[15]

$$H = \begin{bmatrix} \frac{\partial F(S_1, W)}{\partial w_1} & \frac{\partial F(S_1, W)}{\partial w_2} & \dots & \frac{\partial F(S_1, W)}{\partial w_N} \\ \frac{\partial F(S_2, W)}{\partial w_1} & \frac{\partial F(S_2, W)}{\partial w_2} & \dots & \frac{\partial F(S_2, W)}{\partial w_N} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial F(S_N, W)}{\partial w_1} & \frac{\partial F(S_N, W)}{\partial w_2} & \dots & \frac{\partial F(S_N, W)}{\partial w_N} \end{bmatrix} \quad \dots \dots \dots \quad (2.18)$$

From Equation 2.18, one may notice that the elements on the main diagonal of the approximated Hessian matrix will be larger than zero. Therefore, with this approximation (Equation 2.18), it can be sure that matrix **H** is always invertible.

By combining Equations 2.16 and 2.17, the update rule of Levenberg–Marquardt algorithm can be presented as

$$\mathbf{W}_{k+1} = \mathbf{W}_k - (\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})_k^{-1} \mathbf{g}_k$$

Chapter 3 Hicam - High resolution gamma camera

From the Hicam project webpage[12]: The purpose of this project is the development of high resolution Anger camera to be used for reliable and earlier diagnosis of cancer diseases. The system is developed for applications involves high overall spatial resolution (less than 3 mm) and system compactness is required. The architecture of gamma camera is derive from Anger architecture having a collimator and a continuous scintillator. The collimator serves as mechanical sieve for incoming gamma photons. The scintillator absorbs energy from each gamma photon and emits visible photons. The electrical signals are than generated from an array of photo-detectors which absorb the visible photons. The Silicon Drift Detector (SDD) is a special type of detector which is used to improve the system performance and is found to perform better as compared to commonly used photomultiplier tubes.

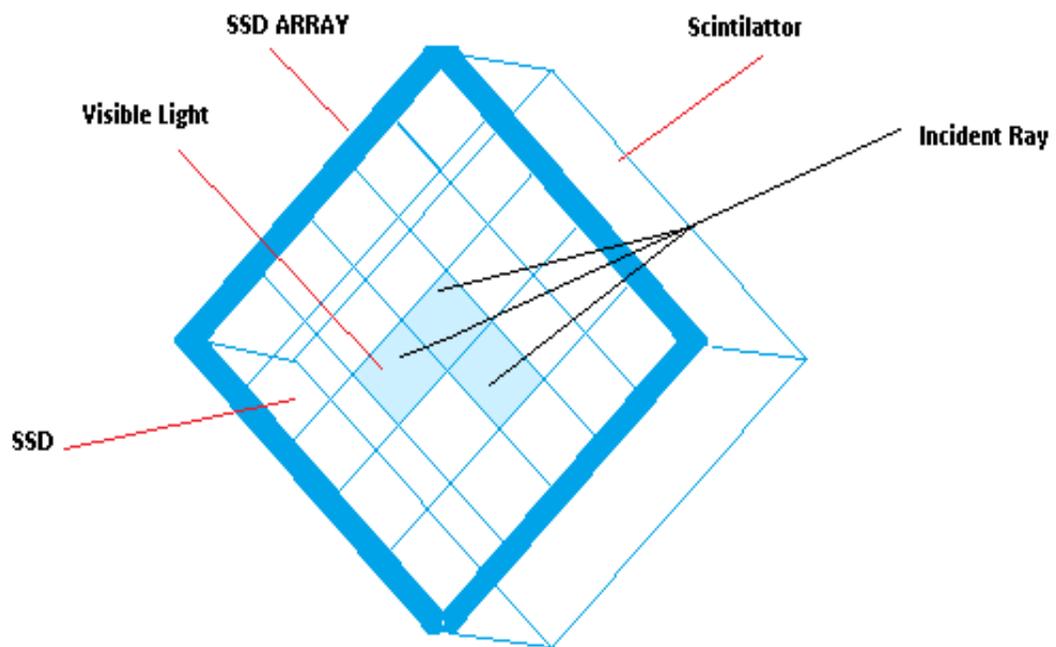


Figure 3.1 Gamma camera

The camera can be used in an annular holder of small diameter for SPECT imaging and also single handedly for planar scintigraphic studies. Compactness and high spatial resolution, are

the two prominent features which are helpful in diagnosing early stage cancer affecting human body. Such imaging is usually difficult to obtain using heavy imaging heads of commercial Anger cameras. Another objective to achieve in the project is the integration of camera with magnetic resonance instrumentation at the system level which is possible due to the large insensitivity of the SDD photo-detectors to large magnetic fields [17].

In short, Hicam is a gamma camera with high resolution camera ($100 \times 100 \text{mm}^2$) and its architecture is derived from Anger camera design. It uses high performance Silicon Drift Detectors as a replacement of photomultiplier tubes. The photo-detector array is made up of a square matrix of 10×10 SSD photo-detectors of the size of 1cm^2 .

3.1 Nuclear Scintigraphy

A gamma camera is a device based on a technique known as scintigraphy and it is used to capture images of radioisotopes which emit gamma radiation. The nuclear medicine imaging is based on scintigraphy and it is used to detect the presence of tumors in human body. The medicine is injected, inhaled or ingested in the human body and radiations are imaged. Non-invasive imaging techniques are gaining much interest in the latest cancer diagnosis and therapeutic proposes.

Imaging techniques such as magnetic resonance, emission of gamma radiation, bioluminescence and fluorescence are being combined in studies cellular processes in living organisms. The methods such as bioluminescence and fluorescence which were initially used for in vitro studies have been improved to consider for in vivo studies. These techniques have limited capabilities as the low energy light signals can be absorbed within a few centimeters in the tissue.

The advantage of combining imaging techniques is that it gives the opportunity to study biological processes at the molecular level. The observation of biological processes at molecular level using these imaging techniques helpful in understanding specific process at cellular level in living organisms. These specific processes includes protein-protein interaction, dynamic cell tracking throughout the entire organism, gene expression and drug action analysis. Thus

imaging techniques at molecular level are of great importance in understanding the psychology of living organisms and gives novel approaches for drug target identification and preclinical testing to improve drug discovery.

The Gamma based imaging is also useful for recognizing patterns of immune cells which has a significant importance for optimizing cell therapies for cancer. Positron emission tomography (PET) and single photon emission tomography (SPET) are the techniques which can be useful for in vivo imaging of molecular activities and composed of radiotracer-binding(modified membrane receptors such as dopamine D2 that bind labeled dopamine-analogs or enzyme like viral TK that trap gancyclovir like molecules). Besides this a number of other radiotracer are also developed to study cell proliferation, dopamine activity, metabolism and other related process [18].

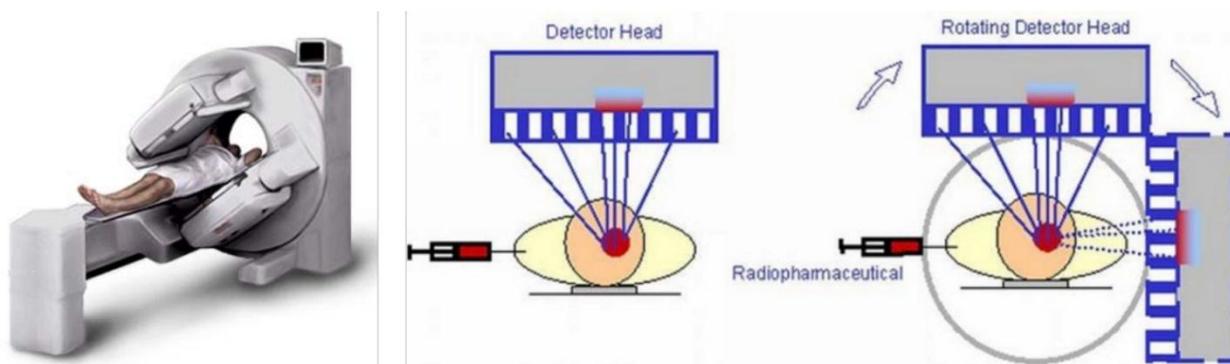


Figure 3.2 Gamma Camera

A gamma camera captures the images of gamma radiating radioisotopes. It consists of detectors, single or multiple crystal planes and an array of photo-detectors which are optically coupled. The gamma cameras are based on original design by Hal Anger. One of the applications of Hicam camera (a gamma camera based on Anger design) is in-vivo imaging of molecular processes and it is possible because of its ability to capture high resolution images

and immunity to high magnetic fields. The immunity allows this camera to work in parallel with MRI imaging. The working principle is simple that it counts gamma photons which are absorbed by the crystal in the camera. The crystal is usually large and in size and composed of sodium iodide with doping of thallium and it is used as scintillating material.

When the gamma radiation incidents on crystal it scintillates in response. A faint ash of light is produced when a gamma photon which is emitted from a radioisotope is absorbed in the crystal. This faint ash of light is detected by an array of photo-detectors placed on the other side of the crystal. The pattern of electrical signals at the photo-detector terminals are the reflections of single ash. The signals of photo detectors are then amplified and transferred to computer for reconstruction of a two dimensional image of relative spatial count density. The reconstructed image gives the information about the distribution and relative concentration of radioactive tracer elements present in the tissues and organs which are imaged.

The location of the interaction can be calculated by weighting the position of each photomultiplier tube by the strengths of its signal and in next step the mean position is calculated from the weighted positions. The energy of the gamma ray interaction and the total sum of voltages from each photomultiplier are proportional to each other. It discriminates between different isotopes as well as scattered and direct photons.

The spatial information of the gamma emissions from the imaging subject can be acquired by correlating detected photons with their point of origin. This correlation can be done by collimator. The collimator is made up of a thick sheet of lead typically of few centimeters and sheet has thousands of holes in it. Each hole limits the photons which can be detected by the crystal to a cone; the point of the cone is at the midline center of any given hole and extends from the collimator surface outward. The lead in the collimator does not totally attenuate incident gamma photons and inference may exist between some holes. This results in blurring within in the images.

The collimator greatly limits the sensitivity of the camera system by attenuating most of the incident photons. In order to detect sufficient scintillation events from a picture, large amounts of radiation must be present to provide sufficient exposure to camera system.

Single Photon Emission Computed Tomography (SPECT) imaging is used in nuclear cardiac stress testing and is done by using a gamma camera. In this imaging usually one or up-to three detectors or heads are slowly rotated around torso to get the image.

If gamma camera design allows to detect 'coincidences' then multi-headed gamma cameras can also be used in positron emission tomography scanning. The two key factors, first, the scintillator crystal has poor sensitivity for the high energy annihilation photons and second, smaller detector are results in inferior markedly of gamma camera to PET imaging with a purpose designed PET scanner. However, due to low cost gamma camera and its flexibility compared to dedicated PET scanner are the reasons which makes this technique for comercial use.

3.2 Hicam - System overview

Hicam is a design which is based on Silicon Drift Detectors. The design has compact gamma camera with high resolution. The design is technologically involves several improvements which has increased the resolution and drastically reduce the size of the system. The principle operation of the system, however, same as that of original Anger camera.

The compactness of system is achieved by using Silicon Drift Detectors (SDD) instead of photomultiplier tubes. the advantages of using SDD are mainly simplified polarized schemes and greater immunity to intense magnetic fields. A combine working of the system with magnetic resonance imaging (MRI) is possible due to its insensitivity to intense magnetic fields. it is an important feature as the combine use of both systems has an increased demand. This combine approach provides optimal visualization of anatomic and functional features. it is possible due to the combination of high anatomic resolution of MRI to the specificity and the flexibility of functional and molecular imaging that characterizes radionuclide imaging systems.

Unlike photomultiplier tubes, silicon detectors do not provide an intrinsic gain mechanism. The SDD design offers very low noise detectors. The noise power in state of the art SDD arrays is negligible as compared to the intrinsic uncertainty which is because of the emission into the scintillation crystal. Another feature is that silicon detectors provide higher quantum efficiency (QE) and the reason behind it is the presence of better refractive index matching between silicon detector and scintillation crystal as compared scintillation crystal and photocathode materials. This graded reflective index shift is achieved by applying optical grease at the time of assembling the crystal onto the photo-detectors and by apposite treatment of the silicon detector surface [19].

There are several features of Anger camera on which its attainable overall maximum resolution depends. These includes characteristics of photo-detectors, several geometric characteristics of the system such as crystal size and thickness, size and number of photo-detectors and the most important one is the collimator which is used for 'focusing' the image onto the crystal.

The following sections introduce various components of the gamma camera and some specific solutions adopted for Hicam gamma camera.

3.2.1 Scintillator

The gamma detector is based on an inorganic scintillator that produces a scintillation light when it absorbs ionizing radiation. Each event of interaction of a gamma ray with the crystal is a 'catastrophic' transfer of energy of the incident radiation to the electrons or nuclei of the constituent atoms. In fact uncharged electromagnetic radiation, as opposed to charged particulate radiation, does not continuously exert Coulomb interaction with surrounding atoms. In the range of energies of interest for medical imaging the interactions happens, for most of the events, by photoelectric absorption or Compton scattering.

In the photoelectric absorption process the gamma photon undergoes an interaction with an absorber atom in which the photon completely disappears, while an energetic photoelectron is ejected by the atom from one of its bound shells. The vacancy is filled through capture of a free electron from the medium or rearrangement of electrons from other shells of the atoms with

the generation of a characteristic X-Ray, which is typically reabsorbed by near atoms by photoelectric effect involving lower shells; the fast electron interacts with atoms in the crystal creating states of excitation and subsequent radioactive decay. The crystal is typically doped in order to create recombination centers that determine the spectrum of emission, so it is possible to engineer the emission spectrum of the scintillator material by controlling the crystal doping.

The interaction process of Compton scattering takes place between the incident gamma ray photon and an electron in the absorber, causing a deflection in the direction of the incident gamma radiation and a partial energy transfer to the electron. Because all angles of scattering are possible, the energy transferred to the electron is distributed from zero to a large portion of the gamma ray energy.

As a rule of thumb photoelectric absorption happens with a probability which depends on the fifth power of the atomic number of the absorber Z , while Compton scattering depends on the density of electrons, therefore it increases linearly with Z .

As we will see, in our application, photoelectric absorption has to be maximized, as events due to Compton scattering will be discarded, so the material of choice will have high atomic number.

If a scintillator coupled with a light detector is used for the purpose of determining the total energy of incident gamma radiation, then a large crystal has to be used in order for all the energy to be converted, in fact in case of small crystal part of the energy will quit the crystal because of Compton scattering.

If the spectrum of incident gamma radiation is of interest, then only events due to photoelectric absorption are significant.

In a configuration for two-dimensional position sensing, such as the Anger Camera, it is desirable to discard events due to Compton scattering as they present a low energy that makes them not distinguishable from events that have scattered before entering the camera. In order to discard these events that constitute noise, it is necessary to discard events that interact by Compton scattering. The radioisotope that is used as a gamma ray source for a typical

application in medical imaging emits radiation at very precise energy, or may have more than one peak of emission due to the atomic structure of the radioisotope. Gamma radiation may scatter inside the patient or other medium, causing radiation from an unknown source location to enter the collimator and scintillate, thus determining image noise, unless a threshold on the total collected charge is applied in order to exclude events caused by gamma rays that have scattered before entering the crystal. Gamma rays that interact by photoelectric absorption deposit all their energy, thus the peak in energy distribution is exactly at the energy of the gamma ray radiation, named photopeak; in order to exclude scattered radiation, a window around the photopeak selects events to be considered valid. This mode of operation excludes events caused by radiation that interacts by Compton scattering. A scintillator material with high atomic number Z is chosen in order to maximize the probability of interaction by photoelectric absorption.

Inorganic scintillators provide very high stopping power as they can achieve high atomic number. Crystal thickness of less than one centimeter can provide efficient collection of incoming gamma radiation.

