Reconstruction of particle tracks using Deep Neural Networks and estimation of reconstruction errors

Karol Białas Mateusz Slysz

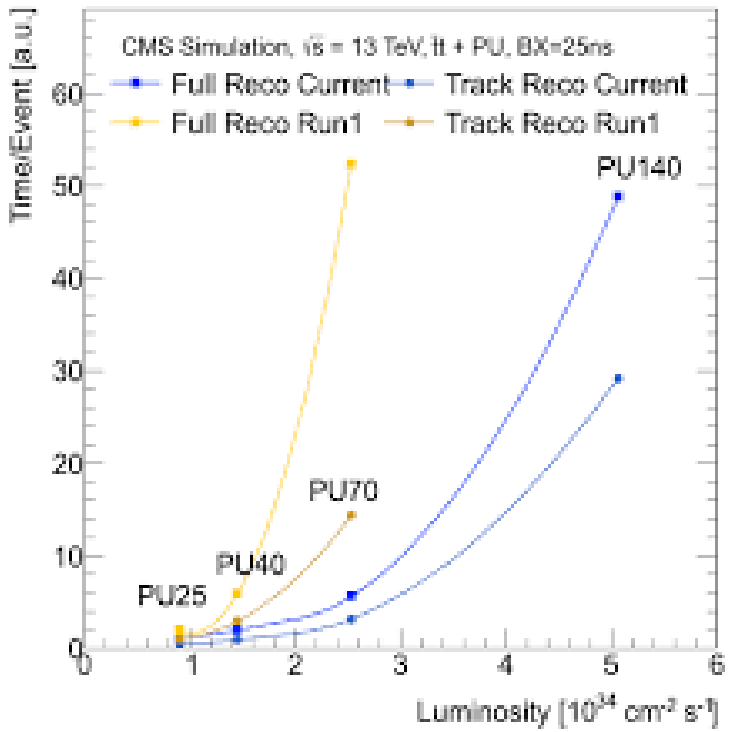
Supervisor: dr hab. Marcin Wolter

PPSS Programme, IFJ

August 2, 2019

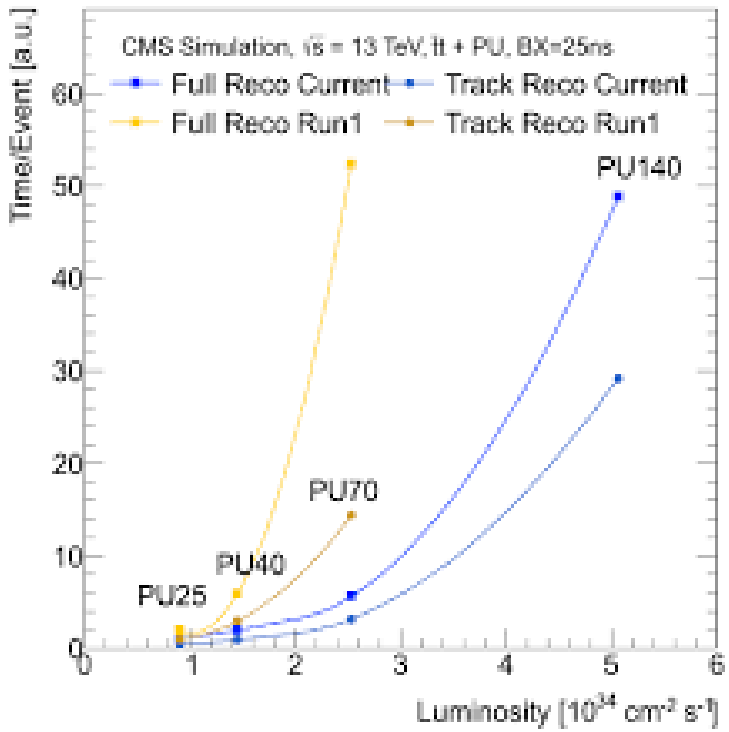
Motivation

- For currently used algorithms tracking time needed to process one event grows quickly (worse than $O(n^2)$) with luminosity.



