Detector spatial resolution optimization based on neural networks

V. Tskhay

vtskhay@lebedev.ru

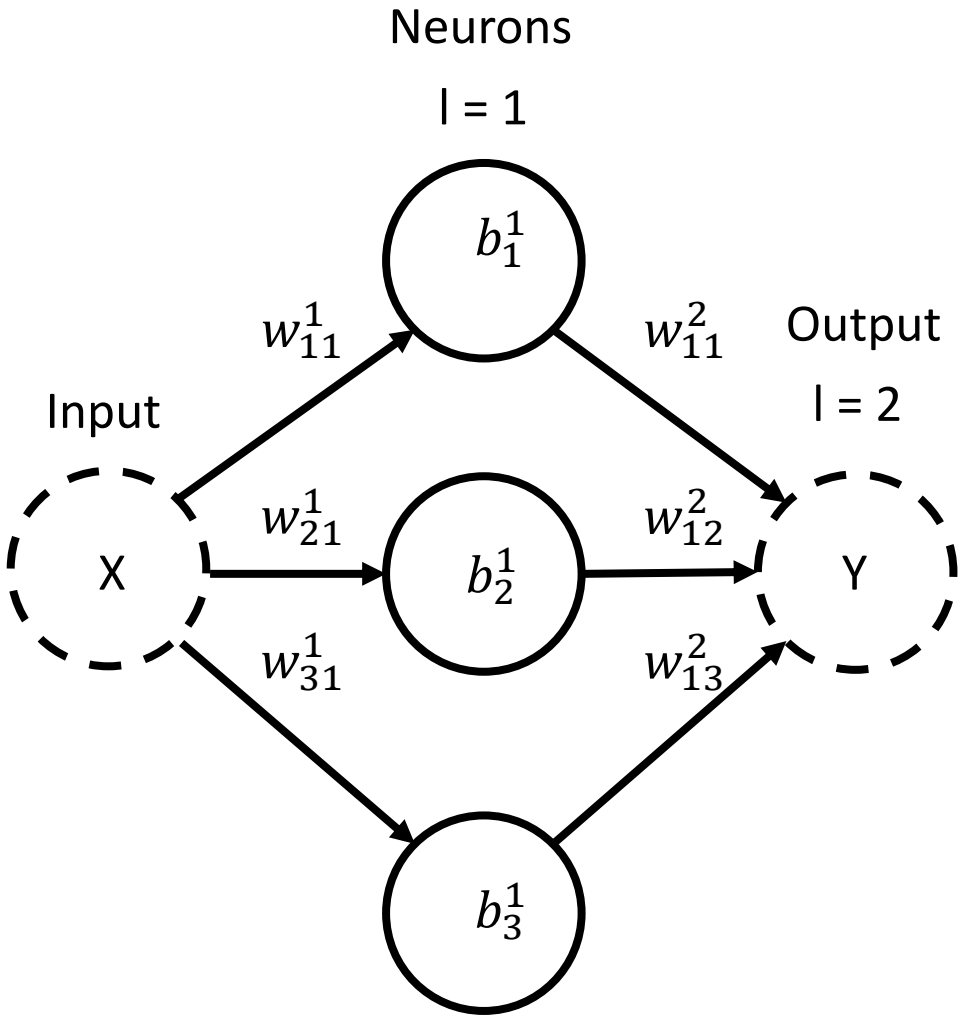
P.N. Lebedev Physical Institute of the Russian Academy of Sciences
119991 Moscow, Leninskii prospekt 53

Introduction

- The concept of neural networks is not new – the idea first appeared in 1943, in a work *"A Logical Calculus of Ideas Immanent in Nervous Activity", by Warren S. McCulloch and Walter Pitts.*
- In modernity, due to increase of calculating power and accessibility, neural networks are used for a wide variety of applications.
- Most neural networks are digital but can also be made in analogue form.
- One particular use case is signal processing, but neural networks are also used for simulations, physical cuts, track reconstruction, etc.
- Neural networks have the potential to approximate any function with any degree of precision.[1]
- However, traditional neural networks require "training", which requires a set of "solved" problems.
- For example, for classifying handwritten digits, a set of labelled digits is necessary.
- One particular application is found in experimental positron-emission tomographs

