Machine Learning project: In-crystal Gamma-interaction localization for positron emission tomography (PET) from Cherenkov photons

Lucas Brunschwig, Ester Simkova, Arthur Valentin

Supervisors: Dr. Pol del Aguila Pla (EPFL-LIB, CIBM-SP), Sofiia Forostenko (ETH Zürich, D-PHYS, IPA)
School of Life Sciences, EPFL Lausanne, Switzerland

***Abstract*—Positron emission tomography (PET) is an imaging technique used in the clinic, most commonly to detect tumors inside a patient's body. After a radio-tracer is injected, it accumulates at certain locations inside the body, e.g., tumors. In the radio-tracer, positron-electron annihilations occur. This causes the emission of 2 γ rays that pass through the body of the patient to reach crystal detectors that surround the patient's body. Current PET machines use detectors that detect scintillation photons produced as a result of the interaction between the γ ray and the crystal detector.**

EPFL, in collaboration with ETHZ, is currently developing a new detector using a different crystal type that detects another kind of photons produced after a γ ray-crystal interaction; Cherenkov photons. The aim is to improve the time resolution of the machine and thus the spatial resolution of the imaging technique. From the position and time of the Cherenkov photons detected, it is needed to recover the position and time when the γ rays interacted with the crystal to be able to predict the annihilation location in the patient's body using classical physics.

In this paper, we show that we managed to improve the prediction resolution of this γ ray-crystal interaction position and time from Cherenkov photons detected using a multilayer perceptron (MLP) instead of the clinically used center of gravity (CoG) method, replicating the results of the article from F. Hashimoto et al. [3].

I. Introduction

Positron emission tomography (PET) is a clinical functional imaging system based on the injection of a radioactive tracer into the patient's body [1]. For example, the most common use of a PET scan is to detect tumor metastases in the human body, when a radiotracer called FDG (18-fluorodeoxyglucose) is injected into a patient. This radio-tracer, being a structural analog of glucose, is incorporated by the organs that need to uptake a high amount of glucose to function (the brain, the heart, the liver and the kidneys) but is also incorporated by tumors because they need a lot of glucose to grow. The tumors are thus well visible on a PET scan as be seen in Figure 1.

The functioning principle of PET is the following: a beta emitter is injected into a patient, accumulates at certain places in the body according to its properties, and emits positrons (the anti-matter form of electrons) while undergoing radioactive decay. The positrons emitted will annihilate electrons inside the patient's body. When this happens, 2 γ rays are emitted with energy 511 keV in two opposite directions, at an angle of approximately 180 degrees. These γ rays then travel through the body of the patient until they reach a detector conventionally composed of several rectangular cuboid crystals. In the detector, the γ rays interact with the crystal and these interactions produce electrons that in turn produce optical photons. The goal of the detector is to detect the time of arrival of these optical photons. The classical detector consists of 1 pixel covering the whole crystal which sends the information of how many photons were detected at a certain time.

Depending on the crystal type and its physical properties, the optical photons created can be of at least two types: scintillation photons or Cherenkov photons. The classical detectors detect scintillation photons, that are not emitted promptly (for example with a rise time (decay time constant) of 300 ns in the BGO crystal) after the γ ray-crystal interaction, which is too much if we want to go below 100 ps in time resolution. Since the spatial resolution is defined by the time stamps of the events, spread in emission time cause uncertainty in $gamma$-interaction position. Indeed, according to the geometry of the problem, we obtain the formula:

$$\delta x = \frac{\sqrt{2}}{2} \times c \times \delta t$$

with δx being the position resolution, c the speed of light and δt the time resolution.