Motivation

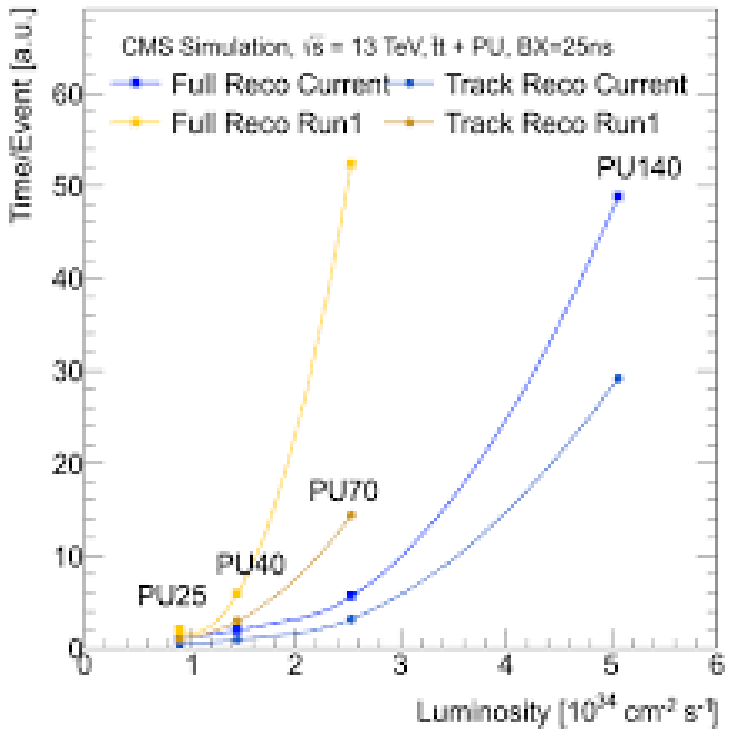
- For currently used algorithms tracking time needed to process one event grows quickly (worse than $O(n^2)$) with luminosity.



- Training neural networks can also be long, but can be parallelized.

Motivation

- For currently used algorithms tracking time needed to process one event grows quickly (worse than $O(n^2)$) with luminosity.



- Training neural networks can also be long, but can be parallelized.
- Once trained, neural networks work much faster.

Classical Approach - Convolutional Neural Network

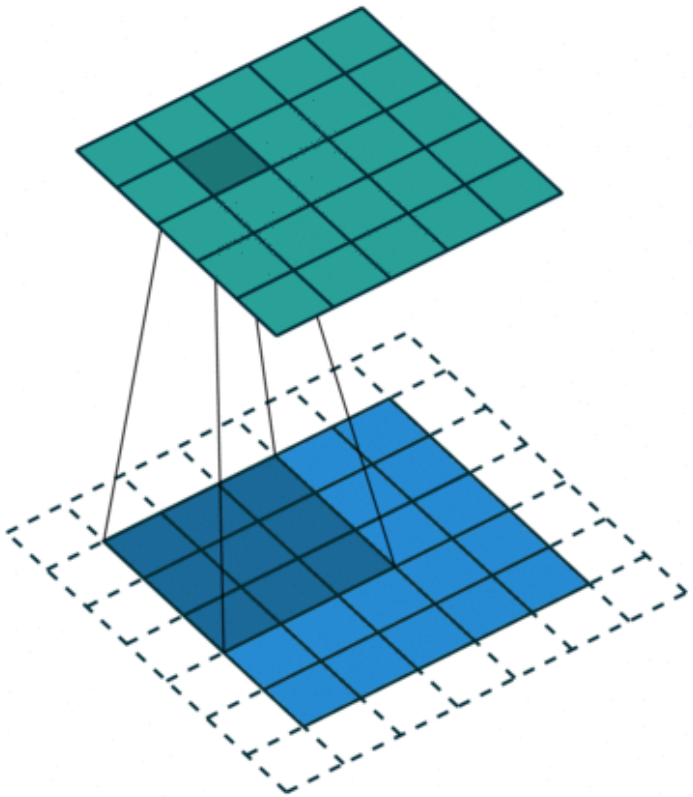


Fig. 1: Convolutional Layer [1]