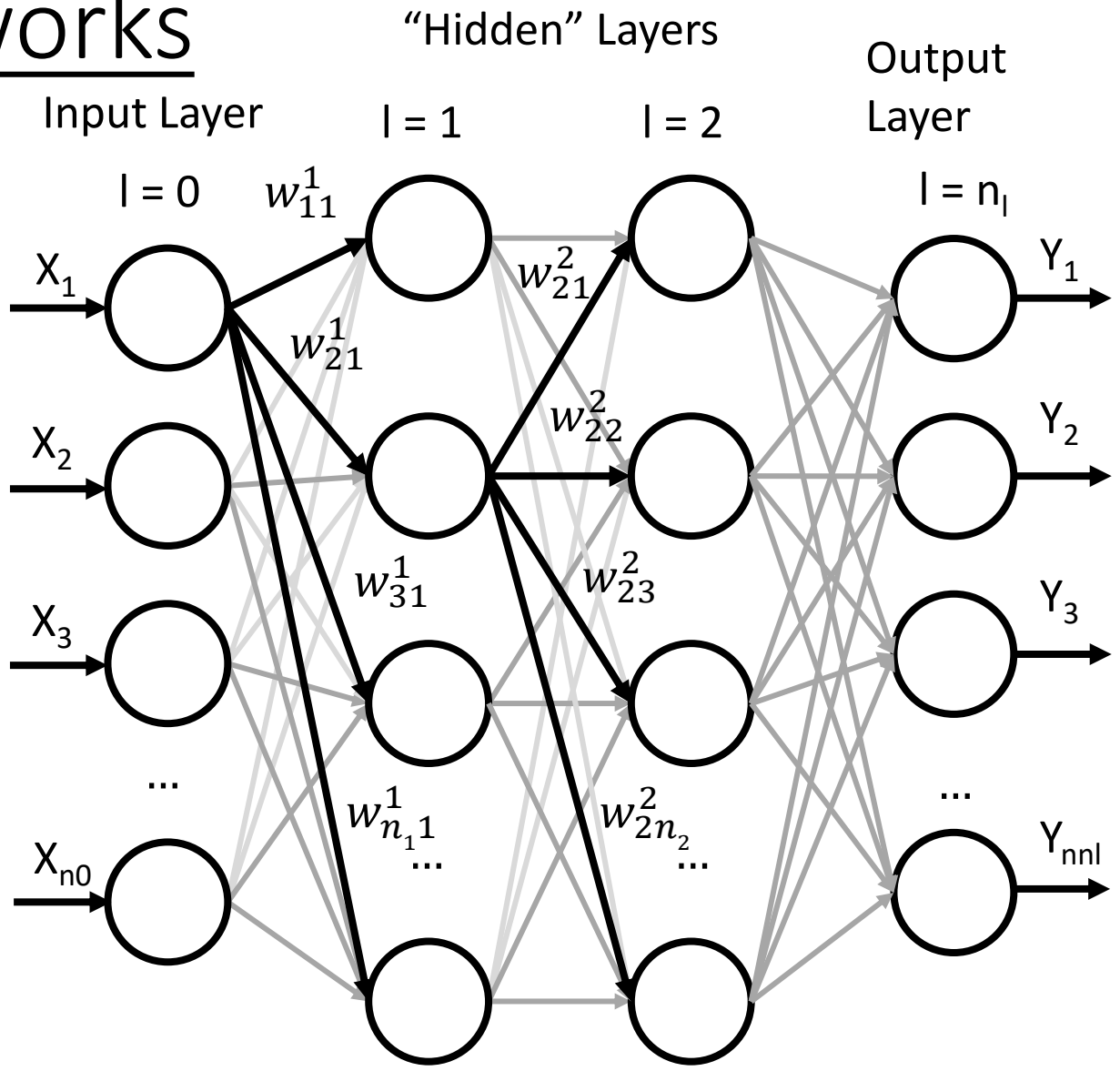
Artificial neural networks basics

- Artificial Neural network consists of nodes called “neurons”, which are divided into layers
- Each neuron has an “activation function”, usually a sigma function $\sigma = \frac{1}{1+e^{-z}}$
- Each node of a layer is connected to each node of a previous layer. Each connection has a specific weight **w** factor to it. The input of a neuron is the sum of the incoming signals multiplied by their weights.
- Each neuron also has a bias **b**.
- The output of a neurons function **a** is called activation $a = \sigma\left(w_{jk}^{l} x + b_{j}^{l}\right)$,
- Such a network, given a large enough amount of neurons, can approximate any function of any number of variables [1]