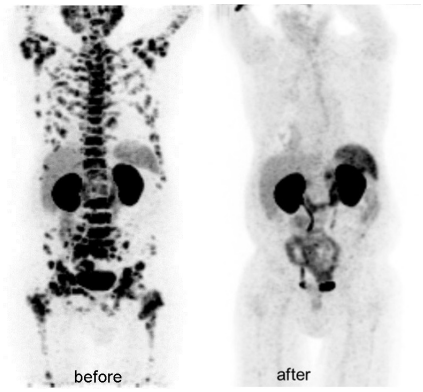

Fig. 1. The PET scan of a prostate-cancer patient before and after his treatment. Thanks to the PET scan, we can see that the metastases, represented by darker areas, are everywhere in the body before the treatment and that they shrunk completely after the treatment [2]. © EU, 2016

EPFL, in collaboration with ETHZ, has been recently developing a new kind of detector which detects Cherenkov photons, consisting of a PbF2 crystal where only Cherenkov and not scintillation photons can be produced. These photons are emitted promptly after the γ ray-crystal interaction and thus the resolution of the PET scanner can be greatly improved. In our case, we also try to estimate the position of the optical photons detected in the crystal by using many pixels per crystal, to try to reconstruct more precisely from it where the initial positron-electron annihilation took place in the body. However, the problem with Cherenkov photons is the small number of photons emitted per γ ray-crystal interaction (about 5 emitted on average, about 3.5 that reach the detector and even less that are actually detected), so it is harder to estimate the original γ ray-crystal interaction position.

Having the position and time of arrival of the optical photons detected by the detector, our goal in this project was to find a Machine Learning model to determine the location and timing of the original γ rays - crystal interaction. Indeed, the phenomena that happen in the crystal are very complex and challenging to model mathematically (the simulation was only achieved through Monte-Carlo methods), so a Machine Learning model is really useful here. Because of the strong non-linearity observed in the physical phenomena, neural networks with non-linear activation functions seemed to be the best choice here.

From the γ rays - crystal interactions, it is then quite straightforward to find the positron-electron annihilation position in the patient using

classical results.

The training data were generated using a Monte Carlo simulation of the phenomena, so their amount was high (10^7), even if not enough in our case, as discussed below.

II. Models and methods

A. The reference article

As a starting point in the project, we read a paper on the 3D interaction position estimation using a deep neural network in Cherenkov-based detectors by Hashimoto et al. [3] and tried to reproduce the results obtained with the neural network they trained, like in our case, on data obtained from Monte Carlo simulations.

B. Data loading, exploring and pre-processing

The input data consisted of a collection of events. For each event there are:

1) the position from where the γ ray originates in [mm], which is always (0, 0, 0) in our case
2) the time and position (x,y,z) in [mm] of interaction for each crystal
3) the position of origin in [mm] and creation time of Cherenkov photons in [ns]
4) the position of detection in [mm] and time of Cherenkov photons at the detector in [ns]

Then, for detected photons there is also an indication of their stopping process, the time and energy that we don't use explicitly. The crystal we have in our case has dimensions $25 \times 25 \times 25mm^3$, so it is a cubic crystal. The data from two crystals were given.

These data were obtained using Monte-Carlo simulations of the phenomenon and packed into a root format (https://root.cern/) by our supervisor Sofiia Forostenko. Our main supervisor, Dr. Pol del Aguila Pla, provided us with a function to load the data into a Python format.

We decided to create three different data set sizes; a first one of 10^4 events to quickly assess our data, a second one of 10^6 to train subsequent models with a significant amount of data and a last one with the total number of events (10^7). In order to sort these events, we created a Python dictionary to classify the events by the number of Cherenkov photons detected. This number ranged from 0 to 32 for each crystal. We only kept the events where the photon number detected was 5 or above, considering that below 5 photons detected, it was not really indicative of an event happening. In Table I are shown different statistics from the 10 million data points.