Classical Approach - Convolutional Neural Network

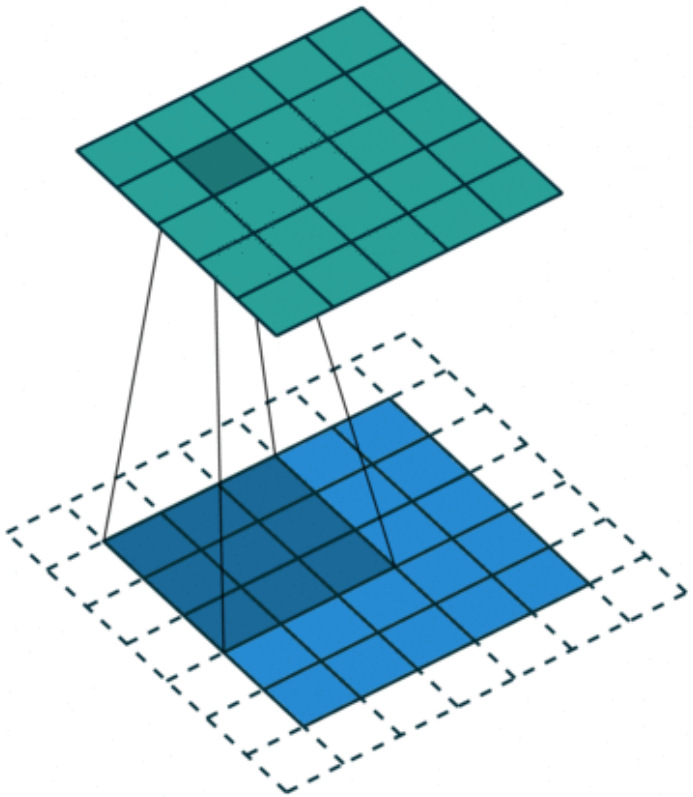


Fig. 1: Convolutional Layer [1]

CNN returns only a single value $\rightarrow$ no error estimation!

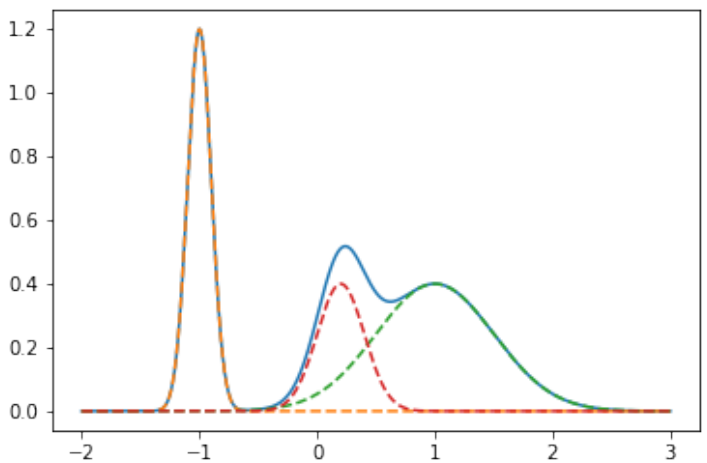
New Idea - Mixture Density Network

- Introduced by C. Bishop in 1994, brought back with Deep Learning

New Idea - Mixture Density Network

- Introduced by C. Bishop in 1994, brought back with Deep Learning
- Returns probability density (weighted sum of normal distributions) instead of discrete values $\rightarrow$ error estimation

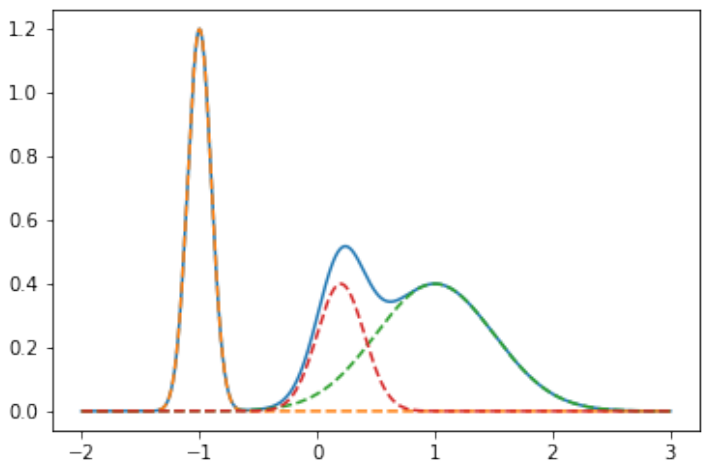
$$\rho(y) = \sum_{i=1}^{M} w_i \mathcal{N}_i(\mu_i, \sigma_i^2, y)$$



New Idea - Mixture Density Network

- Introduced by C. Bishop in 1994, brought back with Deep Learning
- Returns probability density (weighted sum of normal distributions) instead of discrete values $\rightarrow$ error estimation