“Deep” neural networks

- Neural networks can consist of multiple layers
- In such cases, the output of the first layer becomes the input of the second, and etc.
- In such networks, the formulas change accordingly
- $a_j^l = \sigma\left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l\right)$, or in vector form
- $\boldsymbol{a}^l = \sigma\left(\boldsymbol{w}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l\right)$
- The value $\boldsymbol{w}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l$ is defined as **z**
- Generally, instead of feeding every (weighted) value of the input vector **X** to every neuron of the input layer, individual components are used as individual inputs

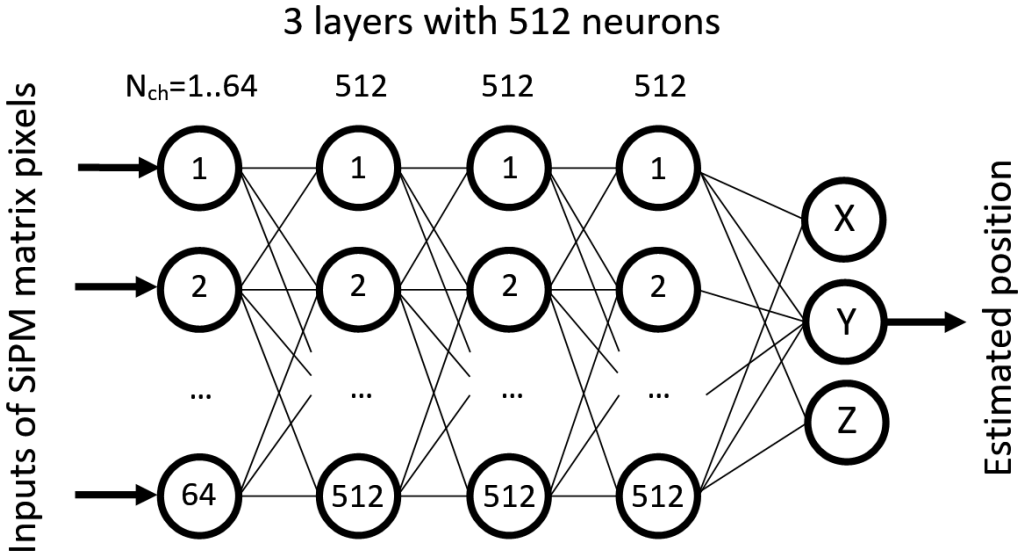


Training the neural network

- For the network to work properly, we need to find the correct coefficients for matrices **w** and vectors **b**.
- To train the network, we need a set of input vectors **x** with respective solution vectors **y**.
- We can evaluate the work of our network with a loss function, for example a mean squared error:
- $C(w,b) \equiv \frac{1}{2n}\sum_{\boldsymbol{x}}\|\boldsymbol{y}(\boldsymbol{x}) - \boldsymbol{a}\|^2$
- Where n – number of training events, $\boldsymbol{a}$ – vector of solutions provided by the neural network for a particular input vector **x,** and **y** is the corresponding vector of correct values.
- The simple approach for finding the coefficients is plain MC. The coefficients are randomized, multiple attempts to analyze the data are made and the variable arrays that perform the best (i.e. the C function is minimal) are kept. Then the process is repeated using the best performing coefficients with some random additions.
- The more complex method is called “error backpropagation”, popularized by a paper “Learning representations by back-propagating errors” by David E. Rumelhart et.al.[2] The errors of each training event are used to make small corrections to the coefficients to “push” the function closer to a local minimum.