Statistics	crystal 1	crystal 2
mean number of Cherenkov photons emitted with (standard deviation)	5.145 (6.011)	5.144 (6.010)
mean number of photons reaching the detector with (standard deviation)	3.470 (4.263)	3.470 (4.264)
min number of photons	0	0
max number of photons	32	32

TABLE I
STATISTICS ON THE WHOLE DATA SET, ROUNDED UP TO 4 DIGITS

In the data we were provided, there were optical photons detected in all 5 sides of the 3-dimensional rectangular detector. However, we first focused on the Cherenkov photons detected in only one of the sides: the face at the back of the detector, to agree with what was done in the article [3]. We thus only selected the Cherenkov photons detected which had a $-85mm$ z coordinate for the first detector and $+85mm$ for the second detector (this corresponds to the back faces of the two detectors), with a tolerance of +/- 10^{-5}, in accordance with the data magnitude. However, we only used the data from the first detector to be in accordance with the article [3].

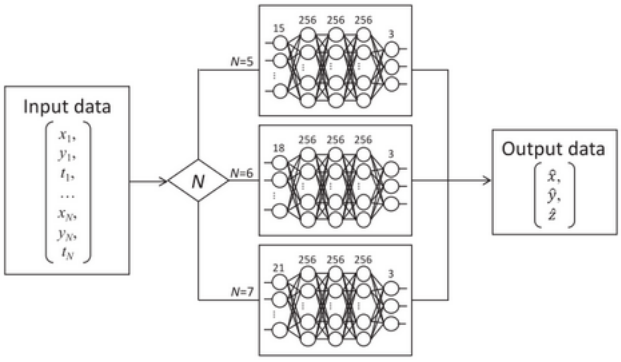

Fig. 2. The diagram of the neural network employed in the paper by Hashimoto et al, which changed according to the input photon number per crystal [3].

C. Multilayer Perceptron (MLP)

In the paper [3], the best neural network architecture they found to fit this case was a fully-connected multilayer perceptron (MLP) with 3 hidden layers, each with 256 units. All hidden layers consisted of a linear transformation followed by batch normalisation and a ReLU activation function. They then transformed the final hidden layer into a 3-dimensional γ ray-crystal interaction position (x, y, z). In our case, the final interaction position also had a fourth component which is the time (t) at which the interaction took place. The output data were thus in 4 dimensions. The loss function used was the mean square error (MSE), the optimiser was Adam and they ran the neural network for 200 epochs. The learning rate used was 10^{-4}. Regularisation was employed to prevent over-fitting of the neural network with a weight decay of 10^{-4}. 90% of the data were used to train the neural network and 10% to test it (as a validation data set). For the data pre-processing, the input data were standardized by removing the mean and dividing by the standard deviation. Moreover, what was striking in the paper is that they used a different neural network according to the number of photons detected per event (as we can see in Figure 2) and not just one general model. In our case, the number of photons detected per event ranged from 0 to 32, so we had to train a different neural network according to the photon number detected per event.

We used PyTorch to re-create the multilayer perceptron as described in the article using the "torch" and "torch.nn" libraries. We used the free GPU provided by Google Collab to train faster.

We first started by using 10^4 data points, then 10^6 data points and finally, the whole data-set (10^7 data points) to train the neural network.

In our model, the batch normalisation layers decreased the performance and increased the training time, so we removed them. Moreover, the number of data points used in the article [3] was a lot bigger than ours (about 100 times), thus, we had to increase the number of epochs to train our MLP from 200 epochs to 10'000 for 10^7 data points to get a good performance.

D. Center of gravity (CoG)

Then, we compared the neural network performance to the performance of the standard clinic method used in PET which is CoG (center of gravity). This method consists of finding the means of the x and y positions of all optical photons detected per event to find the original γ ray-crystal interaction position. Indeed, the x and y coordinates of the interaction position are then said to be equal to the obtained means:

$$x_{\mathrm{CoG}} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$y_{\mathrm{CoG}} = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

However, the CoG doesn't allow us to estimate the z coordinate, nor the time of the γ ray-crystal interaction. For that, we replicated an other method, PCA as used in [3].

