Neural networks for Gamma cameras

A.J. de Hartog

TU Delft

Neural networks for Gamma cameras

by

A.J. de Hartog

to obtain the degree of Bachelor of Science
at the Delft University of Technology,

Student number: 5028833
Project duration: February 2022 – July 2022
Thesis committee: Dr. ir. M. C. Goorden, TU Delft, supervisor
Dr. K. S. Postek, TU Delft, supervisor
Dr. E. van der Kolk, TU Delft
Prof.dr.ir. C. Vuik, TU Delft

TU Delft

Abstract

In this research the effectiveness of analytical neural networks compared to the maximum likelihood method on the prediction of spatial and DOI positioning of a Gamma detector with a NaI(Tl) scintillator of size 590mm x 470mm x 40mm (x,y,z), with a glass lightguide of size 620mm x 500mm x 4mm and a PMT area of 620mm x 500mm x 40mm with 2-inch round PMTs with a Bialkali photocathode is presented. This is done by training neural networks with different cost function, different amounts of hidden layers and different amounts of neurons per hidden layer, trained on different amounts of training data. The resolution of the predictions of the testing data are compared with those of the maximum likelihood method. It was concluded that the neural network with best spatial resolution, had the Huber loss function as cost function, 4 hidden layers and 512 neurons per hidden layer and was trained on 29,970 datapoints. The FWHM and the FWTM were $3.83\pm, 0.54$ mm and 12.49 ± 1.19 mm respectively, while the FWHM and the FWTM of the maximum likelihood method were 3.31 mm and 12.13 mm respectively. The resolution of the neural network was lower than that of the maximum likelihood method. The same was done for the DOI resolution, here a neural network with mean squared error as cost function, 4 hidden layers and 64 neurons per hidden layer trained on 9,990 datapoints, gave the best the resolution with FWHM and FWTM equal to 6.00 ± 0.50 mm and 11.94 ± 0.94 mm respectively. The FWHM and FWTM of the maximum likelihood method were 6.16 mm and 11.22 mm respectively. This made the DOI resolution of the neural network higher then that of the maximum likelihood method. Finally different ideas were presented to increase the resolution of the neural network. These were: training the neural network on independent data, split the neural network in a spatial part and a DOI part, create a more complex architecture and making use of a convolutional neural network.

Contents

1

Introduction

The Single Photon Emission Computed Tomography (SPECT) scan is a technique widely used in nuclear medicine for imaging [1]. SPECT scans are used in a variety of different circumstances, including the diagnosis of Alzheimer's disease [2], localizing different types of tumours [3], evaluating organ activity [4] and cardiac imaging [5]. However, next to the need of studying humans with SPECT scans, there is also a need to study the dynamic of intact small animals. With these preclinical SPECT scans it is possible to clarify molecular interactions in the onset and progression of disease, asses potential imaging agents, and monitor therapeutic effectiveness of pharmaceuticals serially within a single-model system [6].

SPECT scans have a long history, which is briefly summarized as follows: In 1923 Georg von Hevesy used radioactive tracers in an organism which made it possible to look at dynamic systems in the body. For his work, Von Hevesy won the Nobel prize in 1943 [7]. In 1958 Hal Anger developed the Gamma camera (also known as the Anger camera), this resulted in the development of the SPECT scan in 1963 and the Positron Emission Tomography (PET) scan in 1973 [8].

There are several differences between preclinical SPECT scans and preclinical PET scans. Most notable in performance is the difference in sensitivity and resolution. With the use of a pinhole collimator, the sensitivity of a preclinical PET scan is in the order of magnitude 3 greater compared to that of a preclinical SPECT scan, meaning that the SPECT scan detects a lower percentage of emitted events than the PET scan [9]. However the resolution of a preclinical SPECT scan can be better than that of a preclinical PET scan when a pinhole collimator is used [10].

In addition to the difference in performance, there is also the difference in use of radioactive tracers. Multiple SPECT tracers can be used for imaging simultaneously. For PET scans this is not the case, because the energy of the γ photons created by the positron annihilation is always the same (511 KeV). With Versatille Computed Emission Tomography (VECTor) it is possible to make images of the energy ranges of both the SPECT tracers and PET tracers [11].

In Decuyper *et al.* [12], the use of Artificial Neural Networks (ANN) for preclinical PET scans was discussed. Here a 17% improvement in the Full Width Half Maximum (FWHM) was obtained when an optical ANN was used compared to the nearest neighborhood algorithm. Since Decuyper *et al.* made such in improvement using an ANN, it is interesting to see if an ANN can be created for the more cost efficient VECTor.

In this research an Anger camera is used with a NaI(Tl) scintillator (predominantly used for SPECT scans [13]) of size 590mm x 470mm x 40mm (x,y,z), with a glass lightguide of size 620mm x 500mm x 4mm and a PMT area of 620mm x 500mm x 40mm with 2-inch round PMTs with a Bialkali photocathode. The goal of this research is to create an ANN such that either the FWHM or the Full Width Tenth Maximum (FWTM) becomes smaller than that of the Maximum Likelihood (ML) algorithm. This results in the following research question:

Can an ANN be created with smaller FWHM or FWTM than that of the ML algorithm, for a Gamma detector based on a monolithic 40mm thick NaI(tl) scintillators with a configuration of two inch round PMTs with a 4mm glass light-guide?

In chapter 2, the concept of a neural network will be discussed and the the propagation algorithm will be mathematically derived. In chapter 3, the working of the Gamma camera will be discussed as will the physical phenomenon of Compton scattering. In chapter 4, the construction of the data and the methodology of how the results are achieved will be discussed. In chapter 5, the results of the neural network will be presented, compared and discussed. In chapter 6, a conclusion will be made and advice for future research will be given.

2

Neural Networks

In this chapter the concept of neural networks will be discussed. This will be done by first explaining how neural networks work, and then what type of neural networks there exist.

2.1. Neurons

in 1943 a neurophysiologist and a logician came up with the idea of simulating the brain with mathematics, also known as a neural network [14]. The human brain consists of several million interconnected neurons. A neuron itself consists out of three major parts, axons, dendrites and the soma, as can be seen in figure 2.1.

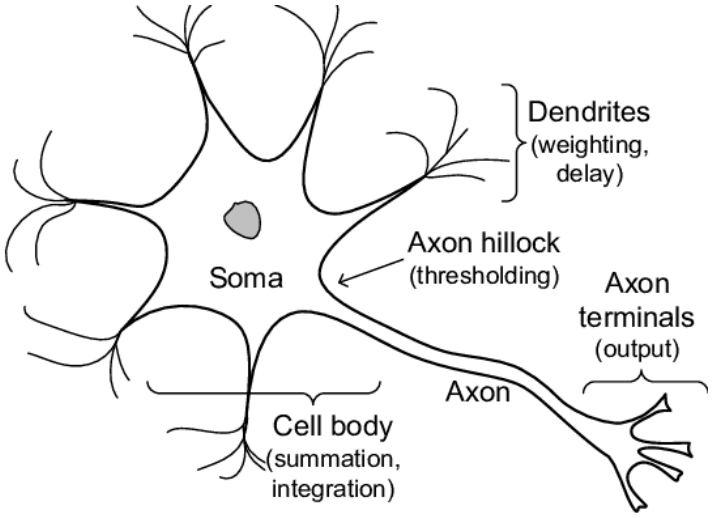

Figure 2.1: Schematic representation of a neuron, where it can be seen that it consists out of three main parts.[15]

These three parts have different properties. The dendrite is the receiving part of the neuron, it has a synapse of an adjacent axon as input and sends a synapse to the soma. The soma then confirms if enough dendrites have received a signal, once a specific threshold is reached. The soma will then send a synapse via the axon to an adjacent neuron where this process will happen again [16].

When there is a system of these neurons they can achieve very complex tasks, actually everything that a human can do is because of these neurons [17].

2.2. Data

To make use of a neural network, several types of data are used. There is the input and output data (also known as label). These two data sets are split in three parts, the training data, validation data and the testing data. The training data is used to train a neural network, the validation data is used to

make sure the neural network does not overfit and the testing data is used after the neural network is trained to test the neural network using different properties.

2.3. Working of a neural network

To determine what the response of a neuron on the previous neurons is (keeping figure **??** in mind), all the values of the previous neurons are multiplied with a weight and then summed, as can be seen in figure 2.2.

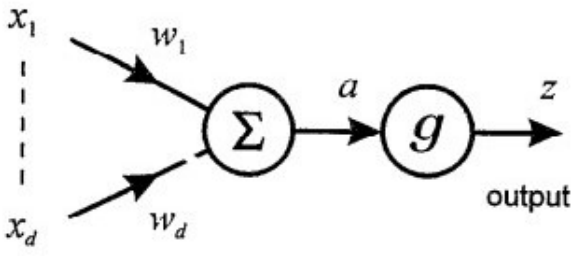

Figure 2.2: Visualisation of the weights, here x_1, ..., x_d are the inputs, w_1, ..., w_d are the weights, a is the summed value of the inputs times the weights and g is the activation function [18].

Often also a bias is added to the summed input. Mathematically, the value of a, depicted in figure 2.2, is given as in equation (2.14).

$$a = \sum_{i=1}^{d} w_i x_i + b \tag{2.1}$$

After the weighted sum is calculated and the bias is added, an activation function is used to determine the output of the neuron. Activation functions are used to create more complex mappings from the input to the output. If activation functions are not used, then the neural network is only a linear regression model, which has a limited performance and power compared to a neural network were activation functions are used [19]. The working of a neural network is schematically shown in figure 2.3

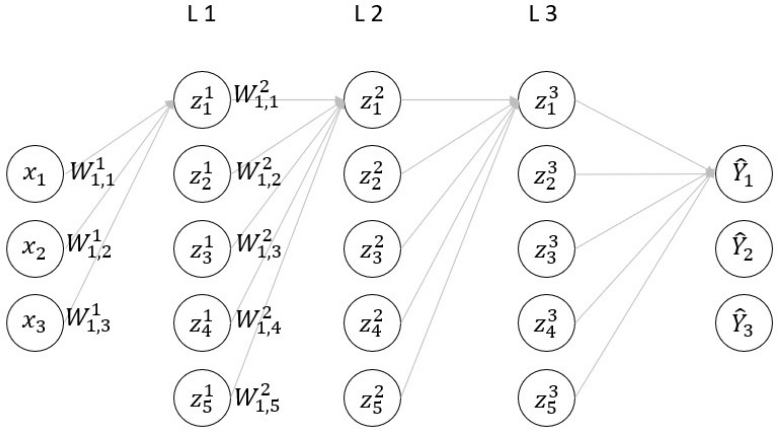

Figure 2.3: Working of a neural network. Here the input is given by x_i and the output is given by $\hat{Y}_i$. The output of the neurons is z^l_j where j is the neuron and l is the hidden layer. The outputs of the neurons are multiplied by their respective weights $W^l_{k,j}$, where k is the neuron with which the output is multiplied, j is the neuron where the weights are added up and l is the layer where the weights are added up.

There is a wide variety of activation functions, some of which are depicted in figure 2.4.