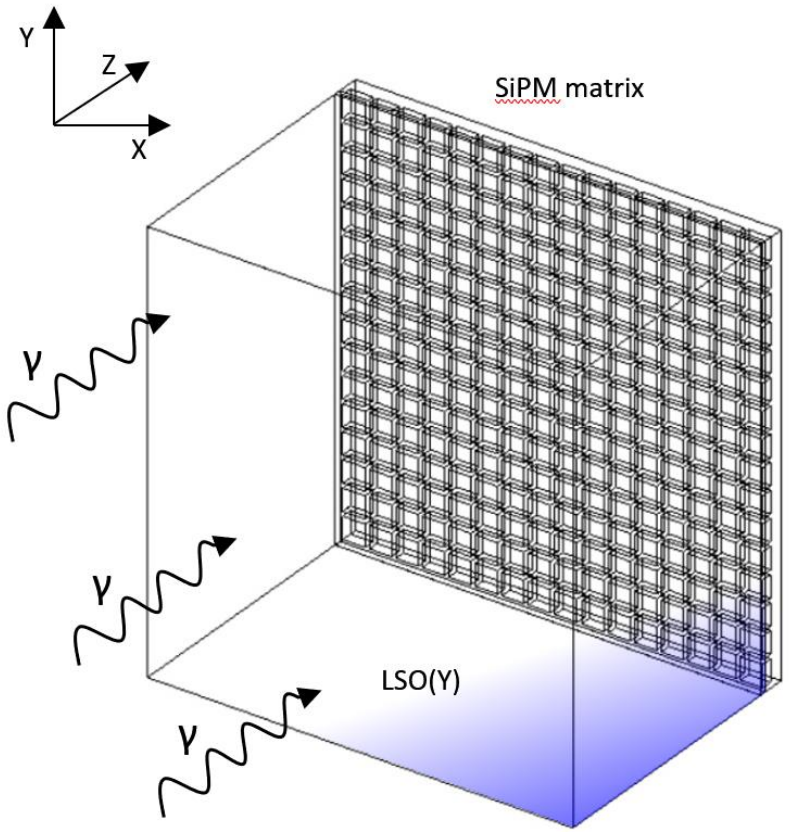
Practical application

- A crystal gamma-ray detector with an array of SiPMs or PMTs produce a matrix of signals, essentially an image.
- The 2d distribution shape will be dependent on the exact point of particle interaction.
- The X and Y coordinates are relatively easy to reconstruct analytically. Z is more complex.
- Compton-effect events will be harder to reconstruct than photo effect, as there will be more than one interaction per event and the distribution will be distorted.



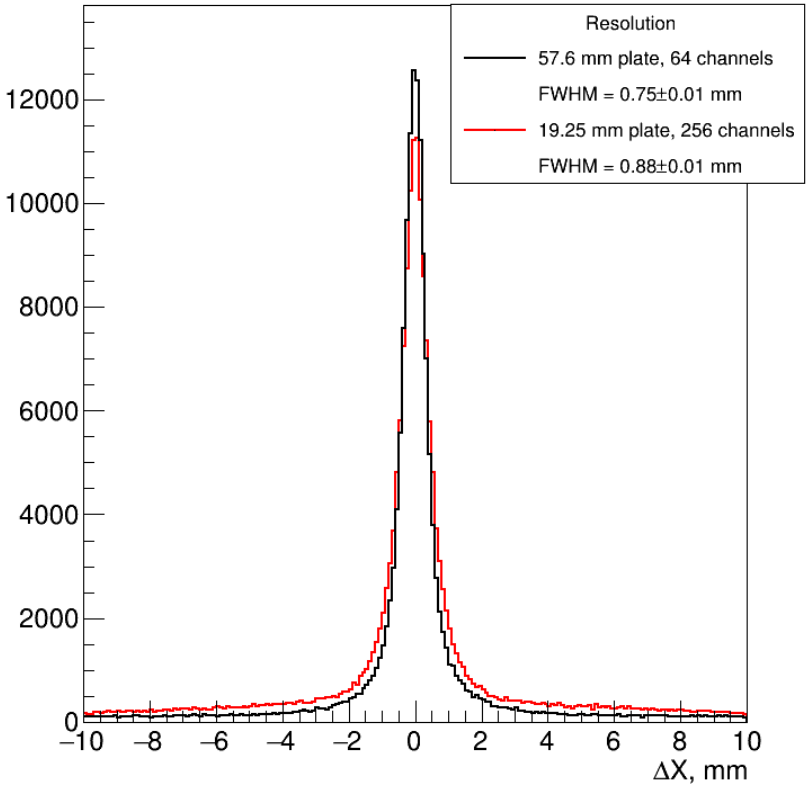
MC model setup

- 2 models were made, both consisted of LSO(Y) crystals of with SiPM arrays
- 1st model consisted of a 19.25 x 19.25 x 12 mm crystal with a 16 x 16 channel SiPM
- 2nd model consisted of a 57.6 x 57.6 x 12 mm crystal with a 8 x 8 channel SiPM
- 1000000 events of incident 511 keV gamma rays were generated for each model. The events were then separated into 2 parts, 500000 for training, and 500000 for testing.
- The events are separated to ensure that the network is applicable beyond the set it is familiar with.