Inorganic scintillators are based on insulating host crystals in which luminescent ions or complexes are imbedded. The main classes are activated scintillators, self-activated scintillators and core-valence luminescent scintillators. In activated scintillators the ionizing photon passes through the host crystal and produces an excited state near an activator atom that is present in low concentration. Examples of activated crystals are NaI:Tl, CsI:Na, CsI:Tl, CaCaF₂:Eu, and Lu₂SiO₅:Ce. In self-activated scintillators, the activator atoms are a major constituent of the crystal. The most common self-activated structures are Bi₄Ge₃O₁₂, CeF₃, and PbWO₄. In core-valence luminescence scintillators, as CsF, and RbCaF₃, the ionizing gamma radiation produces an hole in an upper core level of one atom and the vacancy is filled with an electron from the valence band of another atom to produce the visible photon.

The manufacturing of scintillators uses melt based methods such as Bridgman and Czochralski. These melt-based methods are suited for growth of large volume crystals. In the Czochralski process, the single crystal material is pulled out of the melt in which single-crystal seed is immersed and then slowly withdrawn. The Bridgman process uses multi-zone furnace in which

the scintillator materials are contained in an ampule and in contact with crystal seed. The compound melt is passed from higher to lower temperature zone to obtain the desired crystal volume.

Characteristics of a scintillator may be grouped into physical or luminescent. Among physical characteristics are detection efficiency, chemical stability, mechanical strength, physical form and cost. Luminescent characteristics of most concern are emission wavelength, light yield, energy resolution, proportionality, signal rise time and decay time, afterglow.

Light yield is the light output, in number of photons, per 1 MeV of absorbed ionizing radiation. The fundamental limit on the light output depends on the energy needed to generate an electron-hole pair assuming all the energy of the incident gamma quantum has been dissipated in the crystal. The energy to generate a single pair is usually a factor 2 or 3 greater than the band gap.

The radioactive process is responsible for the light yield of the scintillator; therefore the extent of the radioactive over the nonradioactive processes is a first measure of the scintillator performances for application in a Anger camera. In effect, the energy resolution and sensitivity of the imager increase with the light yield.

Proportionality also impacts the energy and spatial resolution. Proportionality measures the linearity of the relationship between the number of emitted photons and the absorbed ionizing radiation. A scintillator having perfect proportionality produces the same number of electron-hole pairs (within statistics); no matter how many Compton interactions occur before the final photoelectric absorption of an incident monoenergetic gamma ray. Non-proportionality (as a function of energy) in light yield can be one important reason for degradation in energy resolution of established scintillators such as NaI:Tl and CsI:Tl.

Signal Rise and Decay times account for the time required by the scintillator to respond to ionizing radiation; these properties represent the dominant factors determining the counting rate capability of the imaging system. The presence of delayed components in the response of the scintillator is termed afterglow and should be avoided in almost all the applications.

Virtually the only material that has been used for the past 40 years is NaI:Tl. NaI:Tl scintillation crystals are used in most standard applications for detection of γ -radiation because of their high light output (42 photons/keV) and the excellent match of the emission spectrum to the sensitivity of photomultiplier tubes (emission peak occurs at 415 nm). In addition, the primary decay time is quite short (0.23s). The energy resolution (measured for 662KeV gamma rays) is poor, being about 6% to 7%. The limited energy resolution in NaI:Tl is probably due to non-proportionality [23]. Over the range from 60 keV to 1 MeV, non-proportionality has been measured to be about 20%.

Energy resolution is the ability to distinguish the energy of the incoming radiation, which in turn allows for distinction between interactions by photoelectric absorption and Compton Scattering. Energy resolution is defined as the ratio between the width FWHM of the photopeak and its height.

The most desirable improvement for scintillators would be increased luminous efficiency and energy resolution. The improved accuracy in energy measurement would allow a more efficient suppression of Compton scatter background, which actually contributes to up 50% or more of the counts in a nuclear imager and whose not suppressed fraction is linearly proportional to the energy resolution.

In applications where solid state silicon detectors are used, CsI:Na and CsI:Tl (Cesium Iodide, Sodium doped or Thallium doped) crystals are preferred because of the best match of their emission spectrum to the sensitivity of these detectors, with respect to NaI:Tl. The emission peaks occur at 420nm and 550 nm, respectively. For coupling with SDDs, CsI crystals doped with Tl should be preferred since the detection efficiency of the SDDs is maximum in the range 440-700 nm, therefore is best matched with the peak emission wavelength of CsI:Tl.

The light yield of cesium iodide-based materials is also high, about 42-65 photons/keV. The drawback is their relatively high decay constant, usually in the range 0.6 3s, a severe limitation to both the system energy resolution and counting rate capability of gamma imagers implementing high performance photodetectors. Moreover, the energy resolution is still quite low, with values from 5.9% up to 6.6% for gamma rays of energy equal to 662KeV.

Several recent cerium doped inorganic scintillators ($\text{RbGd}_2\text{Br}_7\text{:Ce}$, $\text{LaCl}_3\text{:Ce}$, $\text{LaBr}_3\text{:Ce}$, and $\text{LuI}_3\text{:Ce}$) possess scintillation efficiency and energy resolution that approach theoretical limits, as well as fast decay times and moderately high densities and effective atomic numbers. Because of their high light output and excellent energy resolution, $\text{LaBr}_3\text{:Ce}$ and $\text{LaCl}_3\text{:Ce}$ have the potential to replace NaI:Tl as the materials of choice for gamma imagers. $\text{LaBr}_3\text{:Ce}$ is probably the most promising material.

$\text{LaBr}_3\text{:Ce}$ with 0.5% Ce^{3+} has a light output of 61,000 photons/MeV, 50% higher than NaI:Tl and amongst the highest values for inorganic scintillators. The light output depends on the Ce^{3+} concentration, being nearly 50,000 photons/MeV for 2% Ce^{3+} concentration and within 10% of this value for $\text{LaBr}_3\text{:Ce}$ doped with higher concentrations.

Interestingly enough, future research on high light yield Ce^{3+} based scintillators, obtainable by reduction of the band gap E_{gap} , imply luminescence in the green or red part of the spectrum where PMTs have less detection efficiency and solid state detectors (as SDDs) have higher.

$\text{LaBr}_3\text{:Ce}$ achieves less than 3% FWHM energy resolution for 662 keV gamma rays, a resolution that is twice as good as that of NaI:Tl and has not been achieved with any of the established inorganic scintillators. For 511 keV energies (from ^{22}Na) and 122 keV energies (^{57}Co), the energy resolution is of 3.6% and 6.8%, respectively, at room temperature.

Over the energy range 60-1275 keV, the non-proportionality in light yield is about 6%, which is substantially better than that of many established scintillators. For example, over the same energy range, the non-proportionality is about 35% for LSO and about 20% for NaI:Tl and CsI:Tl . The higher proportionality of $\text{LaBr}_3\text{:Ce}$ is one of the important reasons (in conjunction with the high light output) behind the high energy resolution of this scintillator.

The primary decay time is shorter than 25 ns (over 90% of the emitted photons are emitted with this decay lifetime). This longest decay time value is usually obtained for Ce^{3+} concentration less than 1%. As the Ce concentration increases the principal decay time constant

decreases from 25 ns for 0:5% to about 17 ns for crystals with 5% and more of Ce. Moreover, there is also a significant decrease in rise time constants indicating an efficient and fast energy transfer as well as high charge collection efficiency by Cerium ions in LaBr₃. The average rise time measured for samples doped with 0:5% Ce is as long as 9 ns, whereas for a sample doped with 20% Ce it shortens to less than 0.2 ns.

For the Hicam camera both CsI:Tl and LaBr₃ are taken into consideration, given the higher performance of LaBr₃ and lower cost of CsI:Tl.

3.2.2 Collimator

Given a spatial distribution of radioisotope atoms that emit gamma radiation in all directions, then if the radiation that is absorbed by a planar crystal is observed and it is possible to discriminate the point of interaction of each of the absorbed gamma photons with the crystal, then it is possible to determine the spatial distribution of radioisotope only if some additional constraint is introduced. In fact if all gamma photons, from each of the point sources, that cross the crystal plane determine scintillation, then it is not possible to determine the position in 3D space of the sources.

While in optical imaging, a converging lens allows for the radiation emitted from a point source in any direction that crosses the lens entrance to be focused of the same point on a focal plane, thus providing a constraint that determines a biunivocal relation between points on a given plane in 3D space and a focal plane, while adding a number of replicas of the same image on the phocal plane, with high energy radiation the construction of compact lenses is not possible, thus different mechanisms have been devised in order to determine the coordinates in 3D space of a source. These techniques, based on the use of scintillators, allow for imaging of gamma rays, but still they limit the detectivity because most (> 99%) of the radiation is dissipated.

A collimator, mounted on the crystal, is a piece of thick (order of a few centimeters), high Z material that introduces a constraint to the gamma photons that irradiate the scintillator by

absorbing part of the incoming gamma radiation that comes from unwanted direction and allowing the remaining to interact with the scintillator crystal. Typical collimator configurations are parallel-hole, converging and pin-hole.

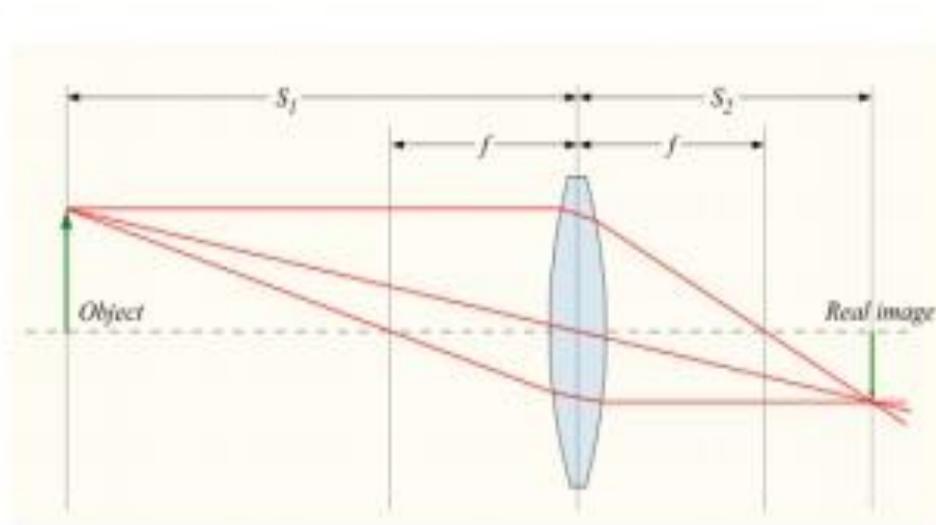


Figure 3.3 Focusing lens

The parallel-hole collimator presents a big number of small size holes perpendicular to the crystal plane. Only gamma photons coming from a direction that is perpendicular to the crystal plane do interact with the crystal, while the others are absorbed by the collimator's septa 2.2.2, thus the rate of events occurring in one small area of the crystal ideally correspond to the total number of point sources that are located in a line perpendicular to the detector plane. The image that can be reconstructed is a projection of the 3D distribution of point sources onto the detector plane.

In a Converging collimator the holes are not parallel but focused. The focal point is normally located in the center of the field of view (FOV). When the Converging collimator is flipped over you get a Diverging collimator, generally used to enlarge the field of view, for example used with portable cameras with a small crystal.[19]

Pinhole collimators have a single hole; a magnified image is obtained of the 3D distribution of point sources as long as the distribution has a limited depth, in fact images from different depths undergo integration onto the scintillator and they appear with different scale factors. Pin-hole collimator offers the possibility to magnify the image, but it generally provides reduced sensitivity as all the light enters from a small hole.

Ideally, the collimator allows a gamma photon to reach the detector only if it has a precise propagation trajectory. In reality, effects of undesired photon losses or transmissions affect the behaviour of the collimator. Undesired photon losses are given by photons having the desired direction and whose propagation path ends on the cross sectional area of the septa. Undesired transmissions are due to three possible mechanisms. A first component accounting for undesired transmissions is due to the presence of photons whose direction is only slightly deviating from the selected direction so that their path does not encounter the collimator septa. A second component, termed the penetration component, arises when photons with undesired trajectory succeed in penetrating the thickness of one or more septa. An additional fraction, the scattered component, derives from photons that have been diffused during their path toward the detection system and have consequently acquired a direction parallel to the septal lengths by chance.

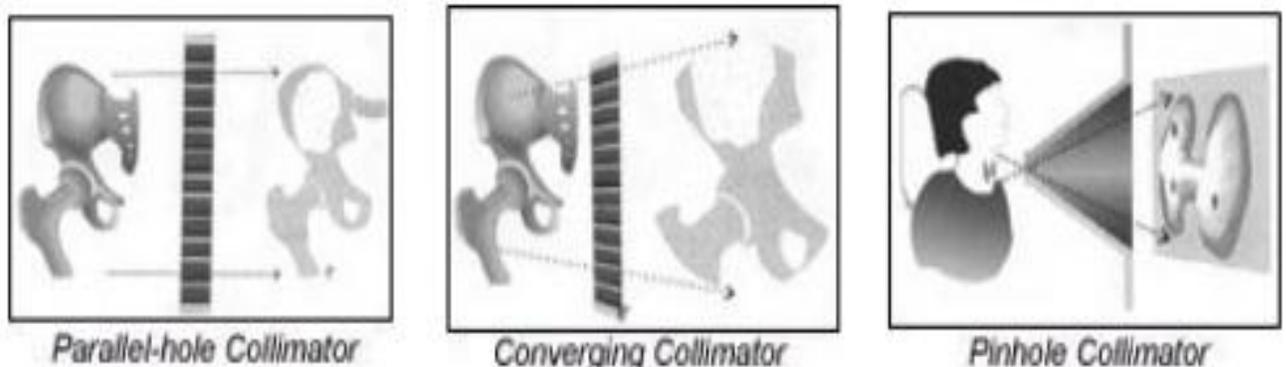


Figure 3.4 Collimators

Collimators are made of materials with high atomic number Z . The high value of Z enhances the probability of photoelectric absorption of gamma photons in their path through the collimating septa, thus decreasing the penetration component for a given value of septal thickness and length.

Manufacturing processes then impose severe constraints on the selection of the material, e.g. in terms of malleability and ruggedness. The materials of common choice are lead and tungsten, whose atomic number is of 82 and 74 respectively. The design of the collimator concerns the choice of septal geometry in terms of shape of the cell (i.e. square or hexagonal holes), length of its sides, height, septal thickness and angular divergence.

Conventional fabrication methods are pressing of thin lead foils and micro-casting. Lead foil collimators can have septal thickness as thin as 100 μ m. However, foil pressing often causes defects in septal angular alignment or deformations, resulting in reduced effective spatial resolution and uniformity. Micro-cast collimators cannot have septa thinner than 0.63 μ m. On the counterpart, they have uniform septa thickness, good septa angular alignment and a stronger structure than foil collimators. In clinical, routine systems, the collimator is usually made from cast lead.[19]

Innovative fabrication methods are chemical etching and lithography. Stacked photochemically-etched collimators made in tungsten can have septal thickness down to 100 μ m, with uniform septa thickness and good angular alignment.



Figure 3.5 Collimator design

In clinical routine imaging systems the collimator is generally a parallel hole collimator with square or hexagonal cross section holes arranged in a honeycomb fashion. Each hole has a diameter, a length and is separated from its neighbours by the septal thickness, typically of 0.2 mm.

The energy range for which the collimator is appropriate is determined by the choice in terms of septal thickness. Applications where high energy photons must be detected and imaged require higher thicknesses with respect to applications based on low energy photons to obtain the same penetration component of undesired transmission. Undesired transmissions should be avoided since they affect negatively the obtainable spatial resolution.

The ideal collimator for a specific energy range should have both high spatial resolution and high efficiency. In general, the spatial resolution may be improved by means of a higher septal thickness and length, but these choices come at the expense of a reduced efficiency because of the higher photon losses.

Resolution scales linearly with the ratio between hole aperture and septal length, usually specified as the aspect ratio of the collimator and roughly linearly with the distance d from the collimator.

The effect of the penetration component on the system spatial resolution depends on the performances of the detector head and electronics, in particular on the intrinsic energy resolution. The reason is that the penetration fraction of the incoming photons is a gamma radiation of reduced energy, since it is attenuated through its path in the septal thickness. If the intrinsic energy resolution of the detector head and electronics was ideally zero, the system would recognize this degraded component of the signal as a scattered radiation, and would not consider it for the formation of the final image. In reality, the detector head and electronics have a non-zero intrinsic energy resolution and may include the misleading information represented by the photons with degraded energy as a component of the final image, thus degrading the spatial resolution of the system itself.