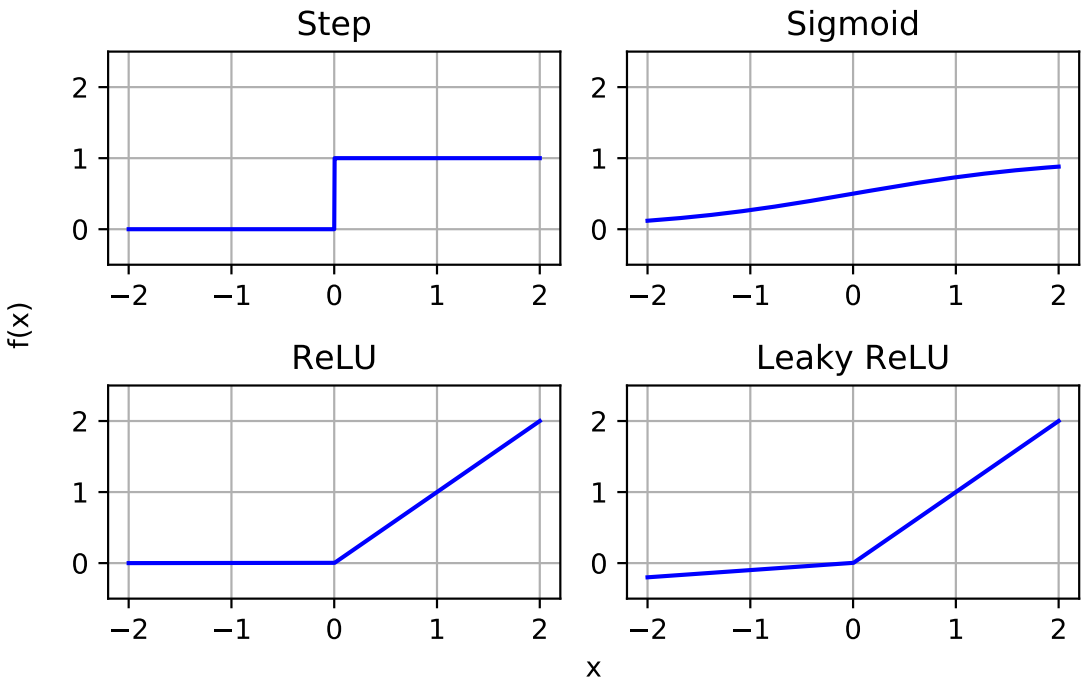

Figure 2.4: Four commonly used activation functions, the step function, the sigmoid function the ReLU function and the leaky ReLU function.

Each activation function has its drawbacks, however, overall the Rectified Linear Unit (ReLU) and its alteration the leaky ReLU are seen as the most efficient activation functions for regression models [19]. These two activation functions are given by equation 2.2 and equation 2.3.

$$ReLU(a) = \begin{cases} 0 & a < 0 \\ a & a \geq 0 \end{cases} \tag{2.2}$$

$$LeakyReLU(a) = \begin{cases} 0.01a & a < 0 \\ a & a \geq 0 \end{cases} \tag{2.3}$$

Between these two activation functions most drawbacks are the same. The main difference is that the leaky ReLU does not suffer from vanishing gradient problem [20], which also will be briefly discussed in the following section.

2.4. Training of a neural network

2.4.1. Gradient descent

When a neural network is set up with different amounts of hidden layers, neurons and activation functions it will have to be trained to see which configuration of weights and biases gives the best result. This is often done with the method of gradient descent. In this explanation first the direct algebraic gradient descent will be discussed, after that the matrix notation of gradient descent will be discussed.

Let z_j^l be the output of the j^{th} neuron on the l^{th} layer. Then we can write:

$$z_j^l = g(\sum_k w_{j,k}^l z_k^{l-1} + b_j^l) = g(a_j^l) \tag{2.4}$$

Where g is the activation function, $w_{j,k}^l$ is the weight on the k_{th} neuron in the $l-1$ layer, b_j^l is the bias and $a_j^l = \sum_k w_{j,k}^l z_k^{l-1} + b_j^l$.

The main goal of gradient descent is to minimize a cost function. Let equation 2.5 be an example of a cost function (in this case the mean squared error) of the training example $\mathbf{x_i}$, where x_i represents the i^th input (set).

$$E_i = \frac{1}{2}\sum_{j=1}^{J}(y_j(\mathbf{x_i}) - z_j^L(\mathbf{x_i}))^2 \tag{2.5}$$

Here y is the desired output, J is the amount of output neurons and L is the total amount of layers. Note that $\mathbf{x_i}$ can also be a vector since the input of a neural network is not restricted to one value. The total cost function is then given by $\sum_i E_i$ [21].

Gradient descent is defined with the following iterative step:

$$z_j^l(n+1) = z_j^l(n) - \eta\delta_j^l(n) \tag{2.6}$$

Where n is an integer denoting the step, η is the step size and $\delta_j^l(n)$ is a slight alteration such that the cost function decreases[22]. We define this alteration as follows:

$$\delta_j^l(n) \equiv \frac{\partial E}{\partial a_j^l(n)} \tag{2.7}$$

Using the chain rule equation 2.7 can be written as:

$$\delta_j^l(n) = \frac{\partial E}{\partial z_j^l(n)}\frac{\partial z_j^l(n)}{\partial a_j^l(n)} \tag{2.8}$$

Note that $z_j^l(n)$ is a function of $a_j^l(n)$ given by the activation function, as written in equation 2.4. Keeping this in mind equation 2.8 is equivalent to:

$$\delta_j^l(n) = \frac{\partial E}{\partial z_j^l(n)}g'(a_j^l(n)) \tag{2.9}$$

Where g' is the derivative of the activation function g. For the activation functions given in equation 2.2 and 2.3, the derivatives are quite simple. Note that for the RELU this derivative is equal to zero, for an input smaller then zero. This is the vanishing gradient problem, which has the consequence that the weights do not change in gradient descent method (since δ_j^l is zero). Also note that the first term of the equation (the derivative of the cost function) can also be quite easily computed, as can be seen in equation 2.10.

$$\frac{\partial E}{\partial z_j^l(n)} = |y_j(n) - z_j^l(n)| \tag{2.10}$$

Therefore, with this cost function, we can write equation 2.8 as the following equation.

$$\delta_j^l(n) = |y_j(n) - z_j^l(n)|g'(a_j^l(n)) \tag{2.11}$$

However, for the sake of generality, equation 2.9 will be used from now on.

An interesting aspect of the error is that the error of a neuron on a layer l, can be determined by the error of the neurons on the $l+1$ layer, this can be done by using the chain rule, which is applied as follows.

$$\frac{\partial E}{\partial z_j^l(n)} = \sum_k \frac{\partial E}{\partial a_k^{l+1}(n)}\frac{\partial a_k^{l+1}(n)}{\partial z_j^l(n)} \tag{2.12}$$

Using the definition of $a_i^{l+1}(n)$ (and taking the derivative of it with respect to $z_j^l(n)$) and $\delta_j^{l+1}(n)$ and making use of equation 2.9, $\delta_j^l(n)$ can be determined as follows.

$$\delta_j^l(n) = \sum_k w_{k,j}^{l+1}\delta_j^{l+1}(n)g'(a_j^l(n)) \tag{2.13}$$

Determining the error of a neuron itself is not very interesting. However, with this error the error of the weights and biases can be determined, and therefore they can be optimised. This is once again done by using the chain rule and by making use of the definitions of $a_i^l(n)$ and $\delta_j^l(n)$ [21].

$$\frac{\partial E}{\partial w_{j,k}^l(n)} = \frac{\partial E}{\partial a_j^l(n)} \frac{\partial a_j^l(n)}{\partial w_{j,k}^l(n)} = z_k^{l-1} \delta_j^l(n) \tag{2.14}$$

$$\frac{\partial E}{\partial b_j^l(n)} = \frac{\partial E}{\partial a_j^l(n)} \frac{\partial a_j^l(n)}{\partial b_j^l(n)} = \delta_j^l(n) \tag{2.15}$$

When the gradient descent method described in equation 2.6 is used on the weights and biases, these will be optimised.

2.4.2. Matrix-vector notation

Since a neural network is a grid of neurons, it is a lot easier to rewrite this method in matrix notation. To do this, all normal matrix manipulations are used, except that also the Hadamard product is also used. Which is defined as follows:

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \odot \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ \vdots \\ a_n b_n \end{bmatrix} \tag{2.16}$$

It can easily be seen that a lot of equations of the previous section can be written directly in matrix notation. Therefore, the derivation of these equations will not be shown, but the results will be.

First equation 2.4 can be rewritten as follows:

$$\mathbf{z^l} = g(\mathbf{w^l z^{l-1}} + \mathbf{b^l}) = g(\mathbf{a^l}) \tag{2.17}$$

Where $\mathbf{z^l}$ is a vector with all the z_j^l values, $\mathbf{w^l}$ is a matrix where the i^{th} row corresponds to the weights of the i^{th} neuron, $\mathbf{b^l}$ is a vector with all the b_j^l values and $\mathbf{a^l}$ is a vector with all the a_j^l values.

Using these matrix-vector notations, the three most important equations, namely equation 2.9, equation 2.13, equation 2.14 and equation 2.15 can easily be written in the simplified form of equation 2.18, equation 2.19, equation 2.20 and equation 2.21.

$$\delta^l(n) = \nabla_{\mathbf{z^l}} E \odot g'(\mathbf{a^l}(n)) \tag{2.18}$$

$$\delta^l(n) = (\mathbf{w^{l+1}})^T \delta^{l+1}(n) \odot g'(\mathbf{a^l}(n)) \tag{2.19}$$

$$\nabla_{\mathbf{w^l}} E = \mathbf{z^{l-1}} \odot \delta^l(n) \tag{2.20}$$

$$\nabla_{\mathbf{b^l}} E = \delta^l(n) \tag{2.21}$$

Using using these equations in combination with the gradient descent method, results in a model that constantly improves on the input data [21].

2.4.3. Back propagation

The back propagation algorithm works as follows [21]:

1. Set the input, $\mathbf{x}$, for the activation layer.
2. For each layer compute $\mathbf{a^l}$ and $\mathbf{z^l}$.
3. Compute the error δ^L via equation 2.18, where L is the final layer.
4. For each layer compute δ^{l-1} via equation 2.19.
5. For each layer compute $\nabla_{\mathbf{w^l}} E$ and $\nabla_{\mathbf{b^l}} E$ via equation 2.20 and equation 2.21.
6. Change the weights and biases of each neuron according to the gradient descent step.

2.4.4. Epochs

An epoch is a parameter that defines how often the back propagation algorithm will run through the training data. This is done because this way the algorithm can more efficiently learn how to tweak the weights and biases without having to create more training data. If there are too many epochs it is possible that the neural network focuses to much on the training data, this has the result that the neural network will not predict new data accurately, this is called overfitting. If there are not enough epochs, then the weights and biases will not have enough training to accurately predict new data, this is called underfitting. For every neural network the amount of epochs differs, this is why it is necessary to use a validation set to check if the network is over or underfitting [23]. This can be seen in figure 2.5.