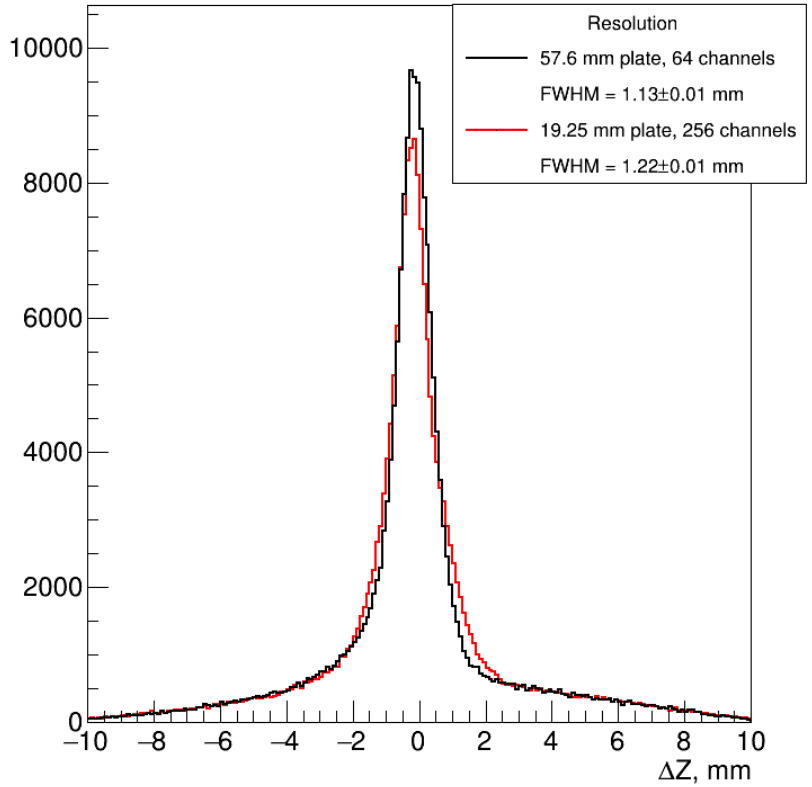


Results 1

ΔX Compton + photoeffect



ΔZ Compton + photoeffect



57.6 mm plate

$\Delta x = 0.75$ mm

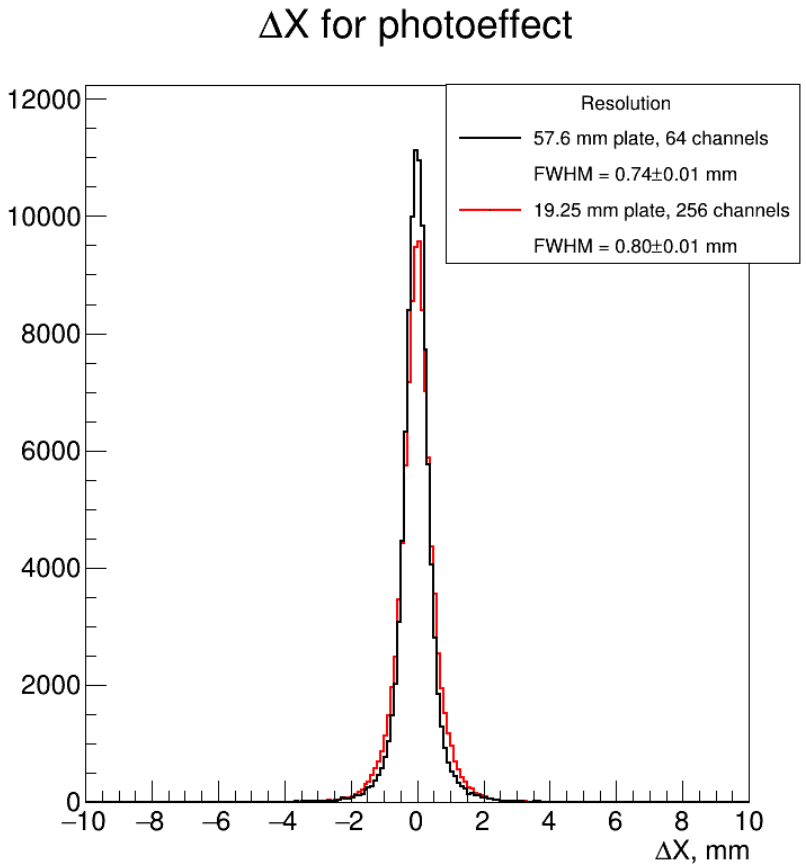
$\Delta z = 1.13$ mm

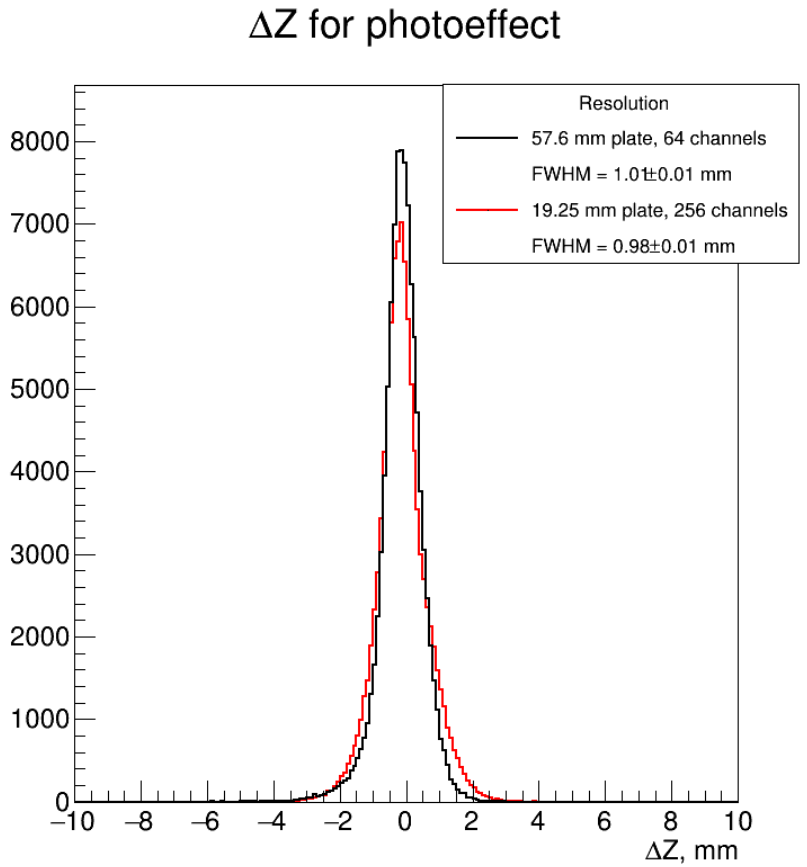
19.25 mm plate

$\Delta x = 0.88$ mm

$\Delta z = 1.22$ mm

Results 2





57.6 mm plate

$\Delta x = 0.74$ mm

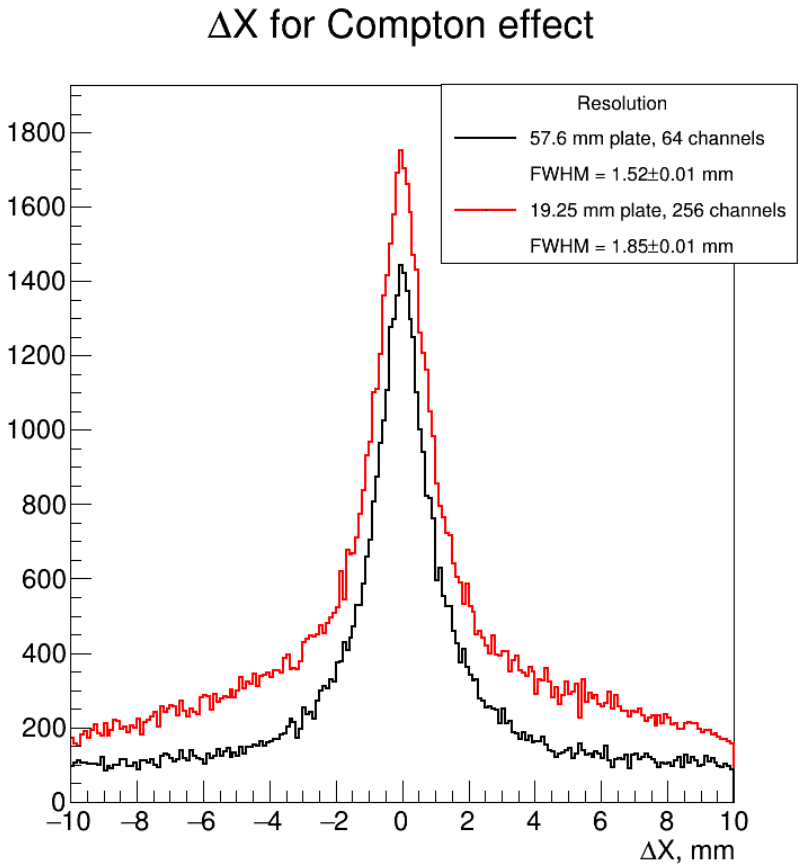
$\Delta z = 1.01$ mm

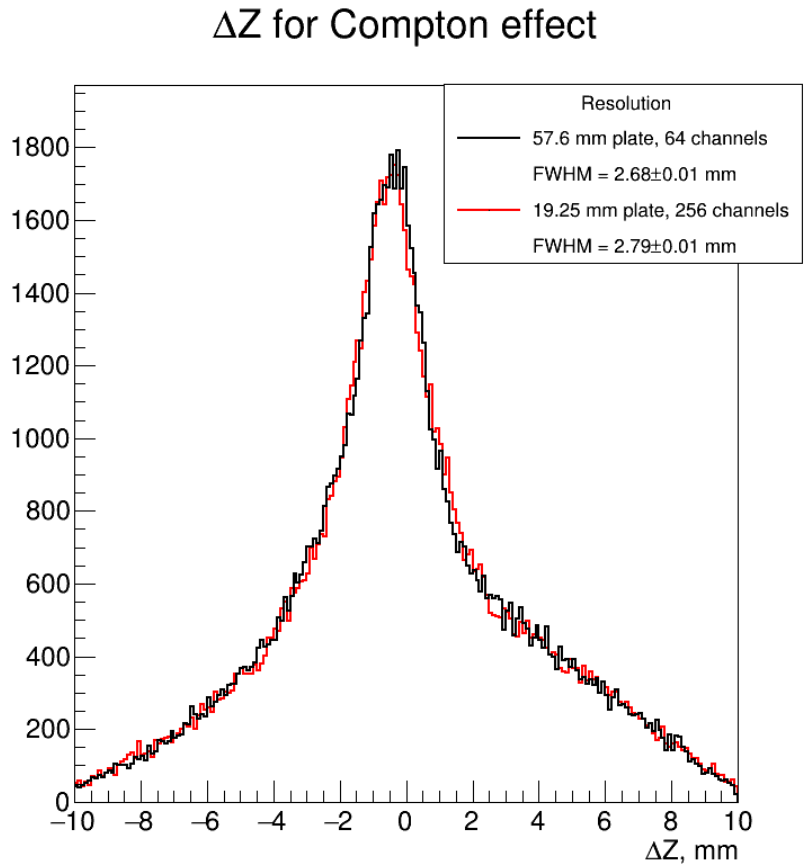
19.25 mm plate

$\Delta x = 0.80$ mm

$\Delta z = 0.98$ mm

Results 3





57.6 mm plate

$\Delta x = 1.52$ mm

$\Delta z = 2.68$mm

19.25 mm plate

$\Delta x = 0.88$ mm

$\Delta z = 1.22$ mm

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