Usually, the collimator design aims at making the contribution from the penetration fraction negligible with respect to the other components. As a rule of thumb, an overall energy resolution of 10% may be estimated for the final system.

Neglecting the penetration factor, arbitrarily high spatial resolution can be achieved by reducing the effective aspect ratio. As already mentioned, this choice is usually impractical because it tradeoffs severely with the obtainable sensitivity (or, detection efficiency) of the imager.

To obtain the system detection efficiency, the intrinsic efficiency of the imager must be corrected for the collimator transmission fraction, defined as the ratio between the number of photons transmitted by the collimator and the number of photons emitted by a point source at a specified distance and position from the collimator itself.

The design of the collimator is thus a challenging task if the aim is to obtain an application-specific high performance system.

3.2.3 Silicon Drift Detector (SDD)

The traditional photodetector of choice for a gamma camera is a photomultiplier tube because it provides an intrinsic gain mechanism that reduces the influence of noise from the electronics in the next stages. However silicon devices have recently shown to be competitive with photomultiplier tubes as they can be designed for very low noise, while coupled to low noise amplification.

Silicon detectors offer several advantages over photomultiplier tubes: compactness, ruggedness, increased immunity to intense magnetic fields, which is an important feature when a gamma camera is to be coupled with magnetic resonance imaging, high quantum efficiency when coupled with scintillator crystals, simplified polarization schemes. The gain mechanism offered by photomultiplier tubes is associated with statistical fluctuation that constitutes an additional noise source, also photomultiplier tubes show strong dependence of their characteristics from temperature and polarization shifts.

The basic form of a Silicon Drift Detector was proposed by E. Gatti and P. Rehak in 1983.[23] It consists of a volume of n-type silicon depleted by a n+ substrate contact reverse biased with respect to rectifying p+ junctions covering both surfaces of the structure. The radiation entrance side is the non-structured p+ junction on back side. The p+ junctions on top side are segmented and biased in order to establish an electric field in the detector volume with a component parallel to the surface. An integrated voltage divider is used to bias the segmented p+ junctions by means of two connections for the first ring and the last junction. The divider is implemented by a sequence of p-channel enhancement MOSFETs operating in the linear region. The principle of operation is represented in Figure 3.7.

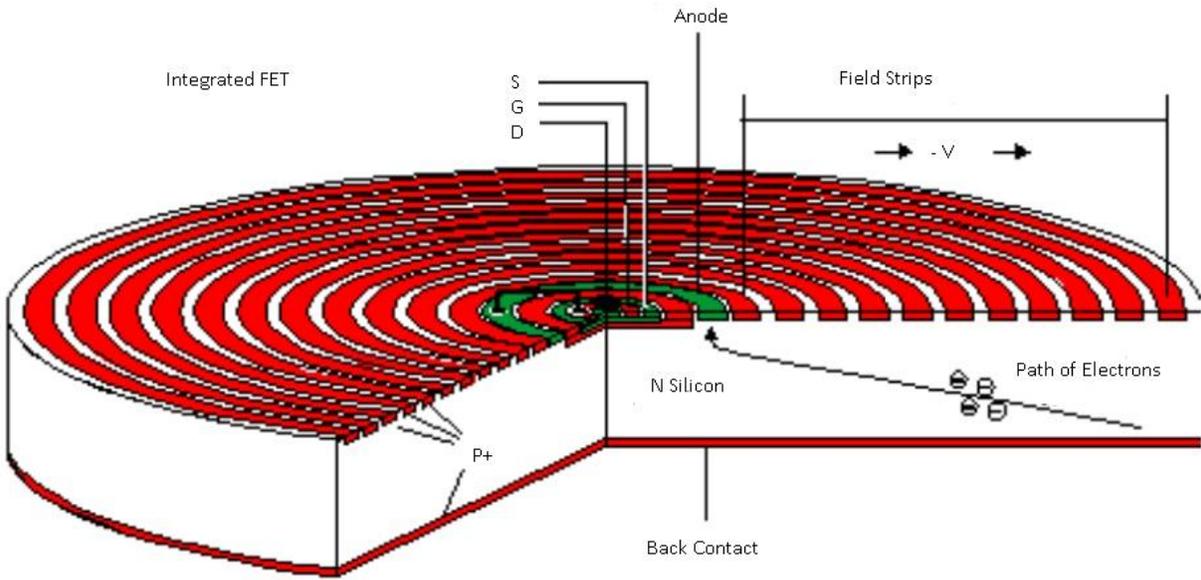


Figure 3.6 Circular SDD

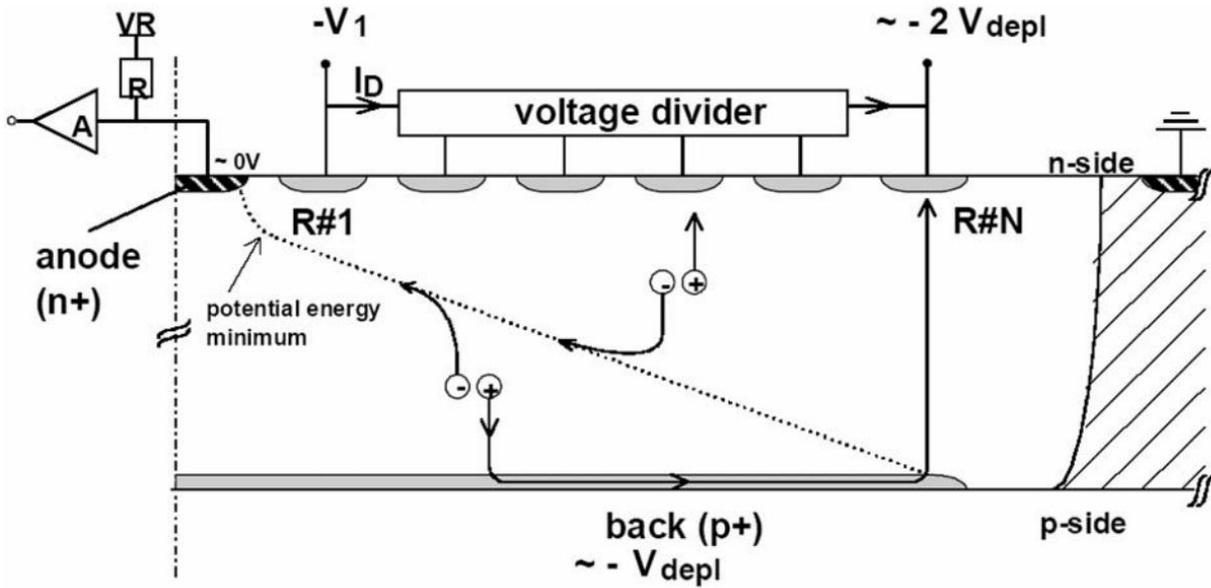


Figure 3.7 SDD basic operation

Holes created inside the depleted bulk of the detector (by the absorption of ionizing radiation or by thermal generation) are collected by the bulk electrode. The holes then flow from the back electrode towards the more negative ring of the n-side by means of the reach-through effect. Electrons released within the depleted volume together with the holes are driven by the electric field towards the collecting n+ contact, acting as an anode and connected to the front-end amplifier.

A drawing of the potential distribution in the detector volume is reported in 3.7 to illustrate the paths of holes and electrons from their generation up to their final collection.

The anode capacitance has a small value, thus reducing the electronics noise component due to the integrated leakage current of the device with respect to a photodetector having the same active surface area. The output capacitance has a value of 100 fF in a standard device.

The parasitic between the device and the electronics is also minimized by integration of a n-channel JFET on the detector chip, very close to the collecting anode. The gate of the SDD is connected to the anode of the detector by a thin metal strip. The signal charge and sensor leakage current accumulate on its gate capacitance, in parallel to the anode capacitance. The JFET transistor provides a voltage signal to the outer circuits at its source and is biased by means of an external terminal and a current supply. The JFET leakage of the reverse biased gate-drain junction discharges continuously the charge accumulated at the paralleled gate and anode. Moreover, the integration of the JFET minimizes the cross talk and simplifies the readout of the multi-element detector signals with respect to the solution based on external transistors.

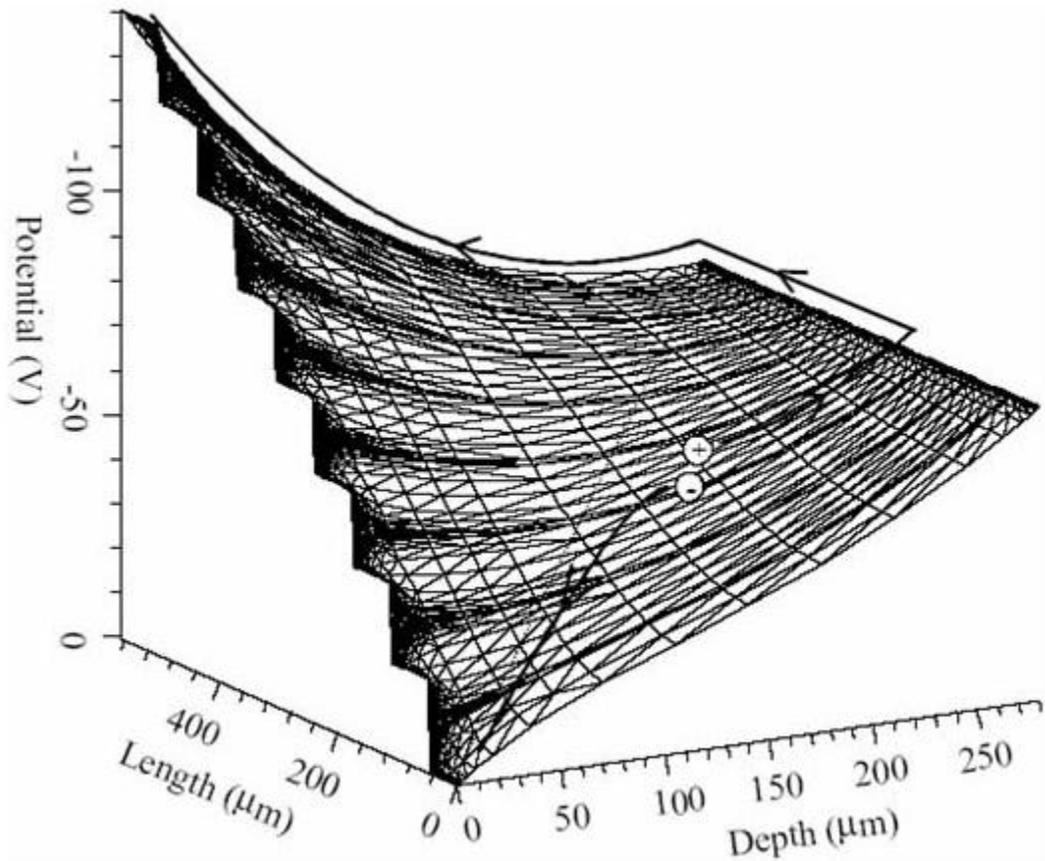


Figure 3.8 Potential distribution in the SDD volume - anode at (0,0)

With respect to an APD, the SDD offers greater uniformity of performances and lower photoelectrons statistic contribution, which in the APD is worsened from pure Poisson statistics by the noise factor. The SDD noise characteristics are usually specified in terms of Equivalent Noise Charge (ENC). The value of ENC may be deduced from the energy resolution measured in a spectroscopic test at known x-ray emission energy.

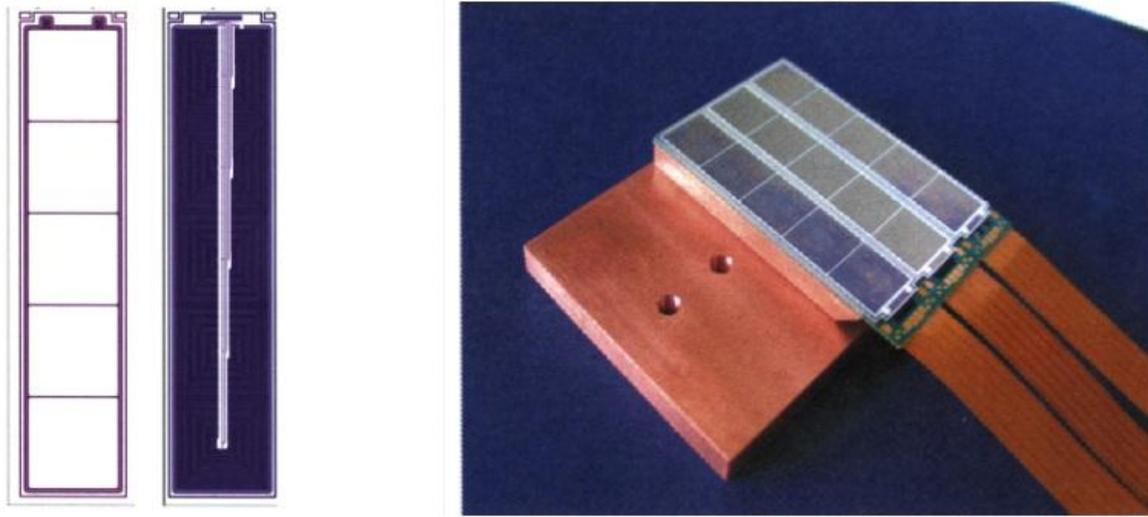


Figure 3.9 Hicam SDD array

Typically the optimum value of ENC is achieved at short shaping time (less than 1s). The small value of optimum shaping time makes the SDDs ideal for high counting rate applications. Compared to a PIN operated under the same conditions of temperature and electronics noise, the SDD has higher energy resolution and may be operated at higher count rates.

SDDs are the photodetectors of choice for the Hicam project, aimed at high position-resolution, system compactness and insensitivity to intense magnetic fields for compatibility with MRI imaging systems. Hicam camera has a matrix of 100 square SDD detectors of the size of 1cm² each.

The camera has a modular structure, based on monolithic arrays of 5 SDDs shown in figure 3.9, as a trade-off between yield in the fabrication process and simplicity of mounting and interconnection. State of the art technology for the fabrication of SDDs produces detectors with very low leakage current level, thus the camera is operated with moderate cooling by means of a single stage Peltier element.

3.2.4 Readout electronics

At the output of the array of photodetectors implemented in the Anger camera, the information on the energy of a single detected gamma photon is in the form of several electrical pulses, each one delivered by an element of the array, whose amplitude is related to the number of visible photons emitted by the scintillator over the active area of the same element. A suitable electronics is necessary to process the signals in order to extract the information about gamma ray energy and interaction coordinates. While reconstruction of the coordinates is implemented on a digital processing unit 2.2.6, the signals from the photodetectors need proper amplification and filtering, before being sampled and digitized by the digital acquisition system. Additional circuitry is needed for detection of events. An application specific integrated circuit (ASIC) has been developed at Department of Electronic Engineering of Politecnico di Milano, that includes the electronics for processing of 25 photodetectors.

Each processing channel implemented into the ASIC receives as its input the charge pulse delivered by a single photodetector. The time length of the pulse is almost equal to the decay time of the scintillator and its amplitude is related to the energy released by the scintillation photons interacting within the photodetector material. Anyway, non idealities of the system (in the form of delivered charge having no relationship with the photon energy) affect the possibility to determine the desired information from the electrical pulse. The electronics processes the signal from the photodetector in order to reduce the effects of noise and to provide at its output a measure of pulse amplitude, that is, of light intensity.