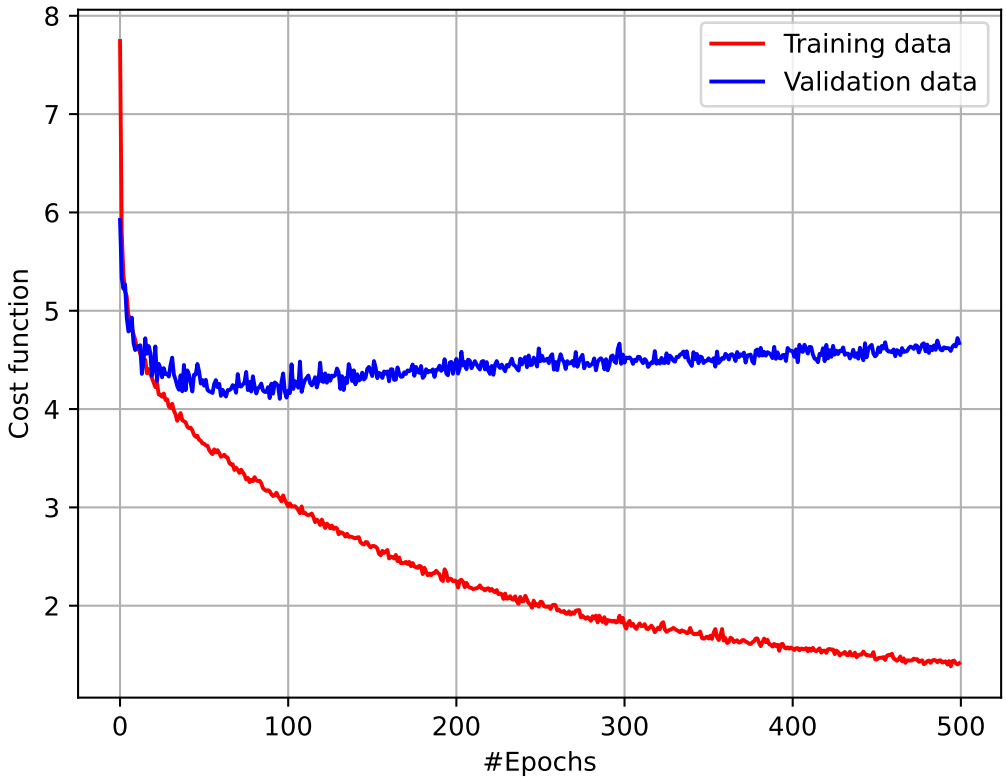

Figure 2.5: Figure of the cost function of the training an validation data as function of the amount of epochs. As can be seen in the figure, the cost function of the training data converges towards zero. However, the cost function of the validation data increases again after a certain amount of epochs.

3

SPECT

3.1. Gamma camera

A Gamma camera (also known as Anger camera) consists of several components, as depicted in figure 3.1. Three of these components will be discussed in this section, namely the collimator, scintillation crystal and the Photon Multiplier Tube (PMT). These will be discussed in chronological order when looking from the perspective of a gamma ray.

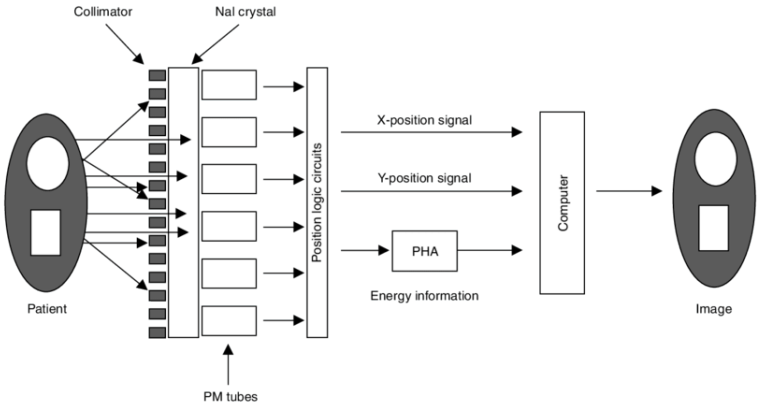

Figure 3.1: Schematic diagram of an Anger camera. Once a gamma ray goes trough the collimator, it will hit a scinitillation crystal. After that it will split up into multiple optical photons, which can interact with the PMT after which the optical photons can be detected. [24]

3.1.1. Parallel hole collimator

Parallel collimators are sheets of dense material, perforated by long thin channels. A parallel collimator does not divert rays, however it does select rays that are moving in the direction of the channels (like a tunnel). Gamma rays that are moving in different directions either collide with the collimator, in which case they will not be detected. Or they miss the collimator entirely [25]. Since photons move in straight lines, the detection of a photon thus indicates that the Gamma ray originated somewhere on a parallel line with the collimator (as can be seen in the most left part of figure 3.1), making it possible to recreate an image. Often only a small fraction of the photons move through the collimator (10^{-4}-10^{-2}), which limits the sensitivity. To change this, the width of the hole can be increased. This however has the effect that the resolution decreases [26].

3.1.2. Pinhole collimator

Pinhole imaging with Gamma rays is based on the same principle as the optical pinhole camera. The resolution of the pinhole collimator can be a lot higher than that of a parallel hole collimator, because if the detector is further away from the pinhole than the emitting object, the image of the source will be

magnified by a factor M. If the image is then magnified to its original size (using algorithms instead of optics), the resolution becomes equal to $\frac{R}{M}$, where R is the intrinsic resolution of the detector [27]. The smaller an object is, the closer it can be placed to the pinhole, and thus the greater the magnification can become, this is why pinhole collimators are often used for preclinical SPECT scans.

The difference between the pinhole collimator and parallel hole collimator can be seen in figure 3.2.

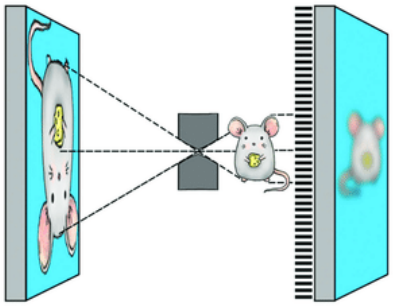

Figure 3.2: Image of a mouse with a pinhole collimator (left) and a parallel collimator (right). Note that for both cases the resolution of the image is the same, but that for the pinhole the image is also magnified. [28]

3.1.3. Scintillation crystal

A scintillator is a material with the ability to absorb ionizing radiation (for example x-rays and gamma rays) and convert the absorbed energy to light photons [29]. Such an ionizing particle dissipates its energy by ionizing and exciting the molecules of the scintillator and free electrons [30]. This will create a shell-hole in the valence band with an energetic electron. The electron in the conduction band will in turn decay back to the valence band, creating a photon [31].

Sometimes (depending on the type of scintillation material), the gap between the valence and conduction band is to big to create a light photon. In this case a small amount of impurity is added to the scintillation material creating extra levels within the forbidden gap of the conduction and valence band, called activation sites. These activation sites will have levels just above and just below the valence and conduction band, making the band gap smaller. This has the result that the energy depleted by the deexcited electron is within the light range [32]. This is schematically depicted in figure 3.3.

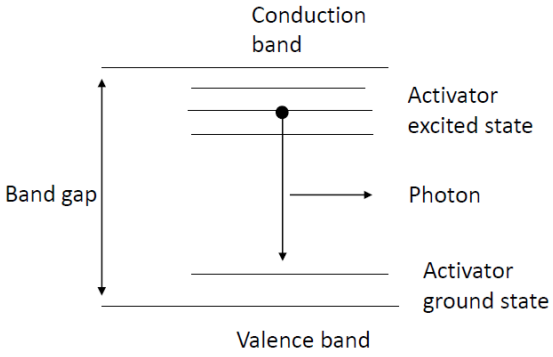

Figure 3.3: Representation of the gap between the valence band and the conduction band with activator bands in between. Activators make extra bands between the valence and conduction band. Electrons tend to migrate to the activator bands that are near the conduction band. Making it more likely that once an electron deexcites, it will send out a scintillation photon in the visible spectrum.

The most commonly used scintillation material is NaI(Tl). This is because it is cheap and has a high Gamma ray absorption coefficient for low energies [32]. However, other scintillation materials have surpassed both the resolution and the sensitivity of the NaI scintillator, but at the expense of being more costly.

Note that for a thin scintillator, the Depth Of Interaction (DOI) does not change the image of an object a lot when a pinhole collimator is used. However, when the scintillator becomes thicker, the DOI

plays a more important role as can be seen in figure 3.4. This makes the prediction of the DOI more important.

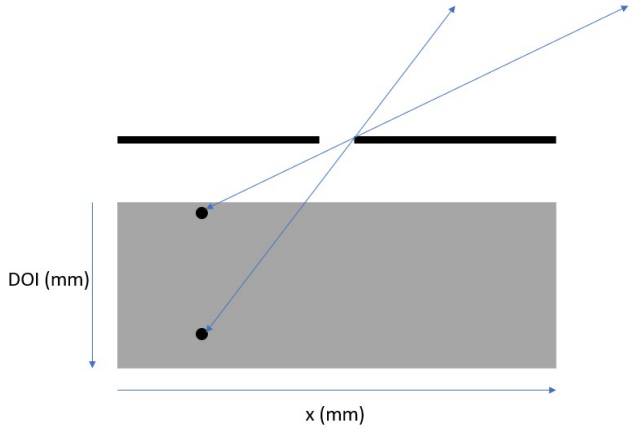

Figure 3.4: Two different Gamma photons passing trough a pinhole collimator interacting at the same (x,y)-position but at different DOIs.

3.1.4. Photon multiplier tube

The final part of the Anger camera is the PMT. A schematic setup of a PMT can be seen in figure 3.5. When a light photon collides with a photocathode, the photocathode will emit an electron. This electron will then be multiplied at the dynodes, since there are multiple dynodes this will happen several times. Finally the electrons will come in contact with the anode resulting in an electrical signal [32].

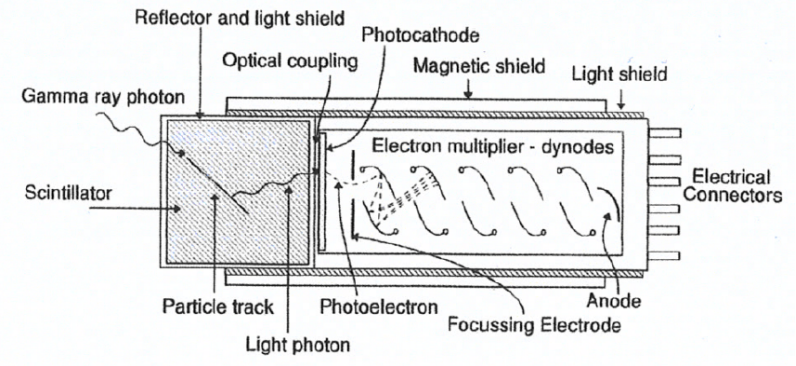

Figure 3.5: Schematic diagram of a PMT. Once a photon (within the visible light range) is emitted from the scintillator, it will be detected by the photocathode. Which will send out a photoelectron that will be sent towards a sequence of dynodes, making the pulse stronger. In the end the pulse will reach the anode where it is transformed into an electrical signal. [32]

In a Gamma camera several different PMTs are used next to each other (as also can be seen in figure 3.1). The (x,y) position is then determined by comparing the signals of the different PMTs.

3.1.5. SPECT scans

To make a medical image with a SPECT scan, a radioactive tracer is inserted in a patient, which will send out gamma rays [33].

A radioactive tracer is a radioactive element or molecule that follows a physiological or biochemical processes. Tracers often mimic a naturally occurring element or molecule in the human body [7].

A SPECT scan itself is conventionally performed with a rotating Gamma camera at a number of angels. This camera will acquire Gamma rays from the tracer after which slices are reconstructed of the required body part with a 3D image algorithm [34]. An example of a SPECT scan image is given in figure 3.6.

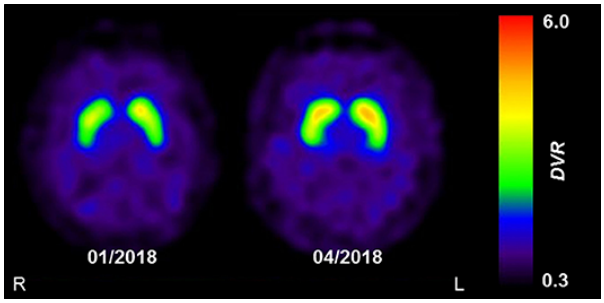

Figure 3.6: An example of a SPECT scan of dopamine concentration in the brain of a patient with Parkinson's during and after the use of herbal medicine. [35]

3.1.6. VECTor

One of the goals of this research is to increase the resolution of a Versatile Emission Computed Tomography (VECTor) system. A VECTor system is based on a specially designed cylindrical collimator containing 48 clusters of 4 pinholes each, located in an existing SPECT system. With VECTor all pinholes focus on a central scan volume (the required volume) [11]. The image can then be reconstructed by the method described by van der Have *et al.* [36]. An image of a VECTor system and its collimator can be seen in figure 3.7.