E. PCA

To do the principal component analysis (PCA) to find the z coordinate, we followed the method done in [3]. We first had to fit an ellipse to the x-y plane of the data points. In this purpose, we first centered the 2-dimensional x-y data by removing the mean. Then, we looked at the covariance matrix, found its eigenvectors and took the one with the highest eigenvalue to calculate the rotation angle. This angle was then used to rotate all the data points that have been standardised. Another covariance matrix is then calculated. The only non-zeroes values left are the two variances on the diagonal. These two variances correspond to the axes of the ellipse. The bigger standard deviation makes up the major axis of the ellipse and the smaller one, the minor axis.

The next step was to find the 7features needed for the PCA:

1) the rotation angle of the ellipse
2) the length of the minor and major axis of the ellipse
3) the time difference between the first and last photon
4) mean, maximum and minimum distance between 2 photons

Then, the true z coordinates were extracted from the data. We split the dataset composed of the parameters and the true z coordinates into a training set and validation set with ratio 0.9. We standardised the resulting training and test sets (test set was standardised with the mean and standard deviation of the train set). Afterwards, we fit the PCA on the training set and retained the first principal component (as done in [3]). A linear regressor was finally trained on this principal component to obtain the predictions for the z coordinate.

F. Time estimation

After having found the position predictions in x, y and z using CoG and PCA, we aimed at finding the time prediction using classical physics. We assumed that the photons travel in straight paths between the crystal interaction position and the detector, and that they travel at the speed of light divided by the refractive index of the crystal ($v = \frac{c}{n}$) where n can be approximated to 1.8 in our case. We know that their paths are not straight lines in reality since complex non-linear phenomena happen in the crystal, but we used this approximation as a baseline to compare the results with the ones obtained with the neural network. The formula used was:

$$\delta t = \frac{\delta x}{v}$$

with δx being the position difference from the interaction position in the crystal to the position of the detected photon and v the speed of photons in the crystal.

The time estimation was found for each photon of each event and the average per event was taken as predictions.

G. Going beyond

Next, we went beyond what was done in the article [3] to stray further away from the ideal situation and to model more closely what could happen in a real situation, with noise and a finite pixel number per crystal.

First, we modeled the physical time resolution of the detector by adding some noise to the time coordinates of the data. The FWHM was chosen to be of $30ps$ in accordance with the current detector performance. The full width at half maximum (FWHM) of the distribution is related to the standard deviation (std) of the Gaussian distribution used to smear the time data by the formula:

$$\text{FWHM} = 2.355 \times std$$

The noise was thus taken as a random number from a Gaussian distribution of mean 0 and standard deviation $\frac{\text{FWHM}}{2.355} = \frac{3\times10^{-2}}{2.355} ns$ in the desired case and then added to the time coordinates.

Finally, we model the spatial resolution of the detector by modeling a detector face as a certain finite number of pixels. In order to do so, we discretized faces along each of their direction with 2^n intervals. We tried with $n = 1, 2, 3$. For example, if we have 2^2 interval along the x-axis which varies from -12.5 mm to 12.5 mm, the center of the pixels would be (-7.5, -2.5, 2.5, 7.5) mm. Then, for each Cherenkov photon, we associated each coordinate to the closest center of the corresponding axis.

III. Results

A. The ideal situation

1) Statistics: We started by considering the ideal situation, like in the article [3], with infinite time and spatial resolutions of the detector. We trained the model on all three data sets mentioned above (10^4, 10^6 and 10^7 data points), choosing only the back face of the first detector and the events with 5, 6 or 7 photons, to be in accordance with [3].

The best model would be with 5 photons events, since we had more data points than with 6 or 7 photons (for 10^7 data points, we had 62'591 data points for 5 photons, 36'624 for 6 photons and 23'000 for 7 photons), but it then takes a lot longer to converge, so we chose to do statistics on the events with 6 photons.