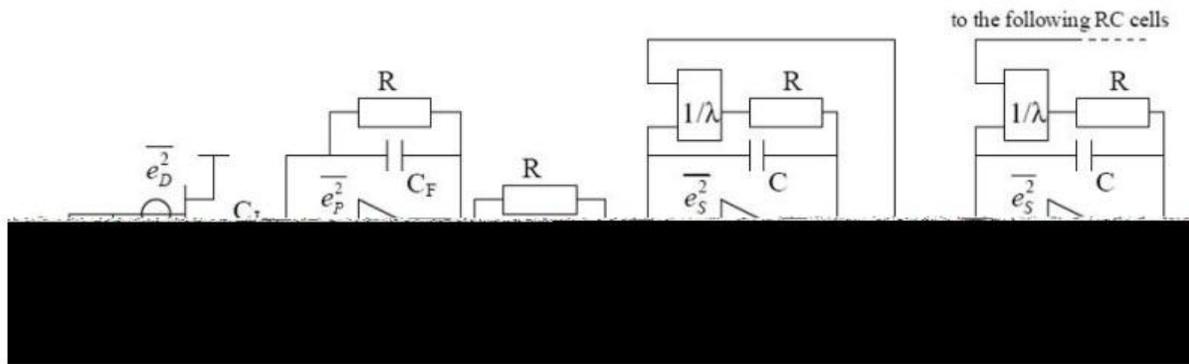


Figure 3.10 Schematic of a processing channel in Hicam ASIC

Each processing channel implemented into the ASIC receives as its input the charge pulse delivered by a single photodetector. The time length of the pulse is almost equal to the decay time of the scintillator and its amplitude is related to the energy released by the scintillation photons interacting within the photodetector material. Anyway, non idealities of the system (in the form of delivered charge having no relationship with the photon energy) affect the possibility to determine the desired information from the electrical pulse. The electronics processes the signal from the photodetector in order to reduce the effects of noise and to provide at its output a measure of pulse amplitude, that is, of light intensity.

At the input, the analog section of the circuit integrates also a current source that is necessary in order to operate the integrated JFET in a source follower configuration. The first stage of the processing channel is a low-noise voltage preamplifier. The pulses produced by the SDDs require amplification before they can be shaped, analyzed and measured. The preamplifier is based on the conventional cascode configuration and is composed of an ac coupling capacitor C_j followed by a charge amplifier with capacitive feedback C_f .

Processing the signals produced by SDDs faces with the problems of pulse shaping to optimize the signal to noise ratio (SNR) within limits imposed by statistics and counting rate. The shaper amplifier is a 6th order shaper with real poles, composed by a sequence of 6 identical RC integrators. Although a complex-pole shaping configuration would have offered a slightly better

similarity to the ideal gaussian waveform, the full real-pole shaper has been chosen since it offers the advantage of the modularity of the time-constant circuits. Moreover, the theoretically evaluated noise performances are almost identical for both configurations.

The amplifier has the possibility to select 1 out of 4 values of shaping time , in order to perform the best noise performances with scintillators having different values of decay time. The selectable peaking times have values of 1:7 μ s, 2 μ s, 4 μ s, 6 μ s. The gain of the shaper is also selectable from 2 possible values.

At the output of each shaping amplifier the signals undergo a threshold that allows, with some additional logic implemented into the ASIC, to detect an interaction event. When an interaction is detected, a peak-stretcher is activated in order to 'sample' the voltages from all the shaping amplifiers.

The ASIC also implements basic logic for communication with an external digital controller, that is implemented into a FPGA on the data acquisition board. The external controller is acknowledged about an interaction having taken place and may obtain the amplitudes from all the processing channels

through only one wire. Infact it clocks a shift register, built into the ASIC, that selects one channel at the time. So, when the FPGA is acknowledged about the interaction having taken place, it shifts the first channel to the analog output of the ASIC, it triggers the ADC (that is on the digital acquisition board) and when the conversion is finished it goes and shift the next channel to the analog output.

There are 4 ASIC for the 100 photodetectors of the Hicam camera; each is embedded on a small PCB. The 4 PCBs are mounted in close proximity to the photodetector array and they all connect to a mother board that includes additional logic for handling event acknowledgements from the the four ASICS and finally connects to the data acquisition system.

3.2.5 Data Acquisition system

Signals from the ASIC are fed to the data acquisition system that samples the analog signals, amplified and filtered by ASIC, performs basic processing on the digital data and finally sends event data to the processing unit through Ethernet interface.

The core of the data acquisition system is a FPGA that generates all the control signals for the ASIC. The FPGA is coupled with a 32 bit microcontroller (Atmel AVR-32 AP7000) that runs embedded Linux and handles the network interface to the processing unit and control of the FPGA. Signals from the photodetectors undergo amplification and filtering in the ASIC and are subsequently multiplexed in order to reduce pin count. The FPGA generates control signals for multiplexing of event data signals

that is then sampled by means of a 14 bit pipe-line ADC (Analog Device AD9240), that is also directly controlled by the FPGA device.

The interface between data acquisition system and the analog board where the ASIC is mounted is based on low voltage differential signaling (LVDS) in order to reduce electronic noise due to digital switching. The interface is based on Texas Instruments SN65LVDS391 and SN65LVDS390.

3.2.6 Processing Unit

In the original design of Anger camera the electrical signals determined, for an event of interaction, by each photomultiplier tube, were processed by an analog electric network that would apply the centroid method by coupling each signal to X and Y lines by a charge division process. The X and Y lines were then fed to an analog persistence oscilloscope for visualization. Most designs still apply the same mechanism but digitize the X and Y signals and the integral signal obtained by sum of all the signals from the photomultiplier tubes and compute the image reconstruction in the digital domain. More complex gamma camera designs directly digitize the

electrical signals produced by the photodetectors, after proper amplification and filtering, and compute the coordinates of interaction in the digital domain.

The Hicam camera is based on this last solution as it provides more flexibility in the implementation of the reconstruction algorithm, in fact as all the components of the camera are optimized for high resolution, it is crucial to apply an optimum reconstruction method that does not degrade image quality. As reconstruction of coordinates of interaction requires much more computing than simple image integration, it is sometimes performed on-line; in such a configuration the processing unit is composed of a data recorder and a computer for subsequent elaboration.

Chapter 4 Reconstruction Program and Algorithm implemented

4.1 Data and zone subdivision

The matrix that will be used in the training will be subdivided accordingly to three parameters:

- Train Data: This data will be used to effectively train the network and back propagate the error. This batch should be at least the 70% of the total data, to avoid to miss some cases during the training.
- Validation: This data will be used to decide when to stop the training. The evaluation on the reaching of the minimum errors will be made on them.
- Test: This data will be used to test the network. The final evaluation in terms of FWHM of the error (seen as the difference between the expected and obtained values) will be evaluated on this parameter.

The matrix may also be divided in zones, to improve the speed and precision of the networks. Each zone should be however overlapping with the neighbors to avoid border effects between each zone.

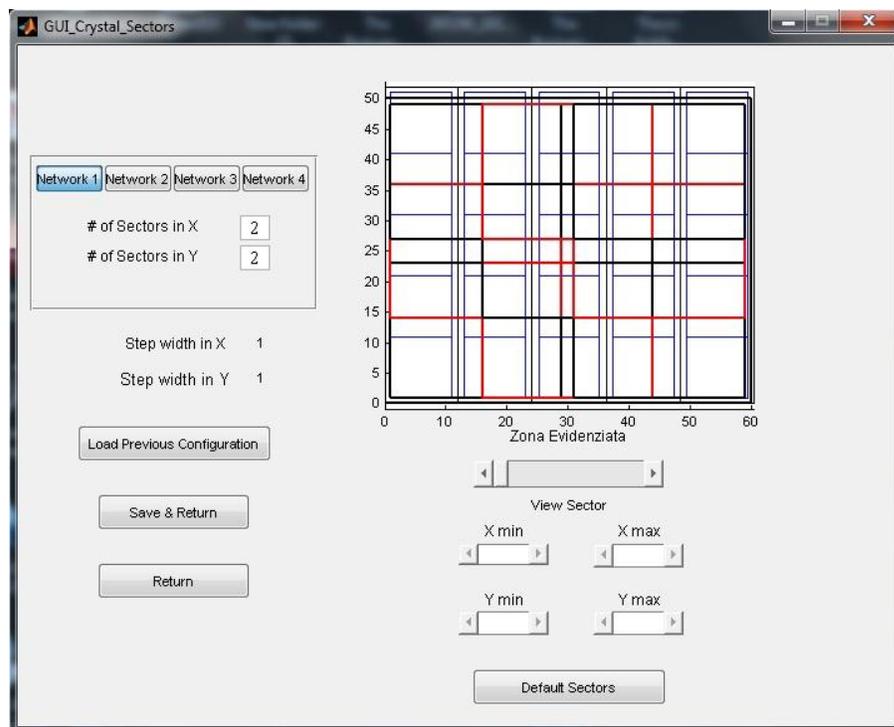


Figure 4.1 All the zones are shown overlapping one each other

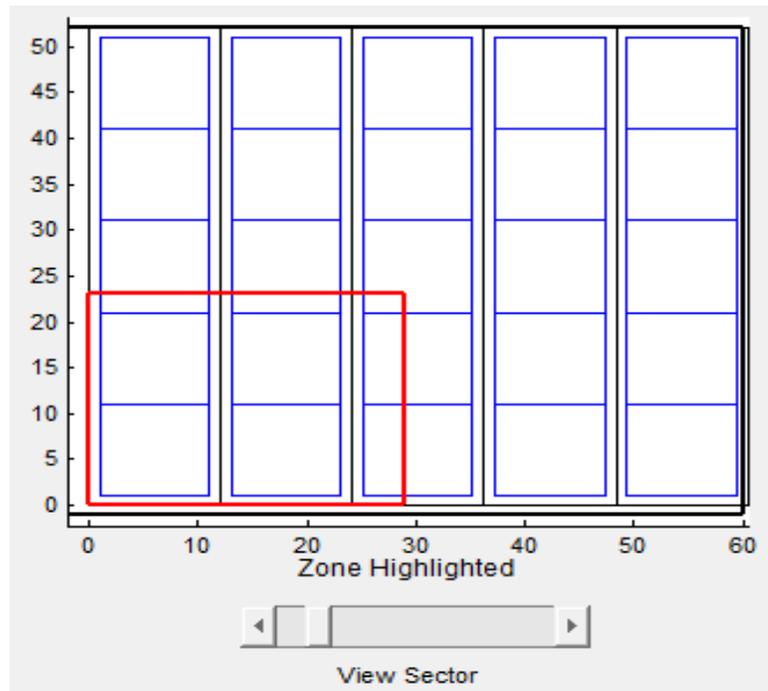


Figure 4.2 The first zone is highlighted

4.2 Main NNs GUI

This is the main user interface that we created to train various types of networks. It utilize the neural network tool box to implement different neural network algorithms.

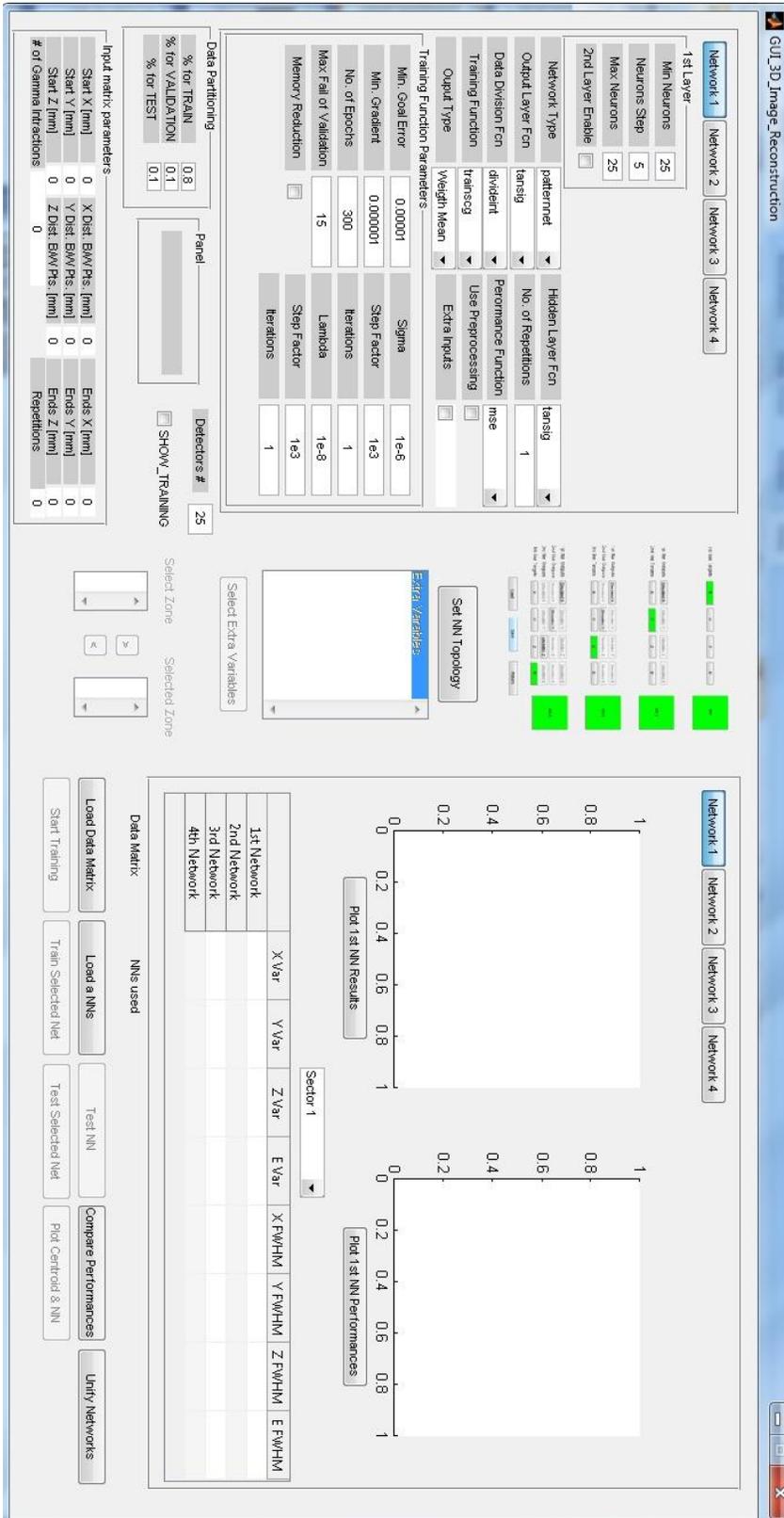


Figure 4.3 Main GUI

It is possible to create concatenate networks, in which the outputs of a network are given as inputs to the following ones.

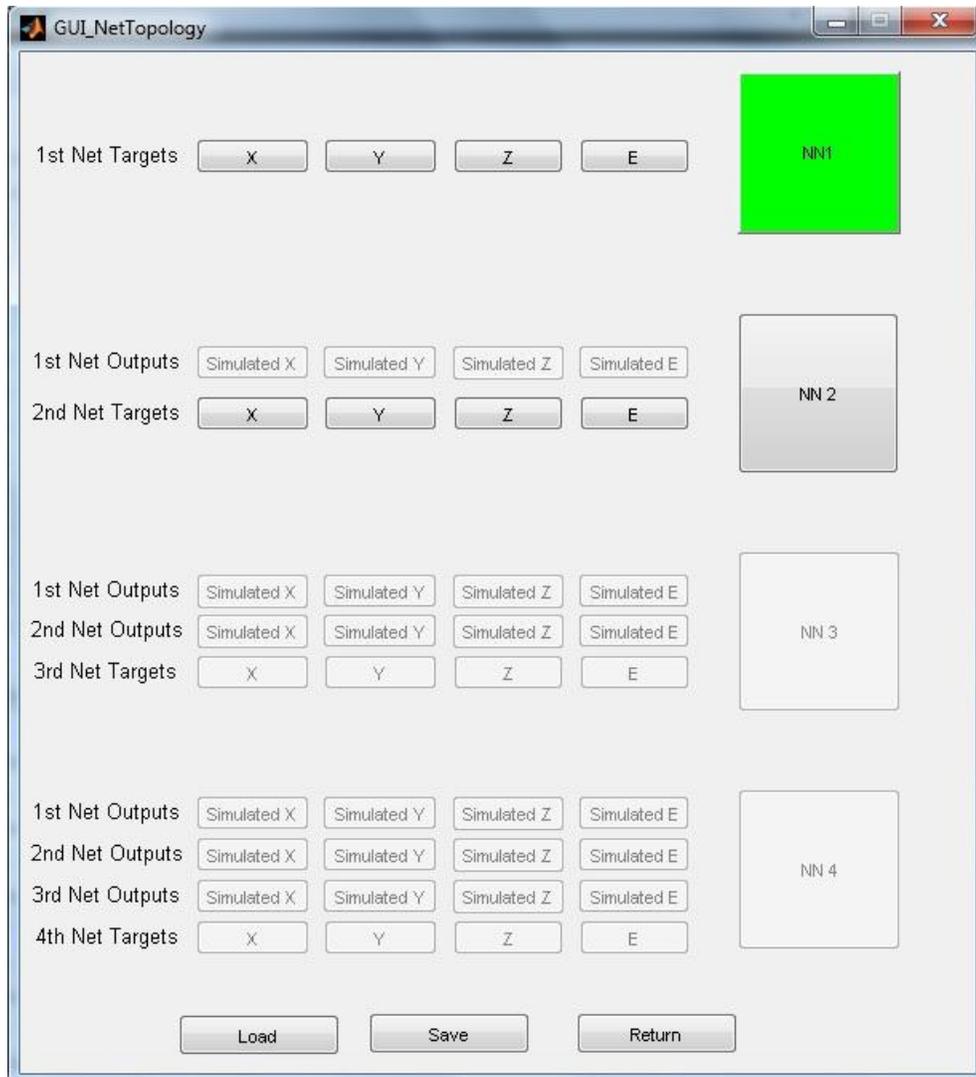


Figure 4.4 Network Topology GUI

For each network created (in the case in figure 4.5 concatenate networks were created) is possible to describe the networks structure: number of layers and how many hidden neurons

are to be simulated inside each layer. It is possible to set up an interval, to try different network configurations.