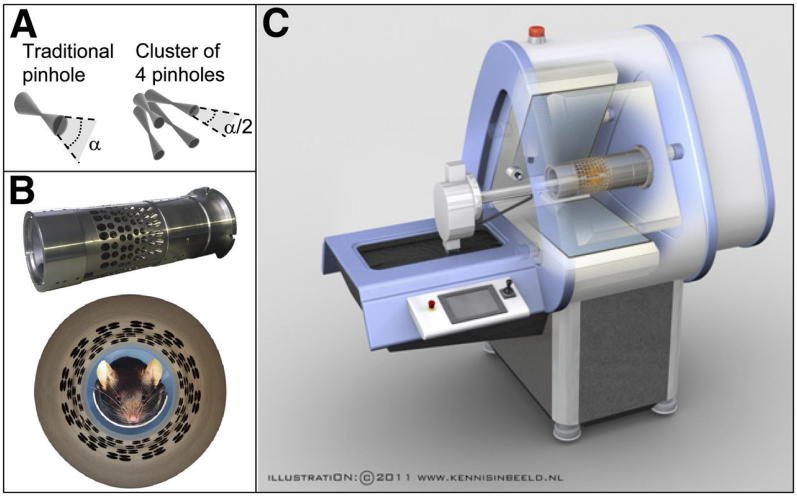

Figure 3.7: Different parts of a VECTor system. (A) Traditional pinhole and a cluster of four pinholes. (B) Clustered-pinhole collimator. (C) Collimator mounted on a SPECT/CT system. [11]

VECTor systems have the benefit of being able to detect Gammas over a broad energy range (30 KeV t0 1 MeV) at sub millimeter resolution. This makes it possible for VECTor systems to image a wide variety of therapeutic isotopes, including those used for SPECT and PET scans.

3.2. Compton scattering

A major obstacle with SPECT scans is Compton scattering [37]. Compton scattering is an elastic collision between a gamma ray and a (free) electron. This has the effect that the electron recoils with an energy T_c and angle Φ with the incident photon. Which in turn reduces the energy of the photon, and lets it scatter in a direction of θ with the incident photon [38]. The effect of Compton scatter can be seen in figure 3.8.

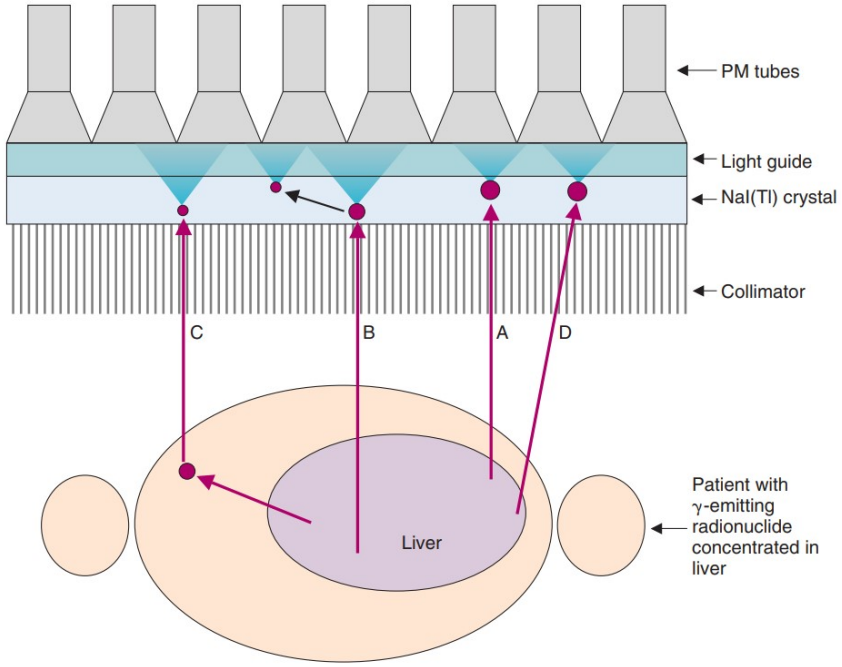

Figure 3.8: Compton scattering in a clinical SPECT scan. Here A indicates an event without Compton scatter, B is scatter inside the detector, C is scatter outside the detector and D is Septal penetration.[7]

Scattering can occur before the collimator, in the collimator and in the scintillation crystal. Energy discrimination filters some of these events, but if the the angle of scatter is less than 45 degrees this will not work anymore. In the case the Compton photon does get detected it is often many centimeters from the original site of interaction inside the scintillator [7].

An example of a multiple scatter event can be seen in figure 3.9

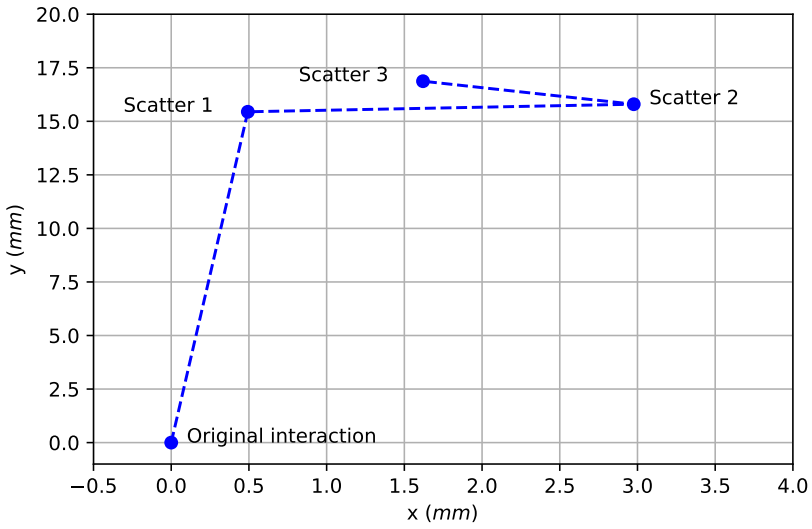

Figure 3.9: Multiple scatter events in a scintillattor. Note that scatter happens in 3 dimensions, but this figure only shows 2 dimensions.

Because Compton photons are often many centimeters form the original sit of interaction, there will be a different PMT signal compared to the same interaction without Compton scatter. This has the effect that when the image is reconstructed, there is a slight error. That is why it is necessary to determine the first interaction position of the Gamma ray.

3.3. Maximum likelihood

Maximum Likelihood Estimation (MLE) is often used for the determination of the location of Gamma rays in a scintillator [39]. The ML method is fully described in Barrett *et al.* [40] and will therefore not be fully discussed. The most important part of ML is that it determines the interaction positions by taking a weighted average, this makes it more susceptible for errors when Compton scattering occurs.

4

Method

In this chapter the method with which the neural network will be changed is discussed and which parameters will be influenced.

4.1. Materials

The data is created with a Monte-Carlo simulation from the advanced opensource software GATE [41]. This simulation is done with a NaI(Tl) scintillator of size 590mm x 470mm x 40mm (x,y,z), with a glass lightguide of size 620mm x 500mm x 4mm and a PMT area of size 620mm x 500mm x 40mm with 108, 2-inch round PMTs with a Bialkali photocathode and in a configuration as shown in figure 4.1.

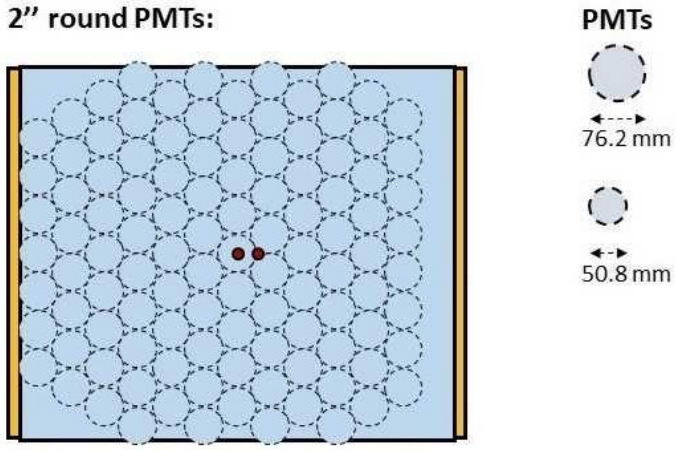

Figure 4.1: Configuration of the 2-inch round PMTs.

a schematic representation of the setup can be seen in figure 4.2.

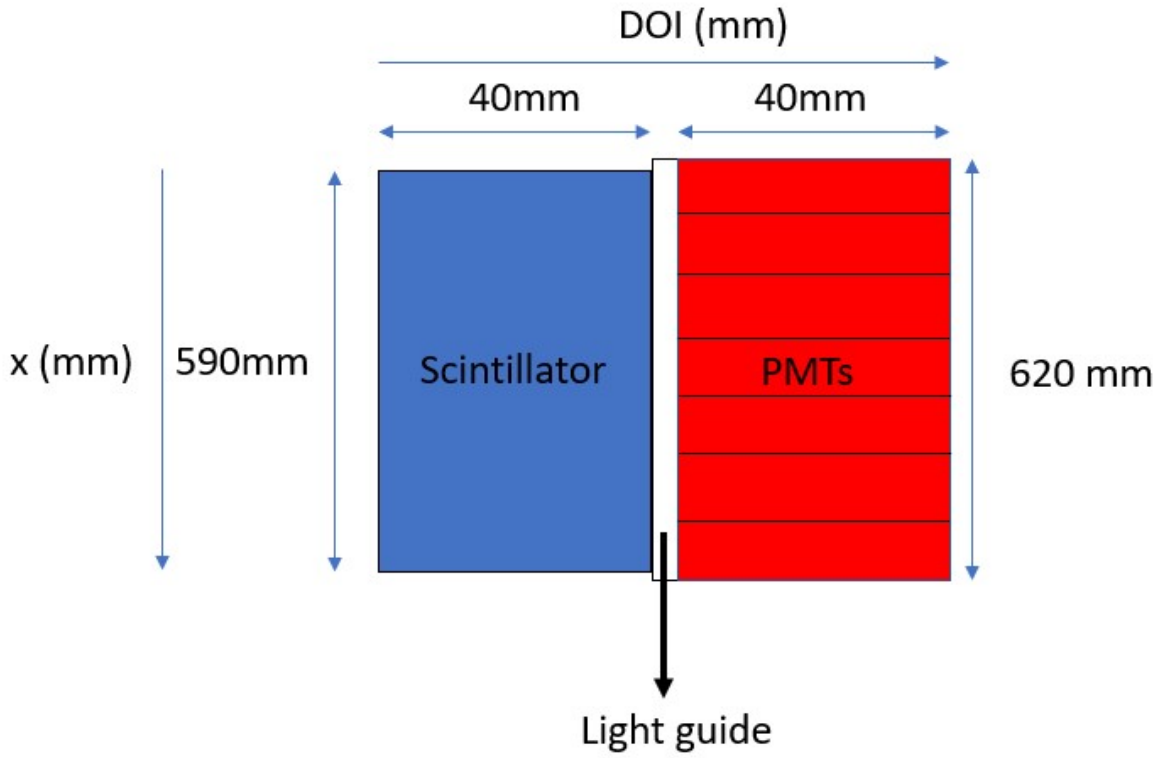

Figure 4.2: Setup of the simulation. The scintillator starts at a DOI of 0mm. The scintillator is of size 590mm x 470mm x 40mm (x,y,z), the glass lightguide is of size 620mm x 500mm x 4mm and the PMT area is of size 620mm x 500mm x 40mm

Note that the Depth Of Interaction (DOI) is measured from the scintillator, meaning that the DOI is 0mm at the boundary of the scintillator.