Физический институт имени П.Н.Лебедева РОССИЙСКОЙ АКАДЕМИИ НАУК
Ф И А Н

BM@N

Results 4

256 channel matrix

Layers \ Neurons	256 neurons		512 neurons		1024 neurons	
	dX	dZ	dX	dZ	dX	dZ
1 layer	0.98	1.78	0.99	1.75	0.99	1.7
2 layers	0.91	1.36	0.94	1.36	0.92	1.3
3 layers	0.91	1.29	0.96	1.3	0.88	1.22

64 channel matrix

Layers \ Neurons	256 neurons		512 neurons		1024 neurons	
	dX	dZ	dX	dZ	dX	dZ
1 layer	0.82	1.3	0.77	1.33	0.78	1.22
2 layers	0.79	1.17	0.75	1.13	0.78	1.15
3 layers	0.8	1.19	0.79	1.17	0.78	1.17

Conclusions

- Neural networks can and are used for event reconstruction in detectors.
- Neural networks have the potential to achieve a degree of precision limited only by experimental constraints.
- Preparing the database of training events is the most challenging part.
- Добавить мотивацию кристаллы пэт, удешевление

- An example of an experimental work can be found in “Gradient Tree Boosting-Based Positioning Method for Monolithic Scintillator Crystals in Positron Emission Tomography” by F. Muller et.al.
- **DOI:** 10.1109/TRPMS.2018.2837738
- A good example of neural network application with training based on grid-like scattering of event can be found in “Artificial neural networks for positioning of gamma interactions in monolithic PET detectors” by M. Decuyper et.al.
- **DOI:** 10.1088/1361-6560/abebfc

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
Физический институт имени П.Н.Лебедева РОССИЙСКОЙ АКАДЕМИИ НАУК
Ф И А Н