The screenshot shows a software interface for configuring a neural network. At the top, there are four tabs labeled 'Network 1', 'Network 2', 'Network 3', and 'Network 4', with 'Network 1' selected. Below the tabs are three columns representing the '1st Layer', '2nd Layer', and '3rd Layer'. Each layer column contains input fields for 'Min Neurons', 'Neurons Step', and 'Max Neurons', along with a checked checkbox for 'Layer Enable'. The 1st Layer has Min Neurons: 25, Neurons Step: 5, Max Neurons: 25, and 2nd Layer Enable checked. The 2nd Layer has Min Neurons: 15, Neurons Step: 5, Max Neurons: 20, and 3rd Layer Enable checked. The 3rd Layer has Min Neurons: 10, Neurons Step: 2, and Max Neurons: 14. Below the layer settings are several dropdown menus and checkboxes: 'Network Type' (patternnet), 'Hidden Layer Fcn' (tansig), 'Output Layer Fcn' (tansig), 'Data Division Fcn' (divideint), 'Training Function' (trainscg), 'Output Type' (Weight Mean), 'No. of Repetitions' (1), 'Performance Function' (mse), 'Use Preprocessing' (unchecked), and 'Extra Inputs' (unchecked). At the bottom, there is a 'Training Function Parameters' section with a grid of input fields: Min. Goal Error (0.00001), Min. Gradient (0.000001), No. of Epochs (300), Max Fail of Validation (15), Memory Reduction (unchecked), Sigma (1e-6), Step Factor (1e3), Iterations (1), Lambda (1e-8), Step Factor (1e3), and Iterations (1).

Figure 4.5 Main GUI . Network Selection

The typology of network that can be selected is “pattern net” or “feed forward net” and consequently the various parameters that define the network (use preprocessing, performance and layers functions, number of repetitions...) can be selected

The parameters on the left are to be set to determine when the network will finish its training. If during the training the minimum goal error or gradient or the maximum number of epochs are reached, then the training will be considered complete. It will be finished also if the internal nodes will not be changed after a certain number (defined Max Fail of Validation) of iterations

The parameters on the right indicate the starting values for the training variables. These will be automatically changed during the network training, but starting from a certain value or another one, may cause the network to converge fast or slowly.

The screenshot shows a software interface with the following components:

- Data Partitioning:** Three input fields for percentages:

% for TRAIN	0.8
% for VALIDATION	0.1
% for TEST	0.1
- Panel:** A large empty rectangular box.
- Detectors #:** A text box containing the value 25.
- SHOW_TRAINING:** A checkbox that is currently unchecked.
- Input matrix parameters:** A table with the following data:

Start X [mm]	1	X Dist. B/W Pts. [mm]	1	Ends X [mm]	59
Start Y [mm]	1	Y Dist. B/W Pts. [mm]	1	Ends Y [mm]	49
Start Z [mm]	1	Z Dist. B/W Pts. [mm]	1	Ends Z [mm]	9
# of Gamma Intractions	260190			Repetitions	10

Figure 4.6 Main GUI . Data partitioning and input matrix dimensions

It is possible to choose how to divide the input data between the data which will be used for TRAINING, VALIDATION and TEST. The figure 4.6 shows that 80% of data of input matrix is used for training, 10 % for validation and remaining for the test of neural network. The **INPUT MATRIX PARAMETERS** depict the start of data, the step size and the end of data for X, Y and Z co-ordinates of matrix data. These are the parameters obtained from the input Matrix that describe its dimensions.

Figure 4.7 provide the utility to select the training input parameters; it is possible to add also other data, like the centroid for X and Y co-ordinates



Figure 4.7 Main GUI – Button to select extra variables as input as NN

If the matrix is divided in zones, to aim to faster and more precise networks, it is possible to train separately each zone, to improve the results of the worst ones as shown in figure 4.8

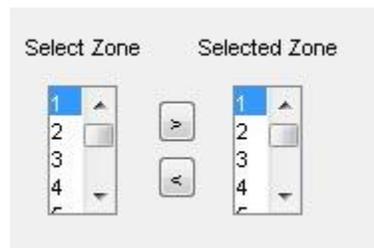


Figure 4.8 Main GUI – Provide to select different sectors

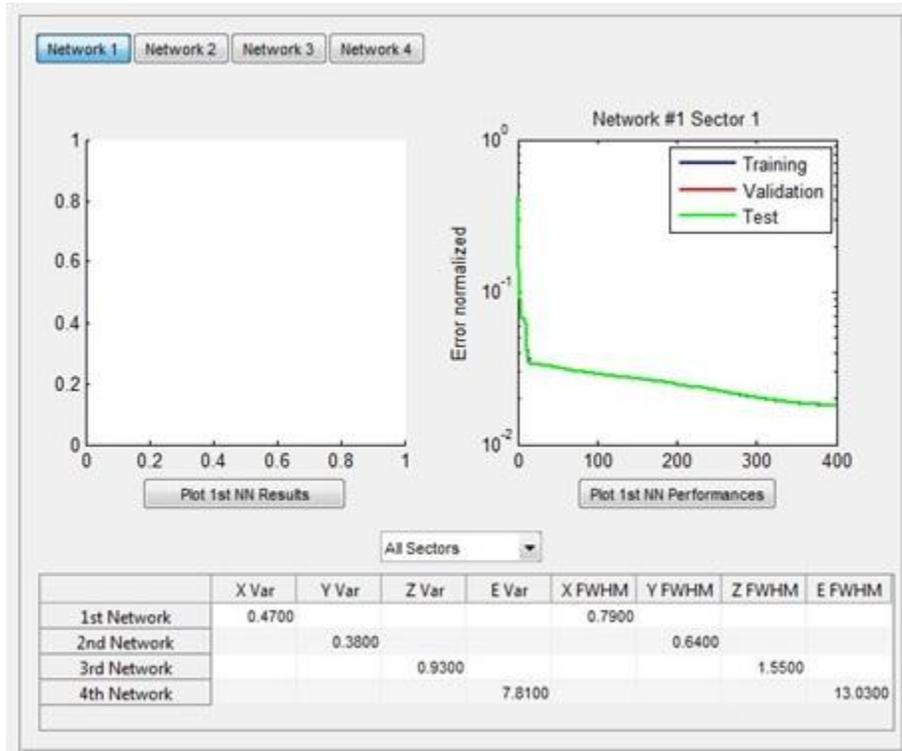


Figure 4.9 Main GUI – Display the training results

Here are shown the results of the training: the error expressed as the difference between the obtained points and the desired outputs is shown on the left, while on the right it is shown the training slope. If this last one is not flat at the end, then it would be possible to obtain a better result in the training simply augmenting the number of epochs in the “Training Function Parameters” section



Figure 4.10 Main GUI – Buttons to load input matrix, trained NN and test NN etc.

Load Data Matrix: Load the matrix that will be used for the training or for the test of the network

Start Training: Start the training in the zones selected with the set parameters and the loaded matrix

Load a NNs: Load a previously created NN to be tested with a input matrix

Test NN: Test the NN created or loaded with a user defined input matrix

Compare Performances: Select two networks previously created and compare their total Var and FWHM obtained

Plot Centroid & NN: Open another program for the testing of the results

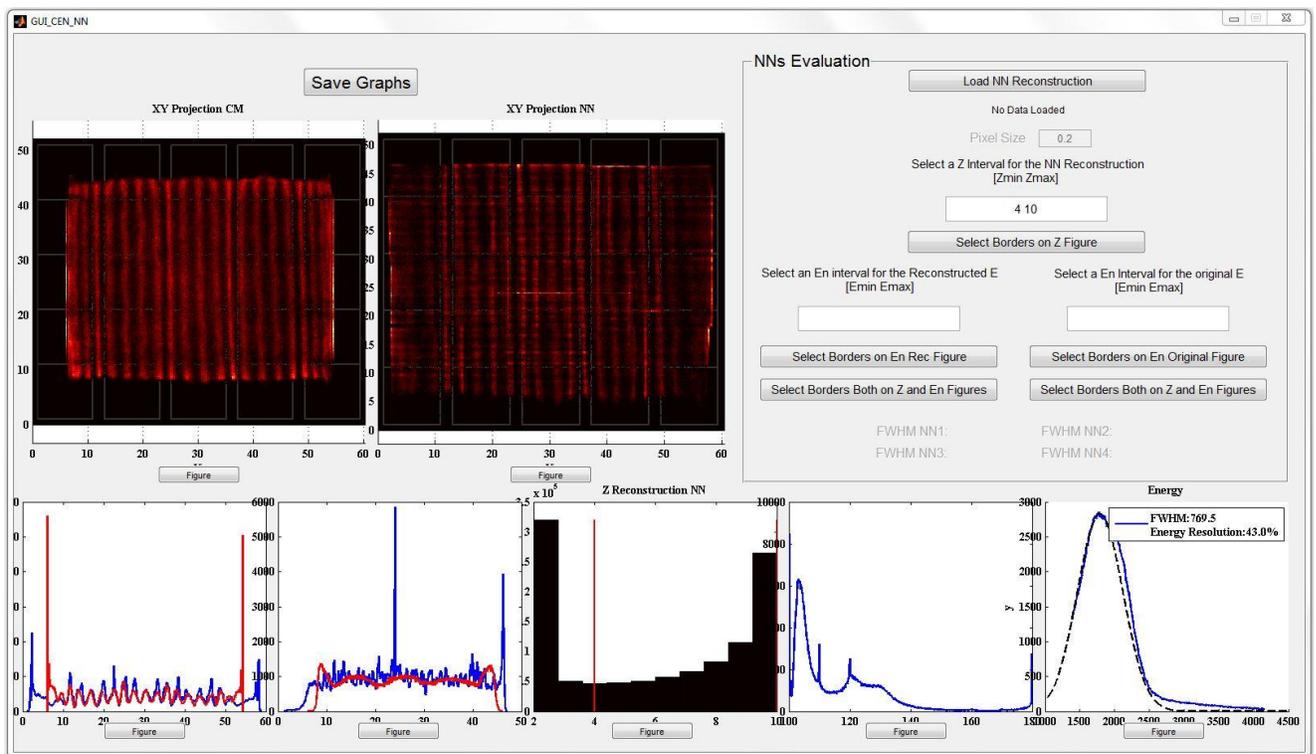


Figure 4.11 GUI to test and visualize experimental data

In this program the results of the NNs are shown respect to the Centroid Method (upper left and red lines on the graphs)). It is then possible to partially select the energy and the Z to see if a better reconstruction is obtained in a certain interval respect to another one

Chapter 5 Experimental Results

5.1 First results obtained from Training – Matrices used for the training

During the work, two type of matrices were used for the training of the networks which differ from the way the Z was determined:

- A 3D Matrix (obtained like a LUT) when each energy of interest (from 100keV to 160keV with a step of 10keV) was simulated per each 3D points chosen. The step along X and Y was linear and equal to 1mm, while for the Z coordinate the step was also linear and equal to 1mm, but the number of repetition for each point was distributed not equally but with a exponential distribution, to mimic the effective experimental one.
- A 2D Matrix when each energy of interest (from 100keV to 160keV with a step of 10keV) was simulated per each 2D points chosen. The step along X and Y was linear and equal to 0.2mm, while for the Z coordinate the positions were simulated using GEANT3 software.

Satisfactory results have been obtained with both training matrices and therefore the choice on which of the two will be finally used is still pending. However the 2D one has the advantage of the fact that the Z distribution is automatically obtained from GEANT3 and there is not the need to manually set the number of repetition accordingly to the exponential behavior.

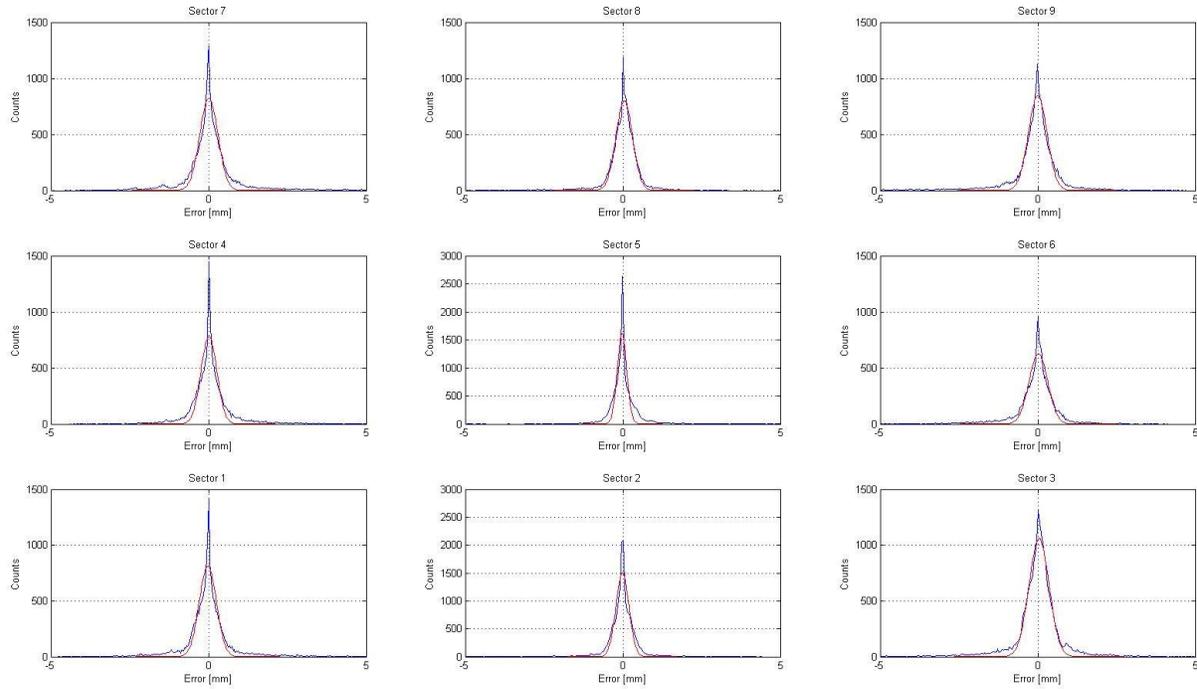


Figure 5.1 Results obtained from Training – X Error

Here is possible to observe the results obtained from the training of the networks using a pattern net with the X coordinate as the only output and as input only the readings of the 25 photodetectors.

It may be noticed that the border areas obtains worst results respect to the center one (as expected) however the FWHM calculated is in all cases less than 0.74mm.