The statistics obtained were the following for the events with 6 Cherenkov photons detected as you can see in Table II:

Data points	number with 6 photons	epochs	train loss	test loss
10^4	89	3000	0.6140	3.7125
10^6	9268	2000	2.4678	2.3255
10^7	36624	1500	2.8690	2.8696
10^7	36624	10^4	1.4037	1.4331

TABLE II
MODEL STATISTICS ON EVENTS WITH 6 PHOTONS DETECTED

We can see that for 10^4 data points, we only have 89 of them on the back face of the detector and with 6 photons. This clearly produces an over-fitting of the neural network, since the train loss is 0.6140 and the test loss 3.7125, a lot higher.

For 10^6 and 10^7 data points, the model doesn't over-fit anymore since we have similar train and test losses. The more data points, the better should be the model, but also the more epochs is required to reach convergence. We can see that with 10^7 data points, we still haven't reached convergence after 1500 epochs, we need 10'000 epochs for that.

In the article [3], only 200 epochs were used, whereas in our case, a lot more epochs were required.

2) Train and test loss: We plotted the training and validation errors as a function of the number of epochs, to verify that indeed no under or over-fitting was happening. In Figure 3, we can see an over-fitting happening due to the lack of data. However, once we increase the number of data in Figure 4, there is no under nor over-fitting, the two curves of train and test (validation) loss overlap completely.

3) Histograms: We also plotted the test (validation) error we obtained in all coordinates (x, y, z and the time t) in the forms of histograms for the CoG/PCA and the neural network model, to compare both performances in the ideal situation.

In Figure 5, we can see that the MLP performance is better than CoG for 10 million data points for the x and y coordinates: the peak is sharper and there is less spread of the error. However, for the z coordinate, PCA seems to work better than the MLP as we can see in Figure 6, so further improvement of the model is needed.

Comparing our results with the article [3], we get more spread in the errors. However, it is normal since we have about 100 times fewer data points (they had 6×10^6 data points whereas we only had 3.6624×10^4) and the variance of a Gaussian distribution diminishes with $\frac{1}{N}$, N being the number of data points. In this mindset, the results we obtained replicate quite well the article and we should be in the right

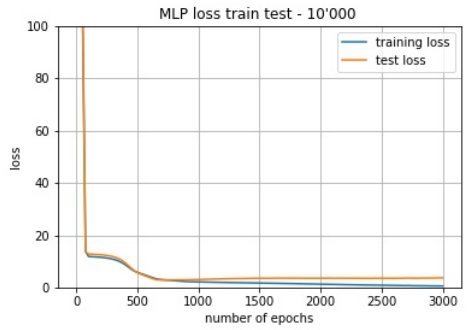

Fig. 3. The training and test (validation) loss as a function of the epochs number with 10'000 data points and 6 photons events. We can see a clear over-fitting.

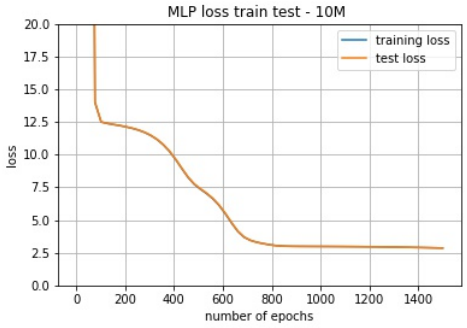

Fig. 4. The training and test (validation) loss as a function of the epochs number with 10 million data points and 6 photons events. We can see that there is no under or over-fitting and that the two curves overlap completely.

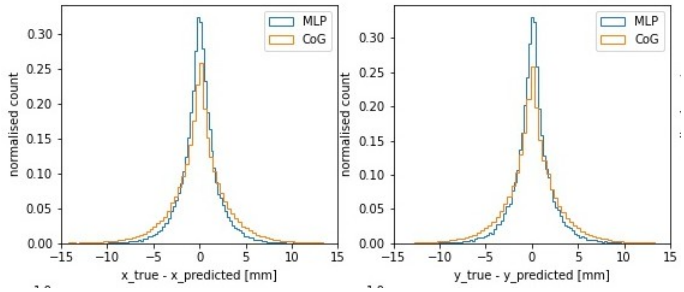

Fig. 5. The histograms of the errors in x and y, comparing the CoG in orange and the MLP in blue, for 10 million data points and 6 photons events.