4.2. Data

The Monte-Carlo simulation gives data as seen in table 4.1.

Run id	Event id	Time [s]	E [MeV]	x [mm]	y [mm]	DOI [mm]	Particle id	Interaction type
0	0	0.12212	0.511	0.00	0.00	10.66	22	PhotoElectric
0	0	0.12212	0	-9.338	10.91	44.00	0	Transportation
0	0	0.12212	0	40.25	31.14	44.00	0	Transportation
0	0	0.12212	0	164.4	-66.96	44.00	0	Transportation

Table 4.1: Table with the simulated data. Here the run id indicates from which run of 1000 s the interaction is, the event id indicates from which event the interaction is, the time indicates when the interaction took place, E is how much energy is deposited in the interaction, the x is the x positioning in millimeter, the y is the y position in millimeter, the DOI is the depth of interaction in millimeter, the particle id and the interaction type indicates what for interaction it was.

Visually a simulation of a Gamma ray will look like this:

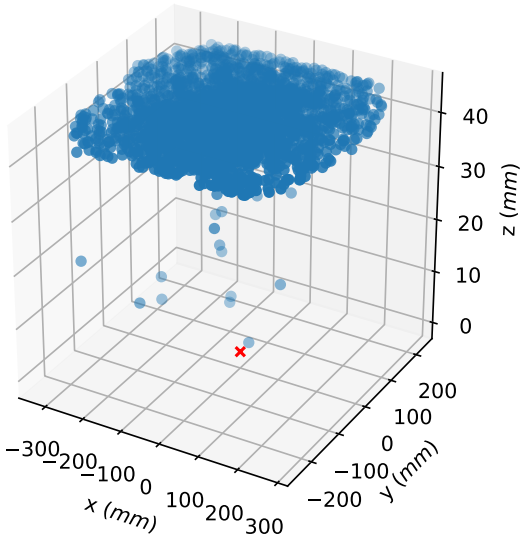

Figure 4.3: Visual representation of a simulation of a Gamma ray in a scintillator. Here the cross at the origin (0,0) represents the initial interaction in the scintillator, the dots with z-coordinate smaller then 44mm are optical adsorptions of the Gamma ray and the dots at z = 44mm are the optical photons interacting with the PMT.

The absorption of the Gamma ray in the scintillator then looks like this:

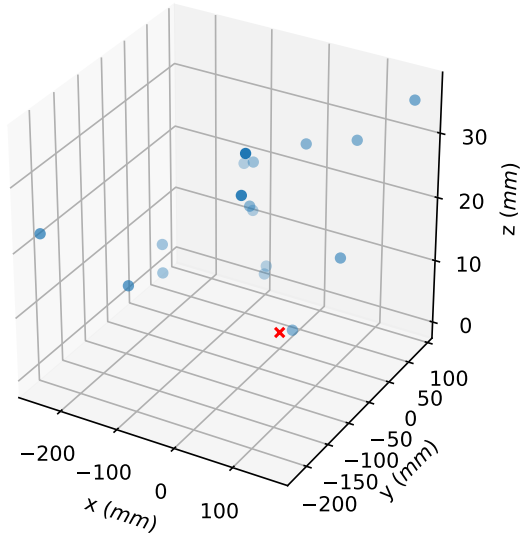

Figure 4.4: Visual representation of a Gamma ray absobed in a scintillator. Here the cross at the origin (0,0) represents the initial interaction and the dots are optical adsorptions of the Gamma ray.

Before initializing, the origin of the (x,y)-positioning (the spatial positioning) is randomized, while the z-positioning (DOI) is kept the same. This is done because the simulated data is simulated such that the gamma-ray's are centered at the origin (0,0). This is very predictable behaviour for a neural network and therefore not a good way to use the training data. The randomization is done by adding a random variable of a uniform U(-50mm,50mm) distribution to both the x and y coordinate in each event (different random variables for the x and y coordinates). This is only done for the training and validation data.

The x and y positioning for the testing data is discussed in section 4.6.

From the data format in table 4.1 the PMT signal can be created via an algorithm for each event. This

algorithm takes as input the energy of the optical photons and the (x,y,z) coordinates and determines in which PMT from figure 3.5 it would be detected. An example of a PMT signal is given in table 4.2.

Time [s]	DOI [mm]	PMT 1	PMT 2	...	PMT 107	PMT 108	E [MeV]	#interaction per event
0.12212	-10.66	7	6	…	7	3	0.511	1
0.54762	-2.378	4	9	…	9	3	0.511	2
3.42736	-0.4929	3	3	…	9	5	0.51101	4
3.52548	-33.67	0	1	…	10	6	0.511	1

Table 4.2: Table with the PMT-signal. Here the time indicates when each event took place, the DOI is the depth of interaction of the gamma-ray, the PMT's what each PMT receives as signal from the interaction and E is the energy that was deposited in the interaction.

After the creation of the PMT signal, a lot of the data will be filtered out. Primarily, because for this research only the positioning of the first interaction for each event has to be determined. First, all rows except for the first will be discarded per event. Secondly, the events that are of no interest, because the energy does not correspond with that of the energy of the Gamma ray of the tracer, are filtered out. These are the events were the energy in the PMT-signal is not in the range of $0.46MeV \leq E \leq 0.56MeV$. Thirdly, since only the (x,y,z)-positioning and the PMT-signal itself are used for the neural network all other columns are discarded.

After this the PMT-signals are sorted such that they correctly correspond to the simulation, this is done by comparing the DOI's of both data sets. Finally the PMT-signal is normalized by dividing all PMT-signals by the overall maximum of the PMT-signals.

4.2.1. Division of the data

The input data is the normalised PMT-signal of the 108 PMTs, and the labeling is the (x,y,z)-positioning of the Monte-Carlo simulation of the first event.

In the simulation 10 runs are simulated with in each run approximately 1000 events. The data is divided as follows. First the training data is set as the data where the run id is equal to 9, about one tenth of the total data. The rest of the data is set to the be divided between the training and validation set, 90% to the training set and 10% to the validation set. The division of the data thus goes as follows: training data 10%, validation data 9% and training data 81%.

4.3. Architecture of the neural network

Within the first until the second-to-last layer of the neural network, the activation function will be the leakyReLU as defined in equation 2.3. The final layer will be the identity function $f(x) = x, \forall x$. The used optimizer will be adam.

4.4. Metrics

For the results two types of metrics are used to determine the resolution, the Full Width Half Max (FWHM) and the Full Width Tenth Maximum (FWTM). Both these metrics will be discussed in this section. The full width half max (FWHM) is determined by first fitting a Gauss2 distribution, given in equation 4.1, on the empirical density function, of the errors in the x and y positions.

$$Gauss2(x) = a_1 e^{-(\frac{x-b_1}{c_1})^2} + a_2 e^{-(\frac{x-b_2}{c_2})^2} \tag{4.1}$$

Here a_1 and a_2 are the weights of the two Gaussians, b_1 and b_2 are the centres of the two Gaussians and c_1 and c_2 are the standard deviations of the two Gaussians.

After this, the maximum value of the Gauss2 distribution is determined. Then the x values for which the distribution is half the maximum are computed and the distance between these two values is calculated. This is shown in figure 4.5.

The full width tenth max (FWTM) is often used to determine what the size of the tail is of a distribution. The FWTM is determined the same way as the FWHM. However, instead of the distance of points where the the graph is half the maximum, it will be the distance of the points were the graph is a tenth of the maximum.

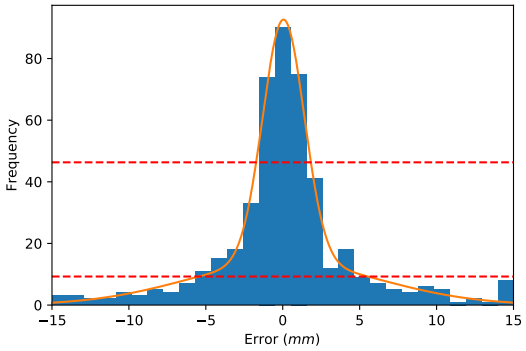

Figure 4.5: Gauss2 distribution fitted on the empirical density function in the x-direction (shown as histogram). Here the upper dotted line is half of the maximum and the lower dotted line is a tenth of the maximum.

The FWHM and FWTM are usually used to determine the resolution for SPECT scans[7]. Therefore they are useful when comparing the effectiveness of the neural network with other methods.

To determine the FWHM in the spatial resolution, the square root of the mean of the FWHM of the x-data and the FWHM of the y-data squared is taken (this is described in equation 4.2). The same is done for the FWTM.

$$FWHM = \sqrt{\frac{FWHM_x^2 + FWHM_y^2}{2}} \tag{4.2}$$

To determine the FWHM and the FWTM for the DOI, first the labeling of the testing data is divided in of 2 mm sections, ranging from 1 mm up to 39 mm. All the predicted testing data, where the labeling of the testing data is in between 9mm and 11mm, is then used for the determination of the FWHM and the FWTM. Also, instead of a Gauss2 distribution, a normal distribution (as described in equation 4.3) is used for the fitting of on the data.

$$\mathcal{N} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \tag{4.3}$$

4.5. Optimization factors

There are a lot of parameters on which a neural network might be optimized. In this section the parameters that will be optimized will be discussed.

4.5.1. Epochs

First the amount of epochs will be optimized, since it is easy to either overfit or underfit the data. This is why when training the neural network, the validation data is used to check when to stop the training. This is done by monitoring the cost function of the validation data when it is minimal. Here a patience of 40 is used, that is to say that the training stops if after 40 epochs the cost function has not become smaller than the previous minimum. When the training is stopped the weights of the epoch where the cost function was minimal will be restored.

4.5.2. Hidden layers

The amount of hidden layers is varied between 2 and 5 with integer steps to see where the neural network performs optimally.

4.5.3. Neurons

The amount of neurons is varied between 64 and 1024 where each step is a power of two. So 2^i where i is varied between 6 and 10 in integer steps.

4.5.4. Cost functions

Three different cost functions are varied to see which works optimally, namely the Mean Absolute Error (MAE), the Mean Squared Error (MSE) and the Huber loss function. These are given in equation 4.4, equation 4.5 and equation 4.6.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{4.4}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.5}$$

$$L_1 = \sum_{i=1}^{n}\begin{cases} \frac{1}{2}(Y_i - \hat{Y}_i)^2 & |Y_i - \hat{Y}_i| \leq 1 \\ |Y_i - \hat{Y}_i| - \frac{1}{2} & |Y_i - \hat{Y}_i| > 1 \end{cases} \tag{4.6}$$

Here Y_i is the predicted value and $\hat{Y}_i$ is the true value.

4.5.5. Amount of training-data

The amount of data that is trained is also varied to see which amount is optimal. Since there is only one original dataset. This is done by creating multiple different sets with a shift from the origin (as also was discussed in section 4.2). The amount of training data is varied from 4995 to 29970 in steps of 4995.

4.6. Testing

To see how effective the neural network is, the FWHM and FWTM are used as measures.

First the neural network is trained over the different amount of hidden layers and neurons and different cost functions. After this the testing-data is used to determine the spatial and DOI resolution (both FWHM). The neural networks with the the best spatial resolutions are then trained on different amounts of training data. The same is done for the neural networks with the best DOI resolutions. In this situation the FWHM and FWTM are both once again computed. In this case however, to more accurately determine them, the FWHM and FWTM are determined for Gamma rays focused on six different locations around the origin and the mean of the FWTM and FWHM is then taken.

Finally the neural networks where the FWHM and FWTM are best when varying the amount of training data, are compared with the method of Maximum Likelihood (ML) as described in Hunter [42] and Barret [40].

5 Results

In this chapter the results of the neural networks are discussed.