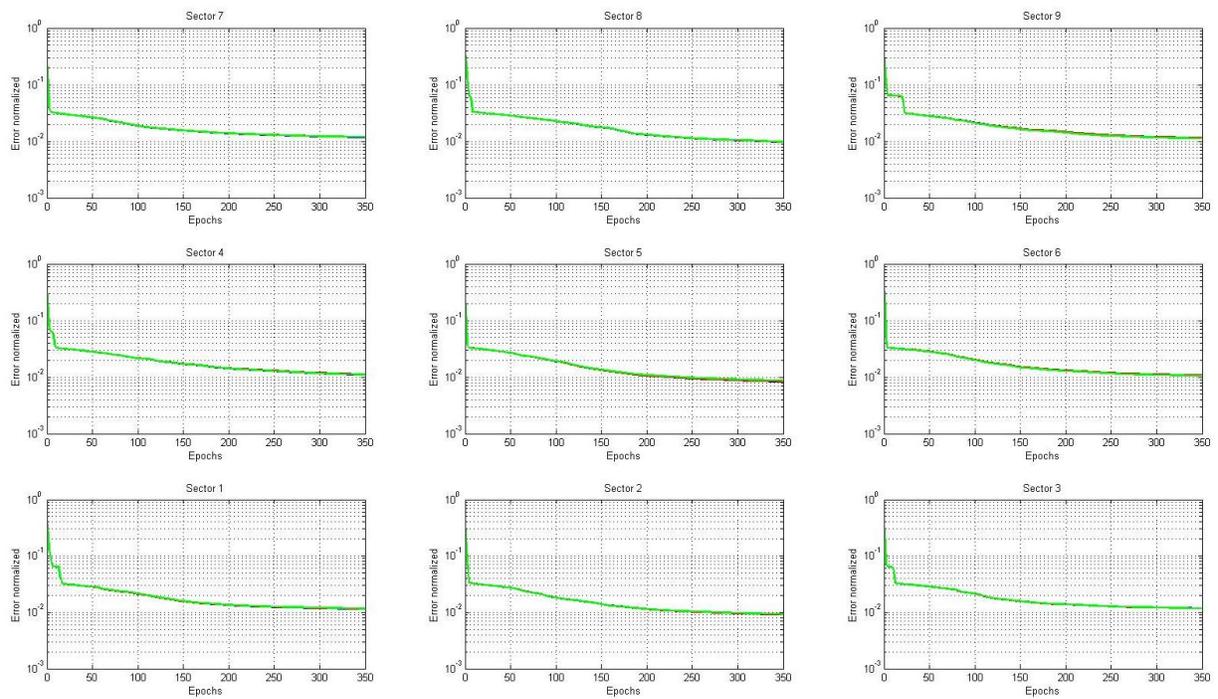


Figure 5.2 Results obtained from Training – X Error slope

These are the training slope for the previous case, it may be observed that they have all reached the maximum number of epochs, therefore a slightly improvement of the results may be obtained augmenting the number of epochs during the training.

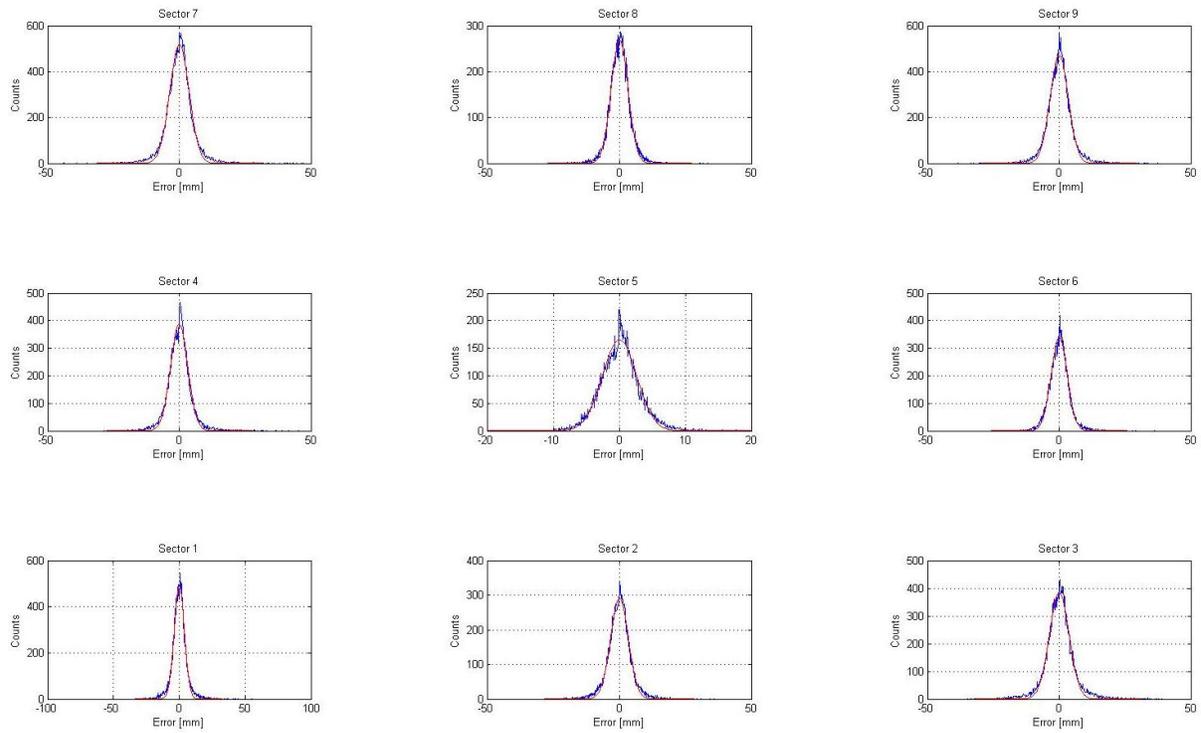


Figure 5.3 Results obtained from Training – E Error

In the case of the energy resolution, the better FWHM obtained both from the pattern net and the feedforward net was of 8keV with a used step of 10keV.

Way to improve this result may be to use a lower step for the energy simulated (at the moment it has been of 10keV) or try different network configurations.

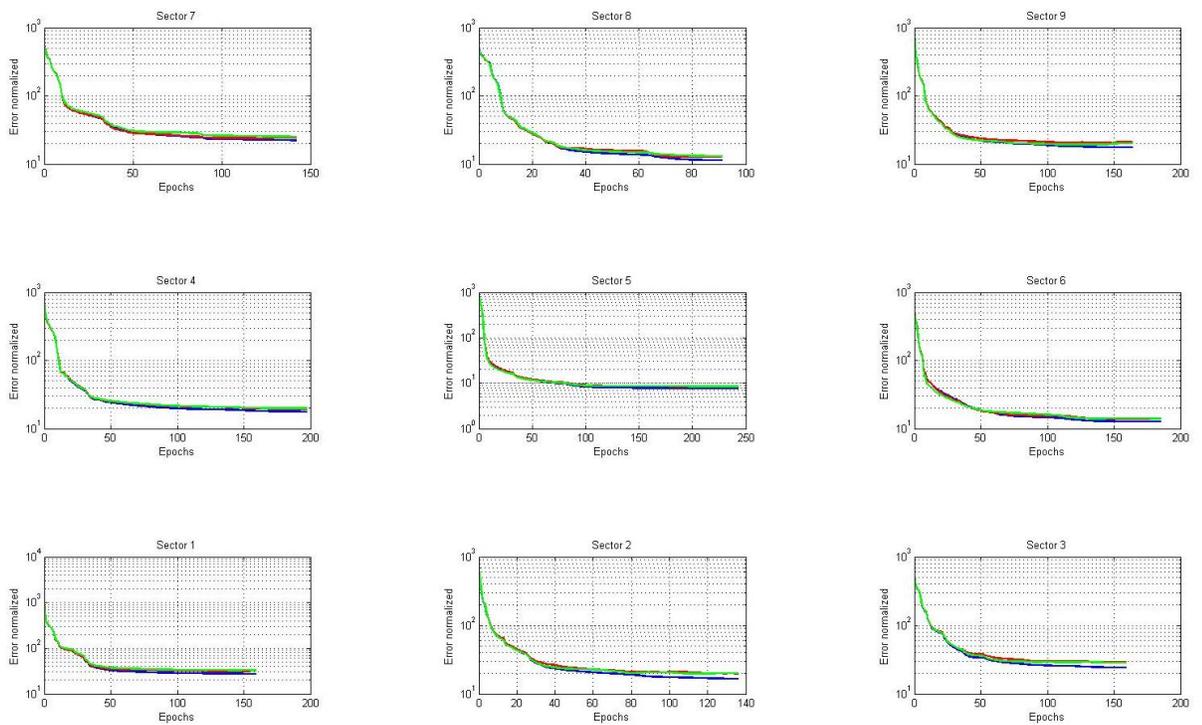


Figure 5.4 Results obtained from Training – E Error slope

It may be observed than in the energy training (differently from the X training), in all the zones the networks training reached a flat state before the last epoch (300 was the limit set), therefore simply augmenting that number can not improve the results, but it is necessary to modify the network structure or the training matrix.

5.2 XY reconstruction from Pattern Net by simulated Data

The previously trained NN with a patter net algorithm was then used on new simulated data with NO noise. A grid of point (green) was simulated and then reconstructed with the NN shown in figure 5.5 and the figure 5.6 depicts the results of Centroid method. It may immediately observed that the NN obtains an improved FOV (the border limit is due mainly either to the fact that the network was trained with a pattern net algorithm or that it was not trained so near the borders) and also that more points are easily distinguishable respect to the Centroid method

A quantum efficiency of 85% and $20e^-$ noise were added to the previous data and the obtained results are shown in the figures above for the pattern net. The same results were obtained using the feedforward net. It may be observed that the FOV is better respect to the simple CM, but the introduction of the noise without re-shifting accordingly the matrix created many reconstruction errors. This phenomenon may be clearly seen in the Z reconstruction as it will be shown in the next slides.

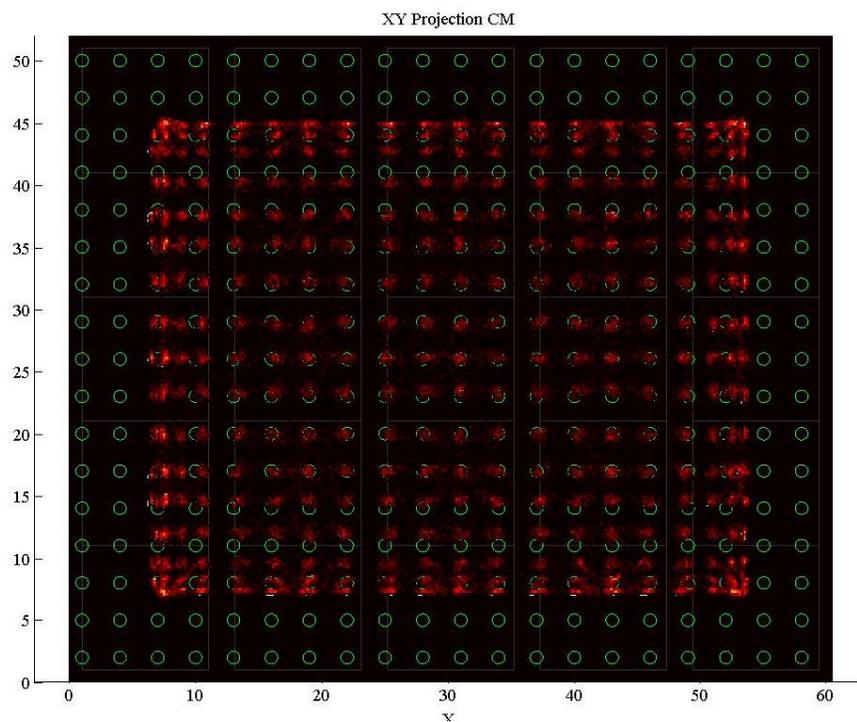


Figure 5.5 Simulated Data XY Reconstruction - Pattern Net

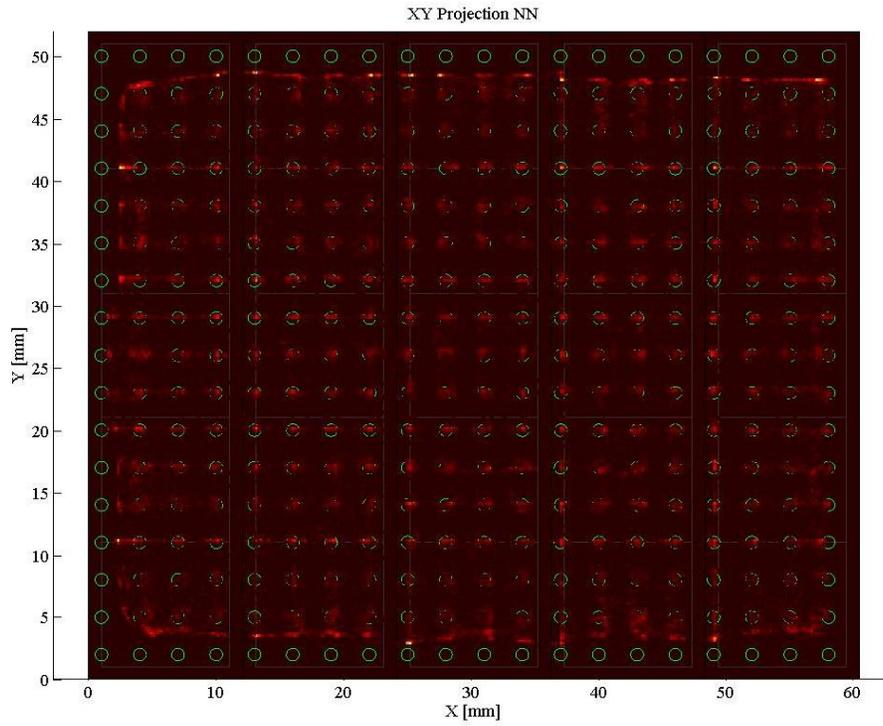


Figure 5.6 Simulated Data XY Reconstruction - Pattern Net

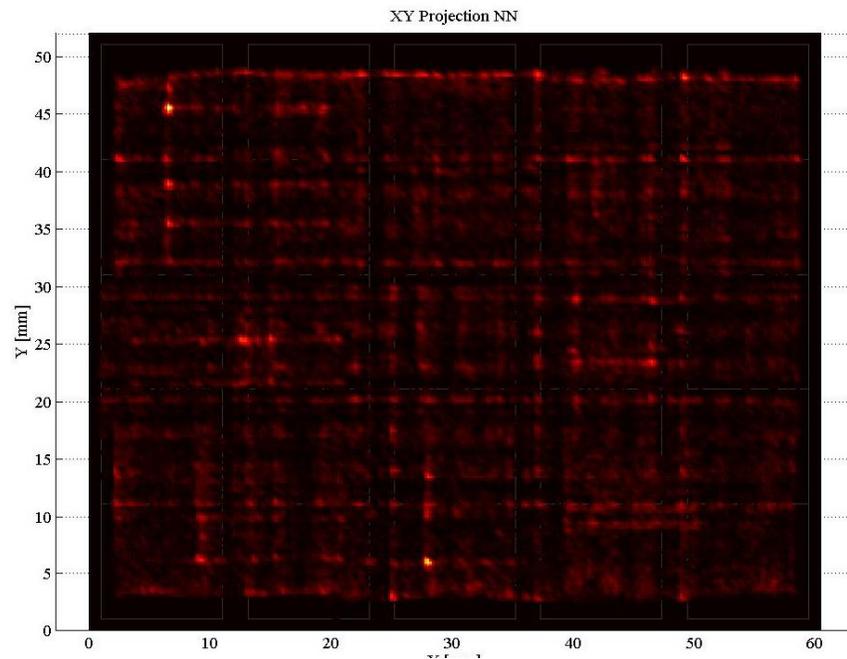


Figure 5.7 Simulated Data XY Reconstruction - Pattern Net

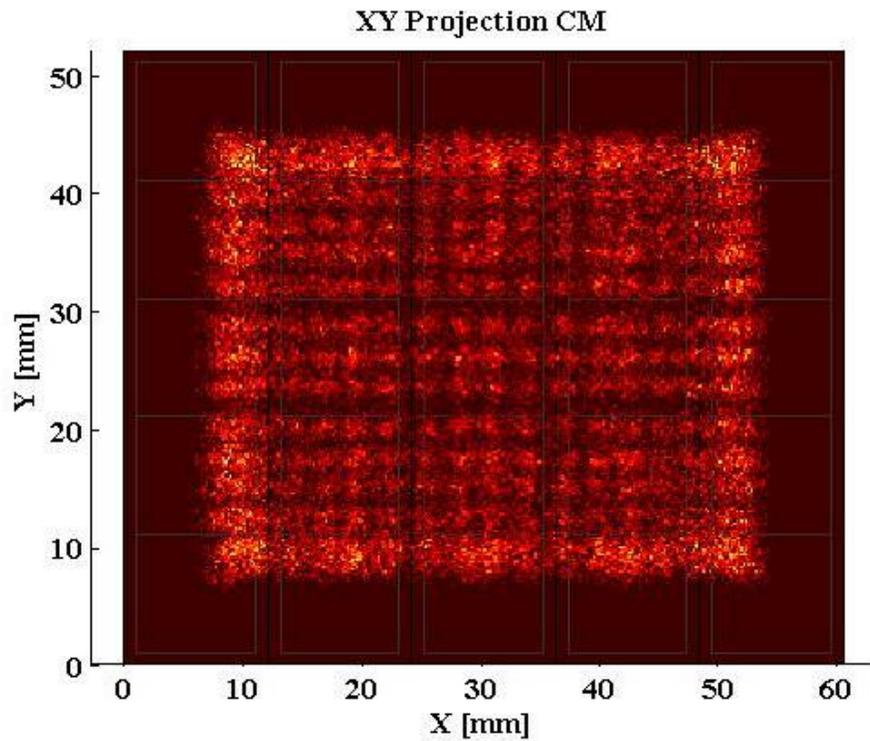


Figure 5.8 Simulated Data XY Reconstruction - Pattern Net

5.3 XY reconstruction from feedforward Net by simulated Data

On the same data (grid with NO noise), another trained NN with a feedforward net algorithm was then used. Figure 5.9 represent the results of FNN and figure 5.10 depicts the results of centroid method. It may immediately observed that respect to the pattern net the FOV here is limited only by the fact that the matrix used for the training was limited in the upper part. All the other points are reconstructed flawlessly by the network.