BM@N

References

- 1. Cybenko, G. (1989). "Approximation by superpositions of a sigmoidal function".
- DOI:10.1007/BF02551274
- 2. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors.
- DOI: 10.1038/323533a0

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
Физический институт имени П.Н.Лебедева РОССИЙСКОЙ АКАДЕМИИ НАУК
Ф И А Н

BM@N

Backups

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ
Физический институт имени П.Н.Лебедева РОССИЙСКОЙ АКАДЕМИИ НАУК
Ф И А Н

BM@N

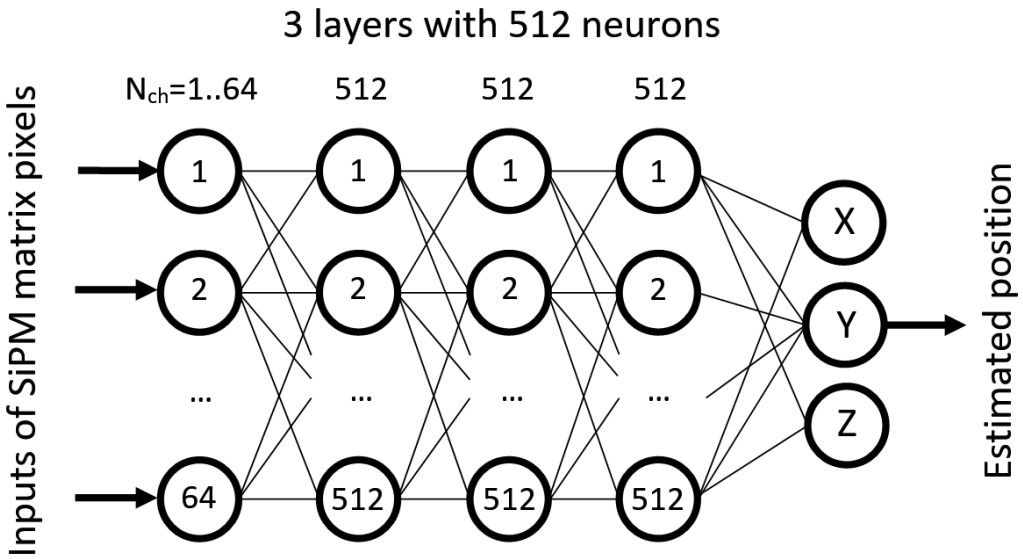
Backpropagation equations

- For the backpropagation algorithm to work, we need to calculate the partial derivatives $\partial C_x/\partial w_{jk}^l$ and $\partial C_x/\partial b_j^l$.
- For this we need to introduce the value δ_j^l, that represents the error of the **j** neuron of the **l** layer, $\delta_j^l = \frac{\partial C_x}{\partial z_j^l}$.
- For the output layer L, $\delta_j^L = \frac{\partial C}{\partial a_j^L}\sigma'(z_j^L)$, where $\frac{\partial C}{\partial a_j^L} = (a_j^L - y_j)$.

- $\delta^L = \nabla_a C \odot \sigma'(\boldsymbol{z}^L) = (a^L - \boldsymbol{y}) \odot \sigma'(z^L)$
- $\delta^l = ((\boldsymbol{w}^{l+1})^T \boldsymbol{\delta}^{l+1}) \odot \sigma'(\boldsymbol{z}^l)$
- $\frac{\partial C_x}{\partial b_j^l} = \delta_j^l$
- $\frac{\partial C_x}{\partial w_{jk}^l} = a_j^{l-1}\delta_j^l$

Training details

- The error back-propagation method was used
- The 2d distribution shape will be dependent on the exact point of particle interaction.
- The X and Y coordinates are relatively easy to reconstruct analytically. Z is more complex.
- Compton-effect events will be harder to reconstruct than photo effect, as there will be more than one interaction per event and the distribution will be distorted.
- The results demonstrated are achieved with a network that has all 3 coordinates as outputs, at the same time, separate networks with individual outputs produce similar results.



Approximation visual

- When w_{11}^1 is large, a $= \frac{1}{1+e^{-(w_{11}^1 x + b_1^1)}}$, behaves like a “step” function,
- $s_1 = -\frac{w_{21}^1}{b_2^1}$ defines the step position.
- w_{11}^2 defines the height of the step.
- If $w_{11}^2 = -w_{12}^2$ and s_1 and $s_2 = -\frac{w_{21}^1}{b_2^1}$ are offset, we get a rectangular function.