5.1. Spatial resolution

5.1.1. Cost functions

The spatial FWHM when different cost functions are trained with different amounts of layers and neurons, can be seen in figure 5.1.

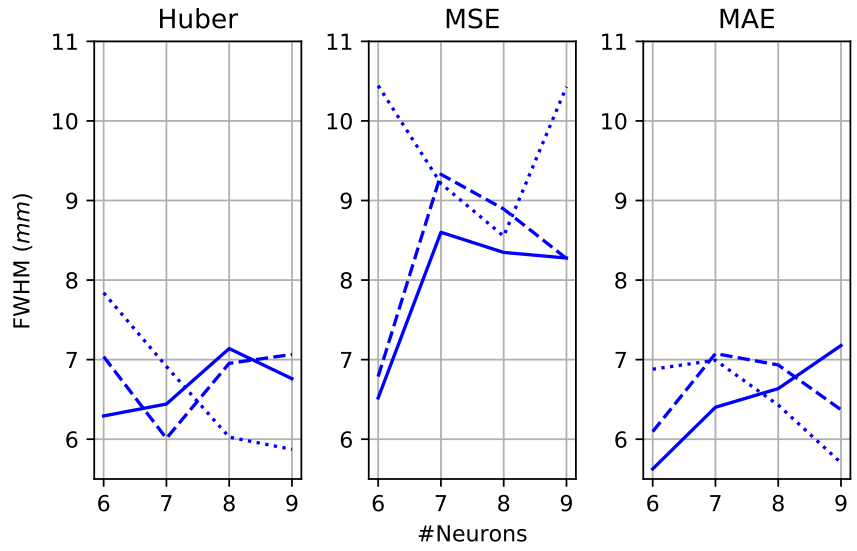

Figure 5.1: Full width half maximum of the (x,y) position as function of the amount of neurons, for different amounts of layers and different cost functions. Note that the neurons are depicted in exponents of two ($2^{\#Neuron}$). In this plot the continuous lines are two hidden layers, the dashed lines are three hidden layers and the dotted lines are four hidden layers.

From the figure it can be seen that there are three neural networks for which the FWHM is the smallest. Namely:

(a) Huber loss, 4 hidden layers and 512 neurons per hidden layer.

(b) Mean absolute error, 2 hidden layers and 64 neurons per hidden layer.

(c) Mean absolute error, 4 hidden layers and 512 neurons per hidden layer.

These neural networks will therefore be used for further testing.

5.1.2. Training data

The results for the FWHM of the spatial resolution, for the neural networks trained on different amounts of training data, can be seen in figure 5.2.

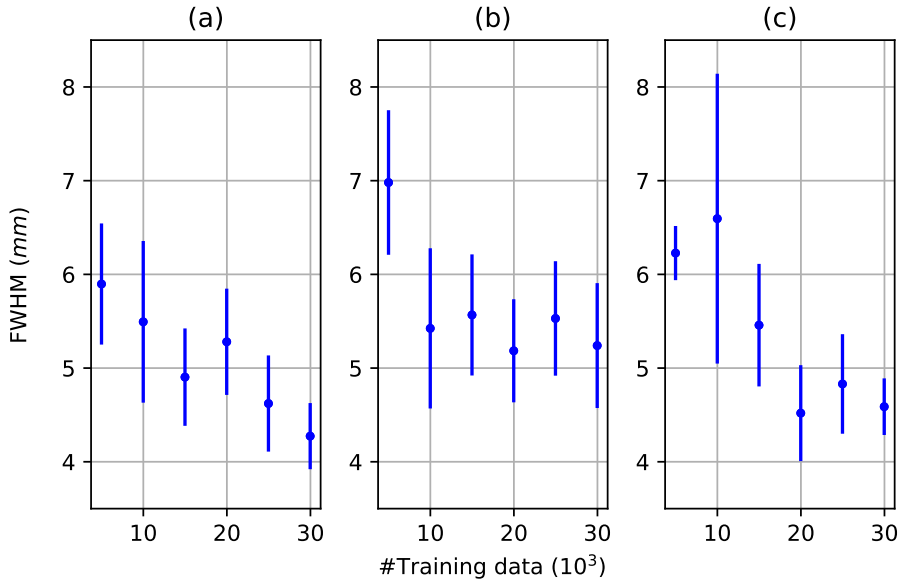

Figure 5.2: FWHM as function of the amount of data that the neural network is trained on. Here the dots is the mean of the FWHM for different spatial locations, and the error is the standard deviation of the FWHM for different spatial locations.

From this figure it appears that (a) has the best spatial FWHM resolution when trained on 29,970 datapoints (so 4,995 original data points and 24,975 augmented data points), with a FWHM of 4.27 ± 0.35mm.

Likewise the same can be done for the FWTM, this can be seen in figure 5.3.

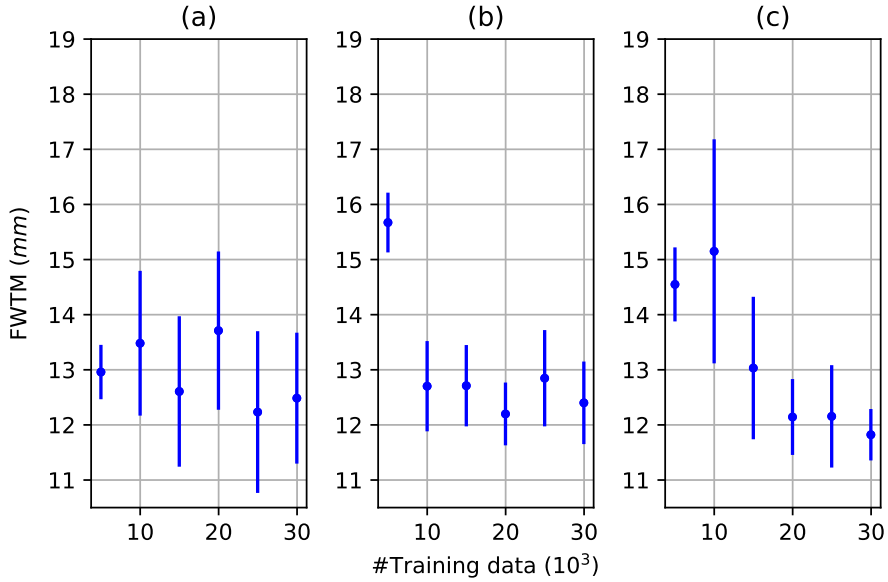

Figure 5.3: FWTM as function of the amount of data that the neural network is trained on. Here the dots is the mean of the FWTM for different spatial locations, and the error is the standard deviation of the FWTM for different spatial locations.

From this figure it appears that (c) has the best spatial FWTM resolution when trained on 19,980-29,970 datapoints. However, the difference in FWTM with (a) trained on 29,970 datapoints is less than 0.5 mm. Because this (a) has the best FWHM, (a) trained on 29,970 datapoints will be used for the comparison with the ML method. The FWTM for this neural network is equal to 12.49 ± 1.19mm.

5.1.3. Comparison with maximum likelihood

When applying the ML method on the testing data, the following 2D histogram for the spatial part of the data can be created.

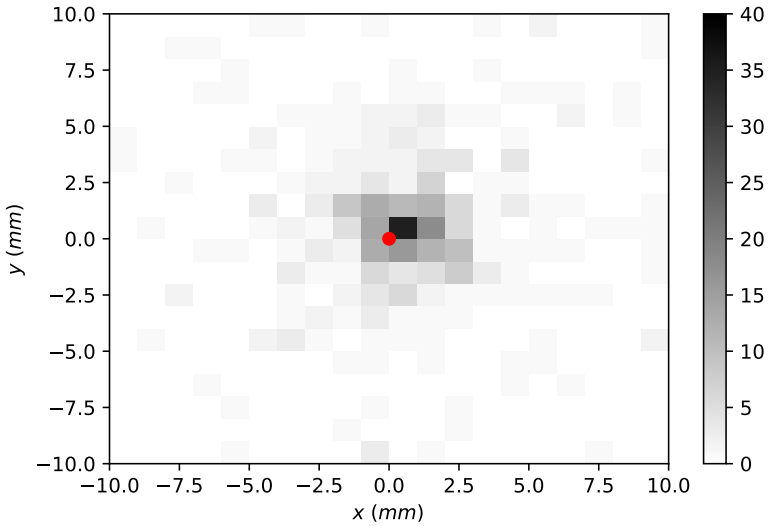

Figure 5.4: 2D histogram of the spatial prediction of the ML method. Here the red dot indicates the simulated location at the origin.

The FWHM and the FWTM of the spatial resolution are 3.31mm and 12.13mm respectively.
For the neural network the 2D histogram can be seen in figure 5.5.

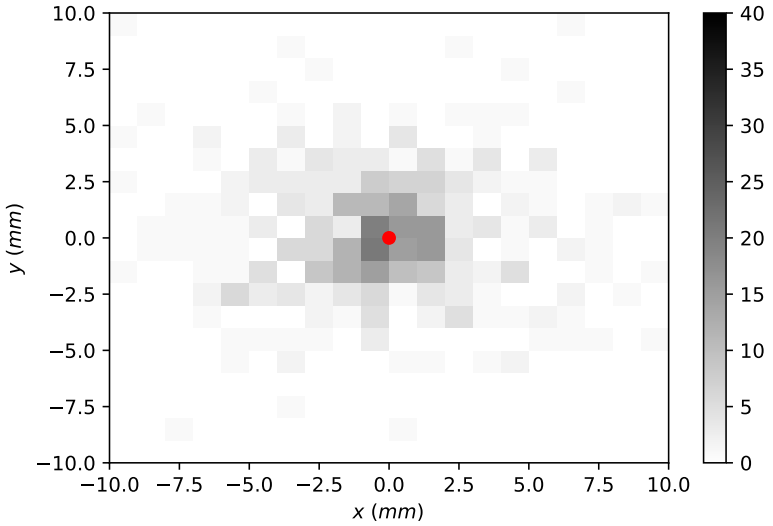

Figure 5.5: 2D histogram of the spatial prediction neural network. Here the red dot indicates the simulated location at the origin.

From figures 5.4 and 5.5 it can be seen visually that the spatial FWHM and the FWTM of the neural network are either worse, or as good as the spatial FWHM and the FWTM of the ML method. In figure 5.6 another visualisation of the errors can be seen.

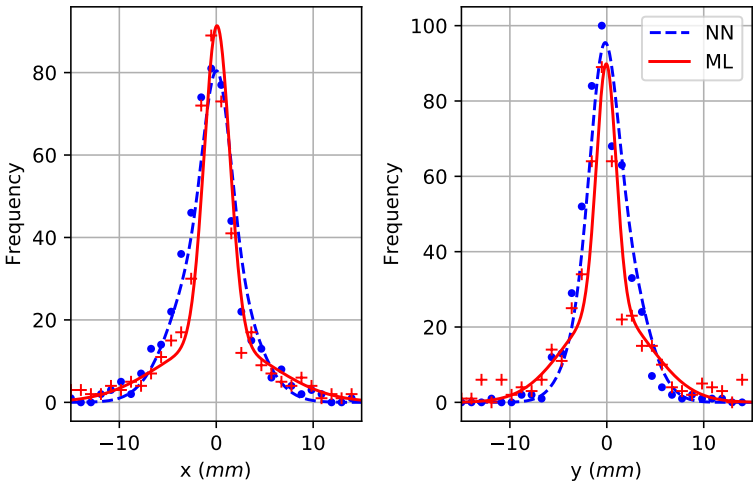

Figure 5.6: Histograms of the predicted x positioning and the predicted y positioning of both the ML method and the Neural Network (NN). Here the plusses and the continuous lines are the predictions with fit of the ML method, and the dots and the stripped lines are the predictions with fit of the neural network

From the figure it can be seen that the FWHM and the FWTM are smaller for the ML method compared to that of the neural network.