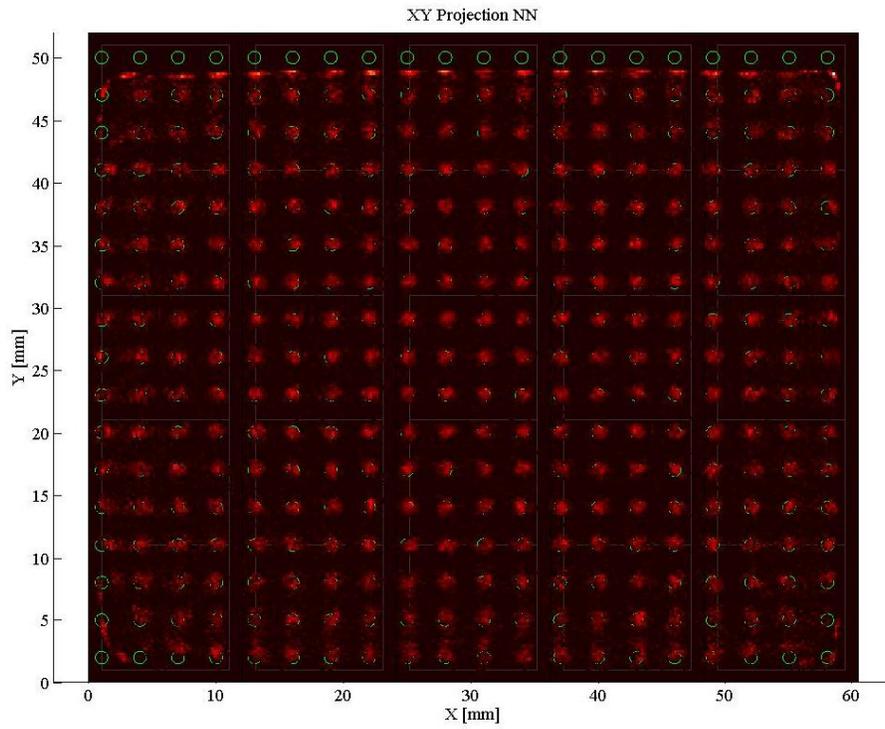


Figure 5.9 Simulated Data XY Reconstruction - Feedforward Net

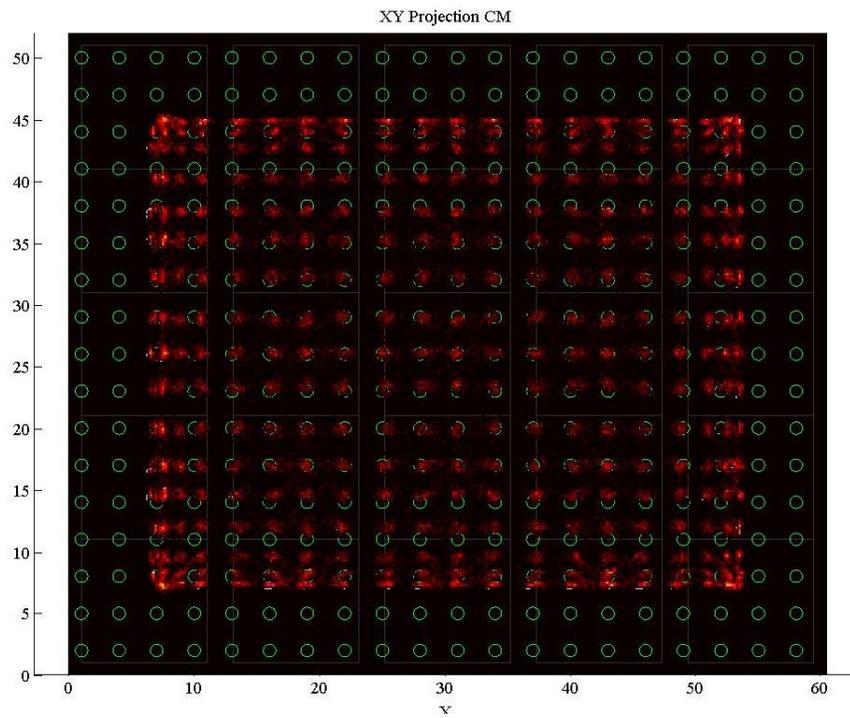


Figure 5.10 Simulated Data XY Reconstruction - Feedforward Net

5.4 Z Reconstruction by simulated Data

Figure 5.11 shows the Z reconstruction obtained with the pattern net algorithm, where different choices were made about how to consider the probabilities distribution. It may be observed that if all the probabilities are taken into account, there is a marked shift towards the center of the interval.

Figure 5.12 shows the Z reconstruction obtained with the feedforward net algorithm. It may be observed that the reconstructed Z is very similar to the simulated data.

Inserting the noise without correcting accordingly the matrix as caused an increment of the error in the XY axis and also the Z reconstruction is affected as shown in the figure 5.13 for the feedforward net with a peak of events with a Z less than 1mm (and therefore not correctly reconstructed and to be discarded during also the XY reconstruction)

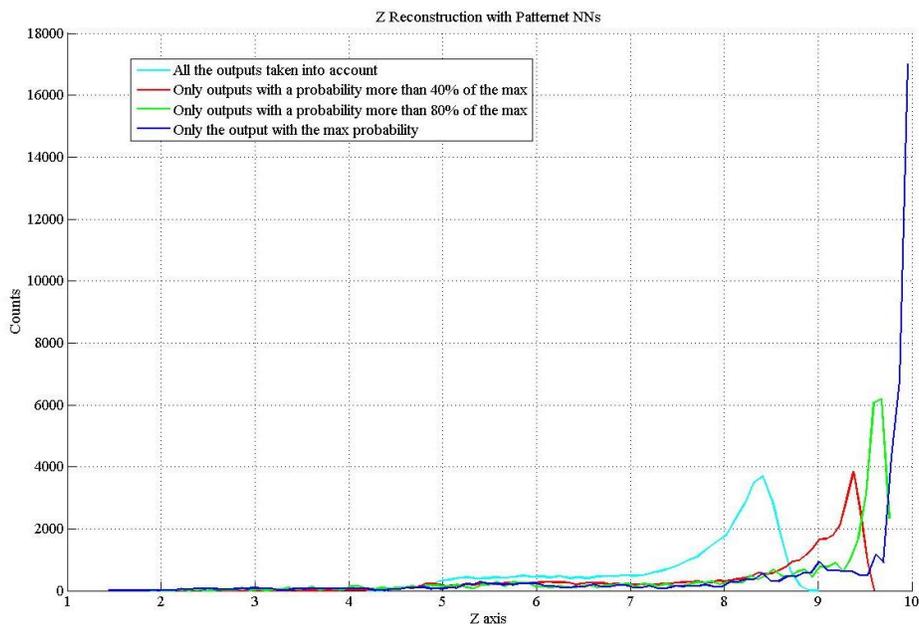


Figure 5.11 Simulated Data Z Reconstruction – Pattern Net

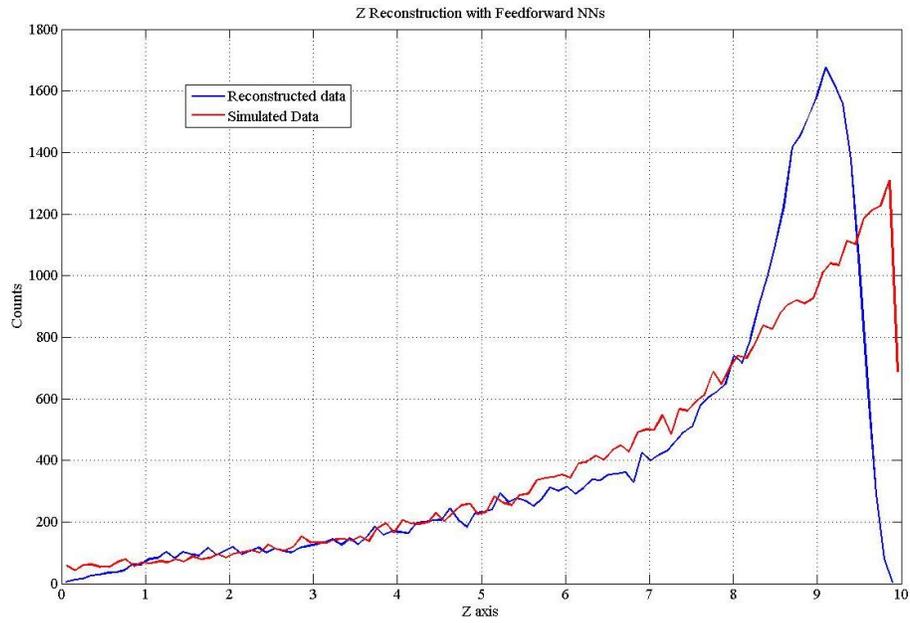


Figure 5.12 Simulated Data Z Reconstruction – Feedforward NN

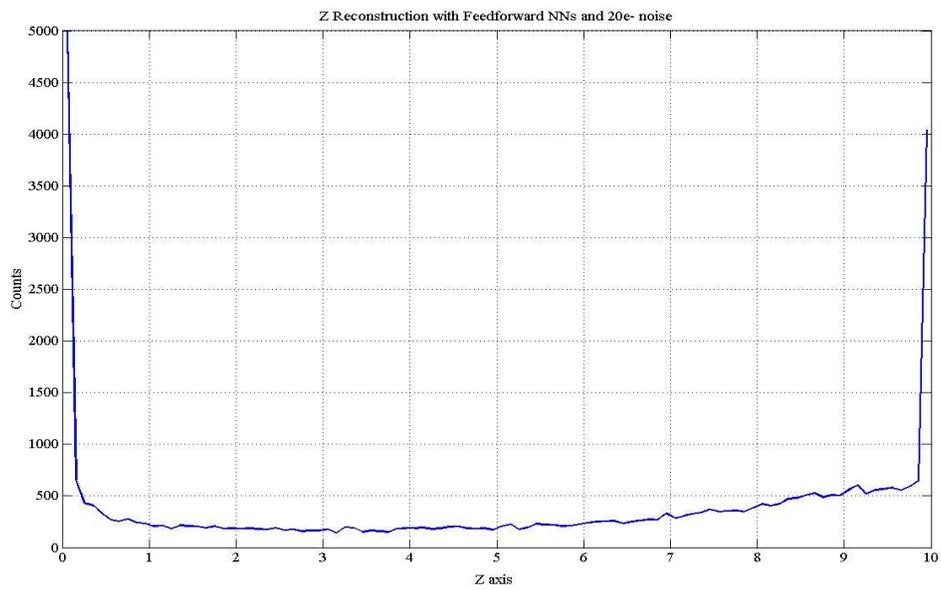


Figure 5.13 Simulated Data Z Reconstruction with Noise – Feedforward NN

5.5 Energy Reconstruction by simulated data

Two identical grids at 122keV and 140keV have been simulated and the energy have been reconstructed with the feedforward network. The result is visible on the left, while on the right there is the sum of the photons received by each photodetectors. The resolution obtained with the NNs is improved of 4 points for the low energy and 3 points for the higher one.

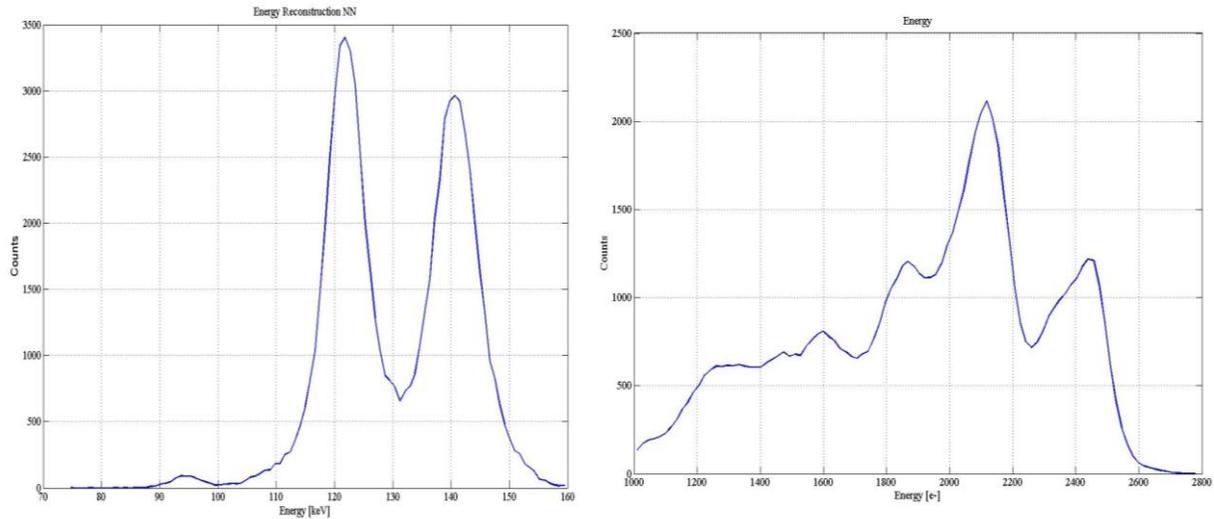


Figure 5.14 Simulated Data Energy Reconstruction

	FWHM		Resolution (%)	
	122keV	140keV	122keV	140keV
Reconstructed	9.8 [keV]	10.2 [keV]	8	7.3
Simulated	264 [ph]	237 [ph]	12.6	10

Table 5.1

5.6 XY Reconstruction from feedforward Net by Experimental Data

From the first results obtained, by looking at figure 5.15 it is immediate to see that the division in zone is causing some problems with the experimental data. The points that are visible in the image represent in fact the corners of the 9 reconstruction zones. This error occurred also when inserting noise in the previously simulated grid.

It may be observed however that the FOV is improved and the shape correctly reconstructed.

A next step to improve this reconstruction will be to try to train the FFN without the subdivision in 9 zones.