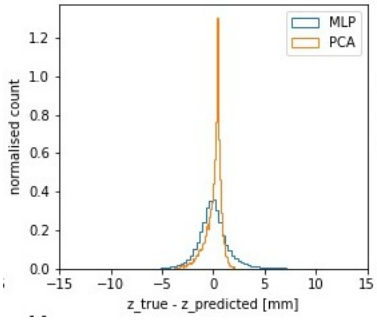

Fig. 6. The histograms of the errors in z, comparing the PCA in orange and the MLP in blue, for 10 million data points and 6 photons events.

direction, even if the model should be tested with more data points to be able to draw trustworthy conclusions. Moreover, further improvement of the model is needed regarding the z coordinate estimation, where PCA seems to still work better.

B. Finite time and spatial resolution

We also tested the neural network in a non-ideal situation when the detector time resolution equals to $\frac{FWHM}{2.355} = \frac{3\times10^{-2}}{2.355} ns$ and the crystal is discretized with 4, 16 or 64 pixels.

For the time resolution, the train loss was of 3.0516 and the test loss 3.0660, running with 1500 epochs on 10 million data points. With perfect time resolution, the two losses were 2.8690 and 2.8696 in the same case. Thus, we can see that the time resolution of the detector plays an important role in model performance.

For the discretization, some statistics can be found in Table IV:

n	pixel number	epochs	train loss	test loss
1	4	1500	4.0671	4.0379
2	16	1500	3.2384	3.2447
3	64	1500	2.8651	2.8702

TABLE III
MODEL STATISTICS ON EVENTS WITH 6 PHOTONS DETECTED WITH A FINITE NUMBER OF PIXELS ON THE DETECTOR.

Thus, we can conclude that the more pixels we have, the better the model performance is. Moreover, having 64 pixels in the detector seems to produce a performance close to the one in the ideal situation.

IV. CONCLUSIONS

To conclude, we indeed managed to find a Machine Learning model that gets close to the performance of the CoG (center of gravity) method used in the clinic in PET imaging in our case where the crystal detects Cherenkov photons and even improves it in the x and y coordinates. This model consists of a multilayer perceptron (MLP) of 3 hidden layers, each with 256 hidden units, following the architecture given in [3]. We adjusted it slightly in our case namely increasing 50-fold the number of epochs used and removing the batch normalisation. We then went beyond the ideal situation and modelled the physical time resolution of the detector, as well as the spatial resolution with various detector pixel numbers. Regarding the latter, we concluded that having 64 pixels is enough to produce a very good performance close to the ideal situation.

To continue the work in the future, we could try to improve the neural network architecture so it works better on the z coordinate. We will namely try to fit recurrent neural networks to the data so the model is more robust to the input size and performs better. We will thus look at the natural language processing tools to improve even more our model.

V. Acknowledgments

We would like to thank our supervisors, Dr. Pol del Aguila Pla and Sofiia Forostenko for their support and help in this project. We would also like to thank our project teaching assistant, Zohreh Mostaani, for her help regarding the neural network code.

References

[1] D. L. Bailey, D. W. Townsend, P. E. Valk, and M. N. Maisey, Eds., Positron Emission Tomography: Basic Sciences. London: Springer-Verlag, 2005.

[2] J. Counet, "Hope for metastatic prostate cancer patients: targeted alpha therapy shows impressive results," EU Science Hub - European Commission, Dec. 21, 2016. https://ec.europa.eu/jrc/en/news/prostate-cancer-alpha-therapy-shows-impressive-results (accessed Nov. 21, 2020).

[3] F. Hashimoto, K. Ote, R. Ota, and T. Hasegawa, "A feasibility study on 3D interaction position estimation using deep neural network in Cherenkov-based detector: A Monte Carlo simulation study," Biomedical Physics & Engineering Express, vol. 5, Feb. 2019, doi: 10.1088/2057-1976/ab098e.