5.2. DOI resolution

5.2.1. Cost functions

As with the spatial component of the FWHM, the same can be done for the FWHM of the DOI component, the plot of the different cost functions with different amounts of layers and neurons can be seen in figure 5.7.

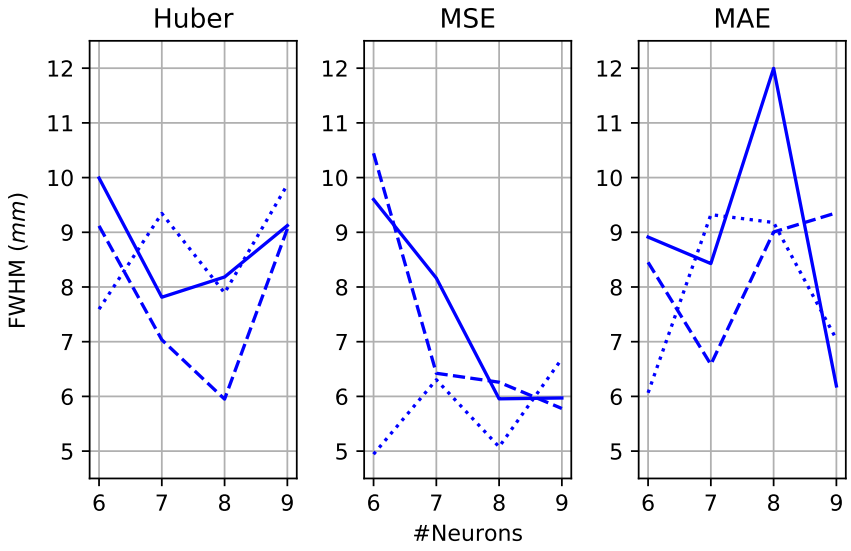

Figure 5.7: Full width half maximum of the DOI at 10mm as function of the amount of neurons, for different amounts of layers and different cost functions. Note that the neurons are depicted in exponents of two ($2^{\#Neuron}$). In this plot the continuous lines are two hidden layers, the dashed lines are three hidden layers and the dotted lines are four hidden layers.

Form the figure it can be seen that the neural network with the best DOI resolution, is the neural network with MSE as cost function, four hidden layers and either 64 or 256 neurons in the hidden layers. Because both neural networks are quite similar in structure, only the neural network with 64 neurons will be examined for further testing.

5.2.2. Training data

The results for the DOI resolution for the neural network, with the MSE as cost function, four hidden layers, 64 neurons per hidden layer and different amounts of training data, can be seen in figure 5.8.

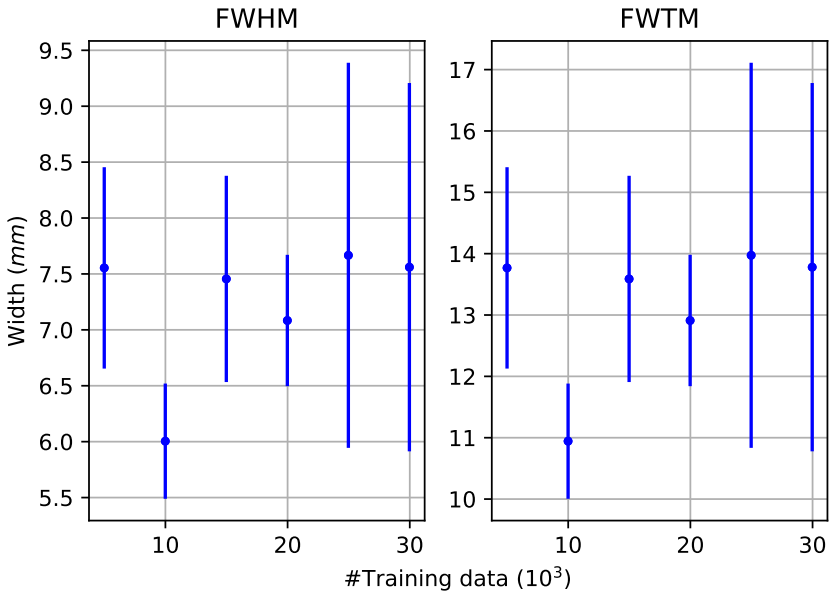

Figure 5.8: FWHM and FWTM as function of the amount of data that the neural network is trained on at a simulated DOI of 10mm.. Here the dots are the means of the FWHM and FWTM for different spatial locations, and the error is the standard deviation of the FWHM and FWTM of different spatial locations.

As can be seen from the figure, the best amount of training data for both the FWHM and the FWTM is 9,990 datapoints (so 4,995 original data points and 4,995 augmented data points). Here the FWHM is equal to 6.00 ± 0.50mm, and the FWTM is equal to 10.94 ± 0.94mm.

5.2.3. Comparison with maximum likelihood

When applying the ML method on the testing data the following error in the prediction of DOI can be visualized.

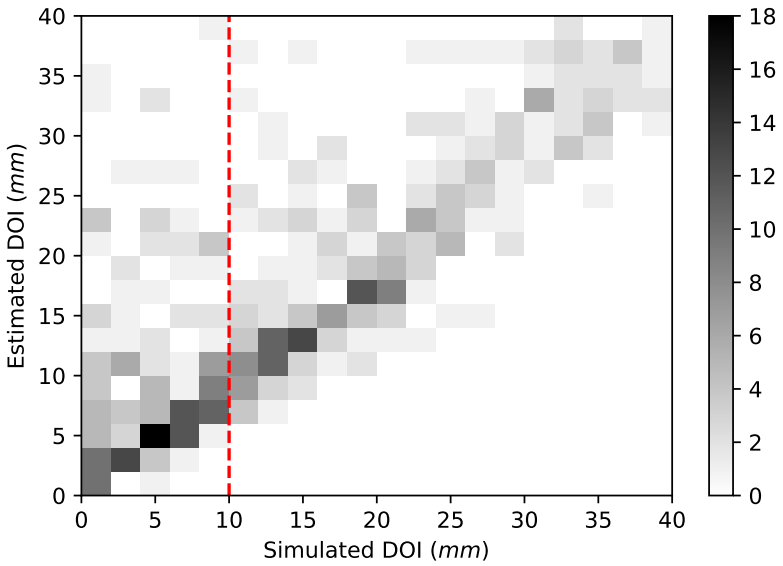

Figure 5.9: 2D histogram of the simulated DOI versus the estimated DOI when the ML method is used. Here the dotted line indicates a simulated DOI of 10mm.

For the neural the same can be done.

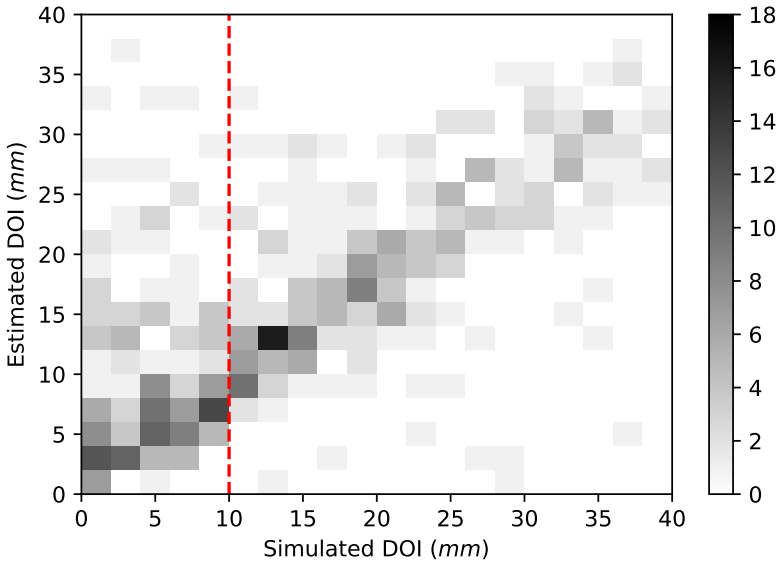

Figure 5.10: 2D histogram of the simulated DOI versus the estimated DOI when a neural network is used. Here the dotted line indicates a simulated DOI of 10mm.

As can be seen from figure 5.9 and figure 5.10, with the ML method, there is a bias towards a positive error (the estimated DOI is greater than the simulated DOI). This does not seem to be the case for the neural network.

In figure 5.11 and figure 5.12, histograms of the predicted data at a simulated DOI of 10mm can be seen for the ML method and neural network respectively.

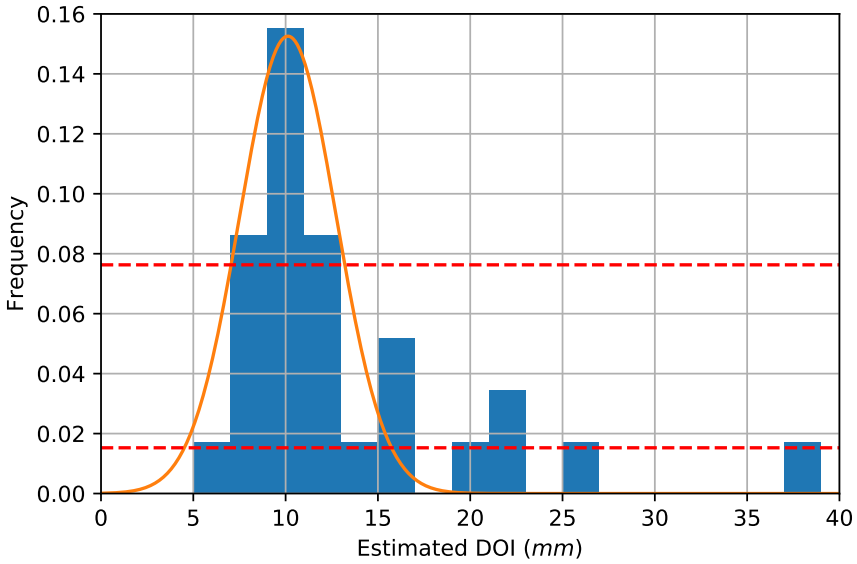

Figure 5.11: Histogam of the DOI estimated by the ML method at a simulated DOI of 10mm. Here a Gaussian distribution is fitted on the empirical density function of the prediction.The stripped lines are a half and a tenth of the maximum.

For the ML method the FWHM and the FWTM, at a simulated DOI of 10mm, are equal to 6.16 mm and 11.22 mm respectively.

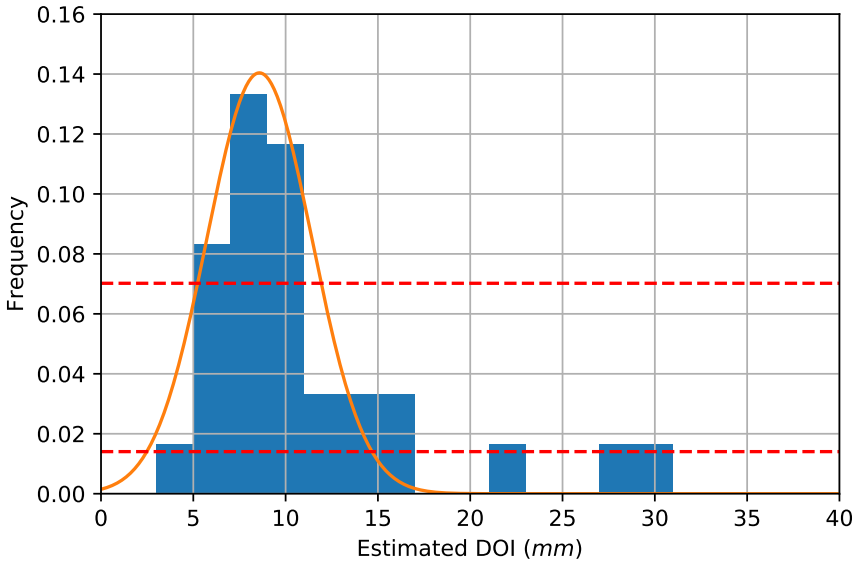

Figure 5.12: Histogam of the DOI estimated by the neural network. Here a Gaussian distribution is fitted on the empirical density function of the prediction. Here the stripped lines are a half and a tenth of the maximum

From these results it can be concluded that, when using a neural network, the resolution can be increased in the DOI, both for the FWHM and the FWTM. Note that the ML method does results in a Gaussian distribution with mean 10mm (as required). However, the mean of the prediction of the neural network is not equal to 10mm, it is 8.5mm.