After considering only 1 zone for the training, the result is shown on the right. There are still some problems relative to the points located on the divisions between detectors and on the dead zones as shown in figure 5.16, but the reconstruction improved

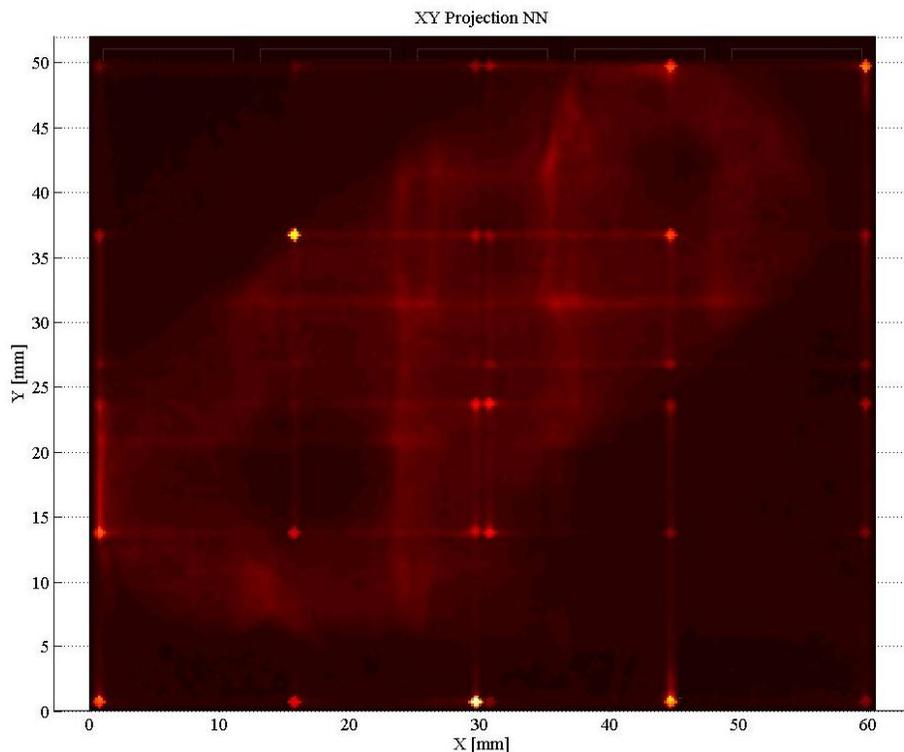


Figure 5.15 Experimental Data XY Reconstruction with nine Sectors - Feedforward net

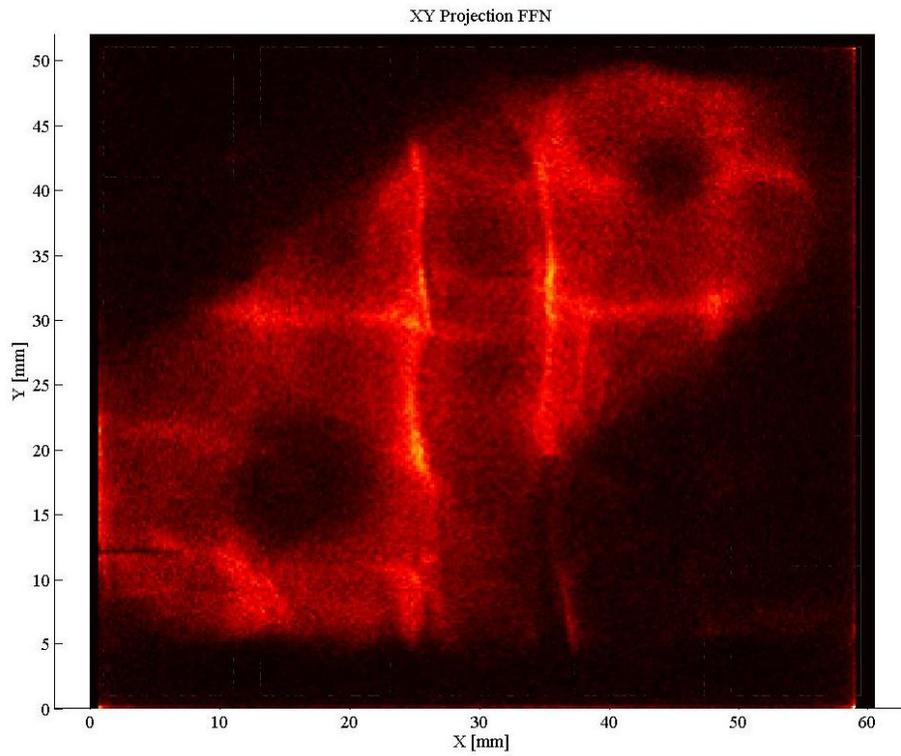


Figure 5.16 Experimental Data XY Reconstruction with one Sector - Feedforward net

5.7 Z Reconstruction from feedforward Net by Experimental Data

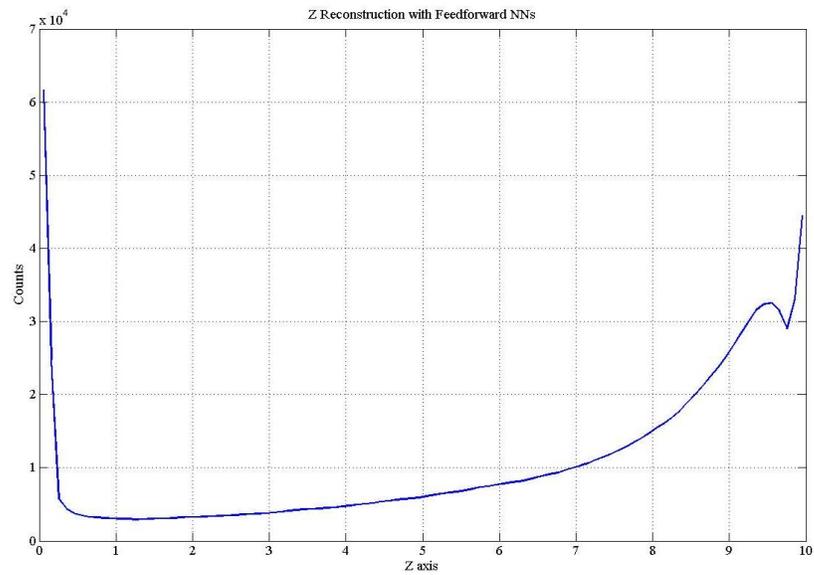


Figure 5.17 Experimental Data Z Reconstruction - Feedforward net

5.8 Energy Reconstruction from feedforward Net by Experimental Data

From this energy reconstruction it may be clearly observed that the number of photons expected by the NN and the one arrived in the experimental situation are very different one from each other.

To solve this problem, two paths are possible:

- Filter more the data matrix, in this way the total number of read photons will be inferior and the peak should move towards the correct position
- Improve the GUI_Stretching program, which adjusts also the Correction Factor between experimental and simulated data, in a way that the Correction Factor becomes not only an area property, but also a property for every single photodetector

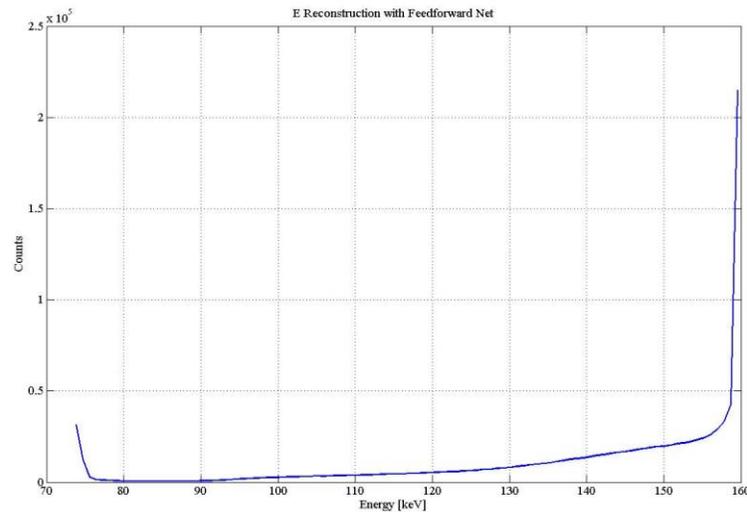


Figure 5.18 Experimental Data Energy Reconstruction - Feedforward net

5.9 XY Reconstruction from Pattern Net by Experimental Data

With the pattern net, the problem is also linked to the subdivision of the 9 reconstruction zones, but instead to put the events at the corners, vertical and horizontal lines are created.

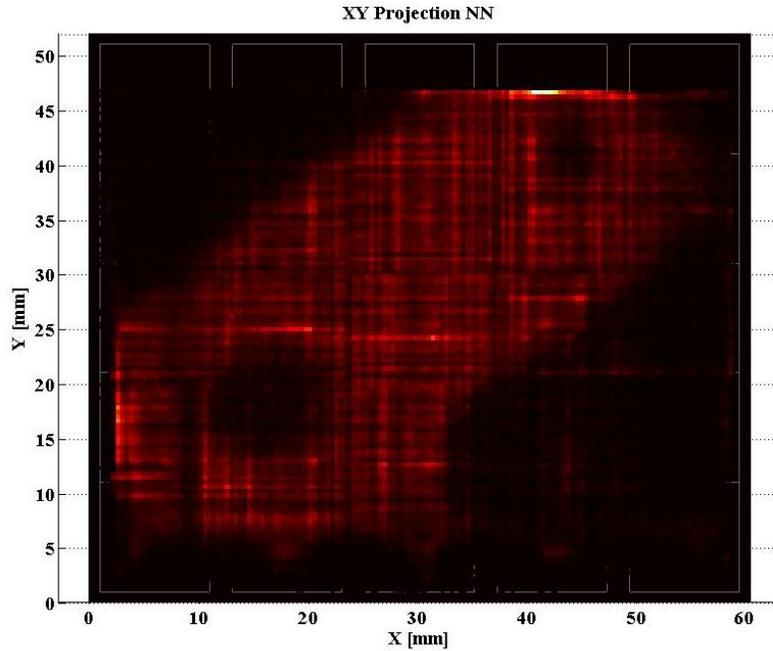


Figure 5.19 Experimental Data XY Reconstruction - Pattern net

5.10 Z Reconstruction from Pattern Net by Experimental Data

It may be noticed that the Z reconstruction (with a filter at the 30% of the maximum) do not have the peak at 1, however, the slope expected was more the one obtained by the FFN.

Also in this case it may be observed that the FOV is improved and the shape correctly reconstructed.

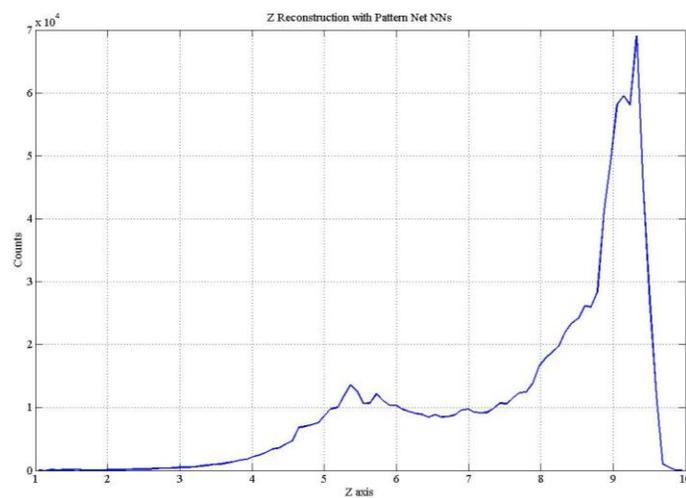


Figure 5.20 Experimental Data Z Reconstruction - Pattern net

Conclusion

In a comparison of Feedforward Neural Network and Pattern Net, Feed forward NN is a promising method for the reconstruction of the first event of interaction of gamma ray at low energies. The distribution of photons inside crystal follow normal distribution as feed forward NN and ML method based on normal probability density function while pattern net based on Bernoulli probability density function which is more useful for Compton effect in order to classify 1st and 2nd interaction.

In future works FNN requires to create a 2D matrix with a lower step between points (which, without considering the 9 zone division, will require either to upgrade the memory of our PC or to try to use different train algorithms respect to the Levenberg-Marquardt, since it will cause an out-of memory error due to the dimensions of the matrix required). Improve the GUI_Stretching program, which adjusts also the Correction Factor between experimental and simulated data, in a way that the Correction Factor becomes not only an area property, but also a property for every single photodetector. Filter more both the data matrix used for the training and the respective experimental one, in this way the noise will be completely filtered out. Methods that can be used at the medium high energies (662keV), but not at the low energies (122-140keV), since it will cause a loss in the signal.

Bibliography

- [1] Static and Dynamic Neural Networks, 5 Apr 2004, John Wiley & Sons.
- [2] Dynamical Systems in Neuroscience, Di Eugene M. Izhikevich
- [3] Rojas, R. (1996). Neural Networks: A Systematic Introduction. Springer, Berlin
- [4] Neural Networks: A Comprehensive Foundation, Haykin Simon, 1998.
- [5] Neuroscience Mark F. Bear, Barry W. Connors, Michael A. Paradiso – 2007
- [6] Christopher M. Taylor 1997
- [7] <http://homepages.gold.ac.uk/nikolaev/311perc.htm>
- [8] John A. Bullinaria, 2012, <http://www.cs.bham.ac.uk/~jxb/INC/I5.pdf>
- [9] Bishop, C. (1995) "*Neural Networks for Pattern Recognition*", Oxford University Press, Oxford.
- [10] IGOR AIZENBERG Published in Soft Computing, vol. 11, No 2, January, 2007, pp. 169-183.
- [11] *Applied Linear Regression, 3rd Ed.* by Sanford Weisberg, published by Wiley, 2005.
- [12] Neural Networks, R. Rojas, Springer-Verlag, Berlin, 1996.
- [13] Adaptive Filter Theory, Haykin Simon, 4e, 2002, Prentice Hall.
- [14] Method of Steepest Descent and its Applications, Xu Wang, 2008
- [15] The LM method for nonlinear least squares curve-fitting problems, Henri Gavin, 2011.
- [16] CERN. GEANT3. From: <http://wwwasd.web.cern.ch/wwwasd/geant/>
- [17] L. Boschini and C. Fiorini. Scidra: a monte carlo simulator for the study and optimization of gamma cameras based on a scintillator coupled to a silicon photodetectorarray. Nuclear Science Symposium 464-469 vol.1, 1:464{469, 1999.
- [18] cgal. Computational geometry algorithms library <http://www.cgal.org/>, 2009.
- [19] S. Cova and A. Longoni. An introduction to Signals, Noise and Measurements. John Wiley and Sons, 1979.
- [20] H. Cramer. Mathematical Methods of Statistics. Princeton University Press, 1957.
- [21] W. Feller. An Introduction to Probability Theory and Its Applications, volume 1. John Wiley and Sons, Inc., 1968.
- [22] C. Fiorini, A. Laratta, R. Peloso, A. Longoni, T. Mennini, and E. Micotti. Study of compatibility of a silicon drift detector with a mri system. IEEE Nuclear Science Symposium Conference Record, M 10-62, 2008.
- [23] E. Gatti and P. Rehak. Semiconductor drift chamber - an application of a novel charge transport scheme. Nucl. Instr. and Meth., A 225:608-614, 1984.
- [24] Hicam project homepage. <http://www.hi-cam.org>.
- [25] G. F. Knoll. Radiation detection and measurement. John Wiley and Sons, Inc., 3 edition, 1999.
- [26] S.K. Moore, W.C.J. Hunter, L.R. Furenlid, and H.H. Barrett. Maximum-likelihood estimation of 3d event position in monolithic scintillation crystals: Experimental results. IEEE Nuclear Science Symposium Conference Record, M19-21:3691{3694, 2007.
- [27] Nuclear fields. <http://www.nuclearfields.com>.
- [28] L. Ottobriani, G. Lucignani, M. Clerici, and M. Rescigno. Assessing cell tracking by noninvasive imaging techniques: applications in experimental tumor immunology. Q. J. Nucl. Med. Mol. Imaging, 49:361-366, 2005.
- [29] S. Pedemonte. Digital signal processing nella scintigraphia. Master's thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Oct. 2004.
- [30] K.S. Shaw, J. Glodo, M. Klugerman, W.M. Higgins, T. Gupta, and P. Wong. High energy resolution scintillation spectrometers. IEEE Transaction on Nuclear Science, 51(5):2395-2399, October 2004.