5.3. Discussion

When choosing which cost function and which architecture for the neural network was best, the amount of datapoints stayed the same, namely 5,000 datapoints. This was done to reduce computational time. However, the different cost functions and architectures do not always respond the same to different amounts of training data as can also be seen in figure 5.2 and figure 5.3. This has the consequence that a neural network which was not chosen for the differing amounts of training data, might perform better than the neural networks used now.

Also, the training data used when training the neural networks to different amounts is not independent of each other since the total spatial position is shifted with a random variable. This has first of all the consequence that the DOI of the simulation does not change. But it also has the consequence that the spatial location between each optical photon does not change, only the mean of all these photons. This can have a negative effect on the neural network itself, making the chance of overfitting greater.

For the determination of the DOI only a small amount of data could be used, 29 datapoins. Note that there are only 29 datapoints because, first of all, the testing data consisted out of 510 datapoints and the datapoints in the range of 9mm and 11mm are thus even less than that (this slice is only one-twentieth of the total width of the scintillator,).Since only this amount of data was within the required range. 29 datapoints are usually not enough to make any accurate assumptions.

The architecture of the neural network itself was not very flexible. For each hidden layer the amount of neurons was the same and the activation function was also always the same. Making this more flexible could have the potential of a better spatial or DOI prediction, since then the mapping between the PMT as input, and the coordinates can become more complex.

The neural network was trained to estimate the first interaction position. If it would be trained to estimate the weighted average (as with ML) of all optical photons, it could be possible that the resolution increases, because the PMT signal is based on the optical photons.

Finally since each neural network had three outputs, (x,y,z), and the spatial resolution is worse than with ML, but the DOI resolution is better. It could be that when the neural network is split in two parts, one part that predicts the spatial coordinates and one part that predicts the DOI coordinates, it would give better results.

6

Conclusion

From the determination of the spatial resolution, it can be concluded that the neural network with Huber loss function as cost function, with 4 hidden layers and 512 neurons per hidden layer, trained on 30,000 datapoints has the best overall resolution. The FWHM and FWTM are 4.27 ± 0.35 mm and 12.49 ± 1.19 mm respectively. When using the maximum likelihood method, the FWHM and FWTM were equal to 3.31 mm and 12.13 mm respectively. Therefore the spatial resolution of the neural network is lower than that of the maximum likelihood .

From the determination of the DOI resolution, it can be concluded that the neural network with mean squared error as cost function, with 4 hidden layers and 64 neurons per hidden layer, trained on 10,000 datapoints has the best overall resolution. The FWHM and FWTM are 6.00 ± 0.50 mm and 10.99 ± 0.94 mm respectively. When using the maximum likelihood method, the FWHM and FWTM were equal to 6.16 mm and 11.22 mm respectively. Therefore the DOI resolution of the neural network is higher than that of the maximum likelihood. Note however, that the fitted Gaussian has as mean 10 mm for the maximum likelihood method, as expected. But is lower than 10 mm for the neural network.

As follow up research a lot can be done. Firstly, it would be interesting to see what the resolution would become if the neural network was trained on independent training data. Secondly, a split in neural network (one DOI part and one spatial part), could result in either better or worse results. Thirdly, a more complex architecture with different activation functions and differing amounts of neurons in each hidden layer would be interesting to look at, because then the mapping of the neural network would become even less linear. Finally, perhaps convolutional neural network, based on a convolution of the PMT image ,would work better than the analytical neural network.

Bibliography

[1] James M Warwick. “Imaging of brain function using SPECT”. In: *Metabolic brain disease* 19.1 (2004), pp. 113–123.

[2] Steven T DeKosky et al. “Assessing utility of single photon emission computed tomography (SPECT) scan in Alzheimer disease: correlation with cognitive severity.” In: *Alzheimer disease and associated disorders* 4.1 (1990), pp. 14–23.

[3] Giuliano Mariani et al. “A review on the clinical uses of SPECT/CT”. In: *European journal of nuclear medicine and molecular imaging* 37.10 (2010), pp. 1959–1985.

[4] Bin He and Eric C Frey. “Comparison of conventional, model-based quantitative planar, and quantitative SPECT image processing methods for organ activity estimation using In-111 agents”. In: *Physics in Medicine & Biology* 51.16 (2006), p. 3967.

[5] Albert Flotats and Ignasi Carrió. “Cardiac neurotransmission SPECT imaging”. In: *Journal of nuclear cardiology* 11.5 (2004), pp. 587–602.

[6] Benjamin L Franc et al. “Small-animal SPECT and SPECT/CT: important tools for preclinical investigation”. In: *Journal of nuclear medicine* 49.10 (2008), pp. 1651–1663.

[7] Simon R Cherry, James A Sorenson, and Michael E Phelps. *Physics in nuclear medicine e-Book*. Elsevier Health Sciences, 2012.

[8] Michael Feld and Michel De Roo. *History of nuclear medicine in Europe*. Schattauer Verlag, 2003.

[9] Arman Rahmim and Habib Zaidi. “PET versus SPECT: strengths, limitations and challenges”. In: *Nuclear medicine communications* 29.3 (2008), pp. 193–207.

[10] Keiichi Magota et al. “Performance characterization of the Inveon preclinical small-animal PET/SPECT/CT system for multimodality imaging”. In: *European journal of nuclear medicine and molecular imaging* 38.4 (2011), pp. 742–752.

[11] Marlies C Goorden et al. “VECTor: a preclinical imaging system for simultaneous submillimeter SPECT and PET”. In: *Journal of Nuclear Medicine* 54.2 (2013), pp. 306–312.

[12] Milan Decuyper et al. “Artificial neural networks for positioning of gamma interactions in monolithic PET detectors”. In: *Physics in Medicine & Biology* 66.7 (2021), p. 075001.

[13] Beien Wang et al. “Novel light-guide-PMT geometries to reduce dead edges of a scintillation camera”. In: *Physica Medica* 48 (2018), pp. 84–90.

[14] Jeremy Howard and Sylvain Gugger. *Deep Learning for Coders with fastai and PyTorch*. O’Reilly Media, 2020.

[15] Konstantin Kravtsov et al. “Ultrafast all-optical implementation of a leaky integrate-and-fire neuron”. In: *Optics express* 19.3 (2011), pp. 2133–2147.

[16] Alan Woodruff. *What is a neuron?* Aug. 2019. URL: `https://qbi.uq.edu.au/brain/brain-anatomy/what-neuron`.

[17] Jinming Zou, Yi Han, and Sung-Sau So. “Overview of artificial neural networks”. In: *Artificial Neural Networks* (2008), pp. 14–22.

[18] Chris M Bishop. “Neural networks and their applications”. In: *Review of scientific instruments* 65.6 (1994), pp. 1803–1832.

[19] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. “Activation functions in neural networks”. In: *towards data science* 6.12 (2017), pp. 310–316.

[20] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep sparse rectifier neural networks”. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 315–323.

[21] Michael A. Nielsen. *Neural networks and deep learning*. Determination Press, 2015.

[22] Daniel A Zahner and Evangelia Micheli-Tzanakou. “Artificial neural networks: definitions, methods, applications”. In: *Supervised and unsupervised pattern recognition: Feature extraction and computational intelligence*. CRC Press, 2017, pp. 61–78.

[23] Jason Brownlee. “What is the Difference Between a Batch and an Epoch in a Neural Network”. In: *Machine Learning Mastery* 20 (2018).

[24] Martin Brunner and Oliver Langer. “Microdialysis versus other techniques for the clinical assessment of in vivo tissue drug distribution”. In: *The AAPS journal* 8.2 (2006), E263–E271.

[25] Miles N Wernick and John N Aarsvold. *Emission tomography: the fundamentals of PET and SPECT*. Elsevier, 2004.

[26] Karen Van Audenhaege et al. “Review of SPECT collimator selection, optimization, and fabrication for clinical and preclinical imaging”. In: *Medical physics* 42.8 (2015), pp. 4796–4813.

[27] Freek Beekman and Frans van der Have. “The pinhole: gateway to ultra-high-resolution three-dimensional radionuclide imaging”. In: *European journal of nuclear medicine and molecular imaging* 34.2 (2007), pp. 151–161.

[28] Eric Meester et al. “Perspectives on Small Animal Radionuclide Imaging; Considerations and Advances in Atherosclerosis”. In: *Frontiers in Medicine* 6 (Mar. 2019), p. 39. DOI: 10.3389/fmed.2019.00039.

[29] Charles L Melcher. “Scintillation crystals for PET”. In: *Journal of Nuclear Medicine* 41.6 (2000), pp. 1051–1055.

[30] R Gwin and RB Murray. “Scintillation process in CsI (Tl). I. Comparison with activator saturation model”. In: *Physical Review* 131.2 (1963), p. 501.

[31] MJ Weber. “Scintillation: mechanisms and new crystals”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 527.1-2 (2004), pp. 9–14.

[32] Gordon R. Gilmore and J.D. Hemingway. *Practical Gamma-ray Spectrometry*. first. John Wiley & Sons Ltd, 1995.

[33] Dominique Delbeke et al. “Procedure guideline for SPECT/CT imaging 1.0”. In: *Journal of Nuclear Medicine* 47.7 (2006), pp. 1227–1234.

[34] RR Fulton et al. “Use of 3D reconstruction to correct for patient motion in SPECT”. In: *Physics in Medicine & Biology* 39.3 (1994), p. 563.

[35] Michel Rijntjes and Philipp T Meyer. “No free lunch with herbal preparations: lessons from a case of parkinsonism and depression due to herbal medicine containing reserpine”. In: *Frontiers in Neurology* 10 (2019), p. 634.

[36] Frans van der Have et al. “System calibration and statistical image reconstruction for ultra-high resolution stationary pinhole SPECT”. In: *IEEE transactions on medical imaging* 27.7 (2008), pp. 960–971.

[37] Carey E Floyd et al. “Deconvolution of Compton scatter in SPECT”. In: *Journal of Nuclear Medicine* 26.4 (1985), pp. 403–408.

[38] John Betteley Birks. *The theory and practice of scintillation counting: International series of monographs in electronics and instrumentation*. Vol. 27. Elsevier, 2013.

[39] Carey E Floyd Jr et al. “Improved SPECT reconstruction using inverse Monte Carlo”. In: *Physics and Engineering of Computerized Multidimensional Imaging and Processing*. Vol. 671. SPIE. 1986, pp. 206–213.

[40] Harrison H Barrett et al. “Maximum-likelihood methods for processing signals from gamma-ray detectors”. In: *IEEE transactions on nuclear science* 56.3 (2009), pp. 725–735.

[41] Open GATE collaboration. *GATE*. URL: http://www.opengatecollaboration.org/.

[42] William CJ Hunter, Harrison H Barrett, and Lars R Furenlid. “Calibration method for ML estimation of 3D interaction position in a thick gamma-ray detector”. In: *IEEE transactions on nuclear science* 56.1 (2009), pp. 189–196.