$$\rho(y) = \sum_{i=1}^{M} w_i \mathcal{N}_i(\mu_i, \sigma_i^2, y)$$

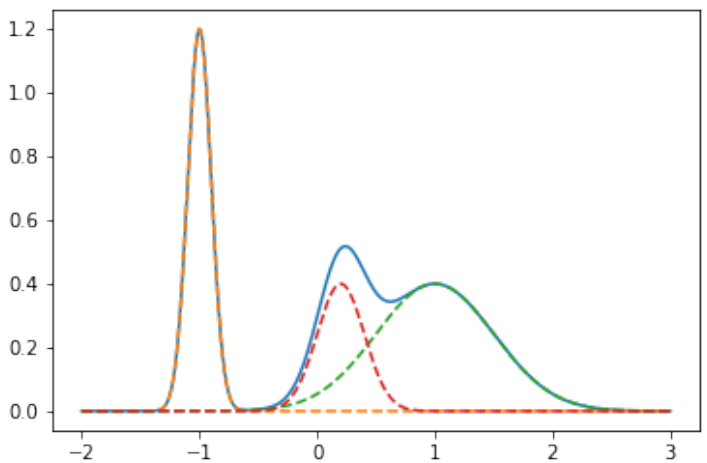


- Only one track parameters are given to the network during training

New Idea - Mixture Density Network

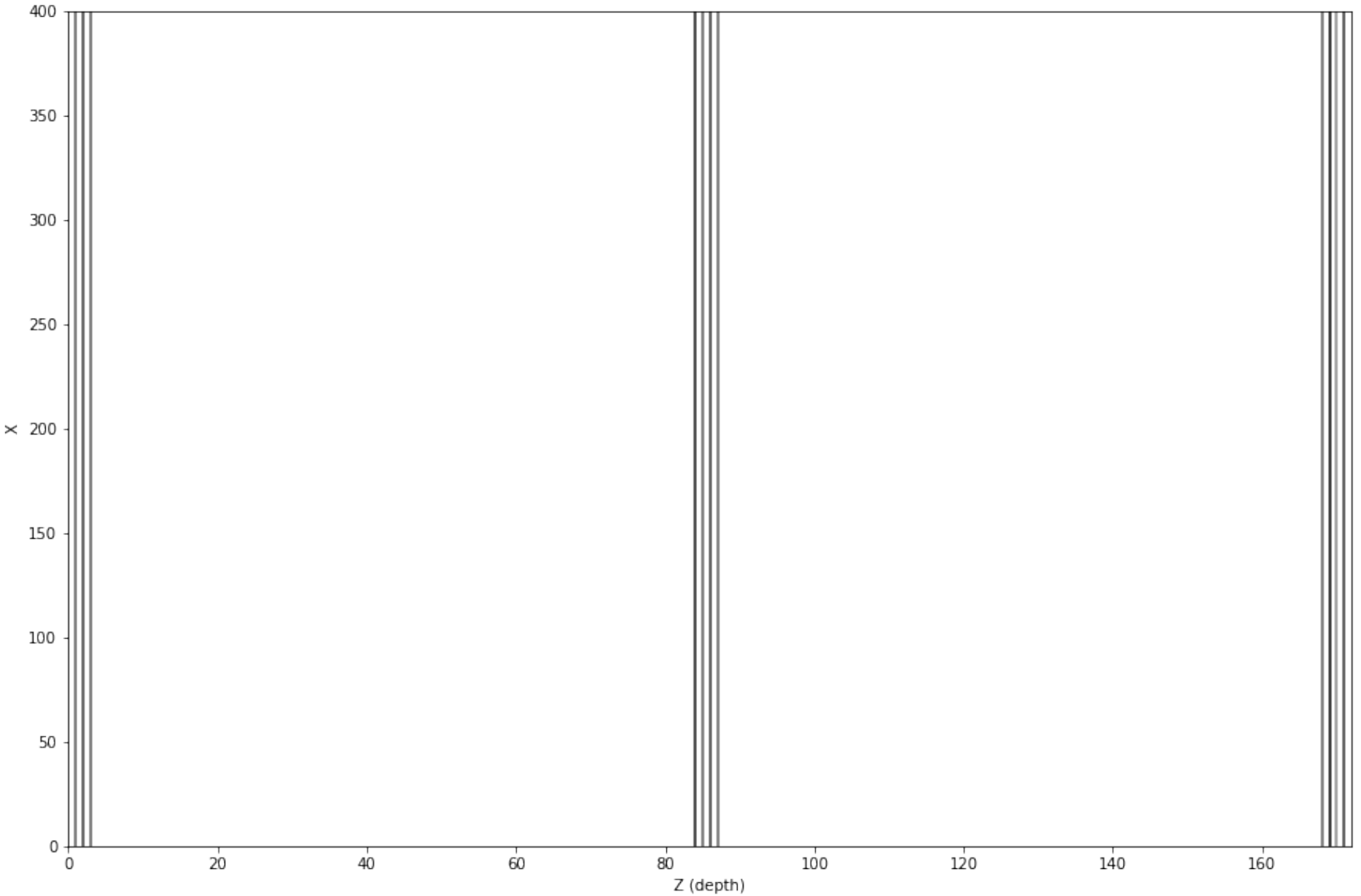
- Introduced by C. Bishop in 1994, brought back with Deep Learning
- Returns probability density (weighted sum of normal distributions) instead of discrete values $\rightarrow$ error estimation

$$\rho(y) = \sum_{i=1}^{M} w_i \mathcal{N}_i(\mu_i, \sigma_i^2, y)$$

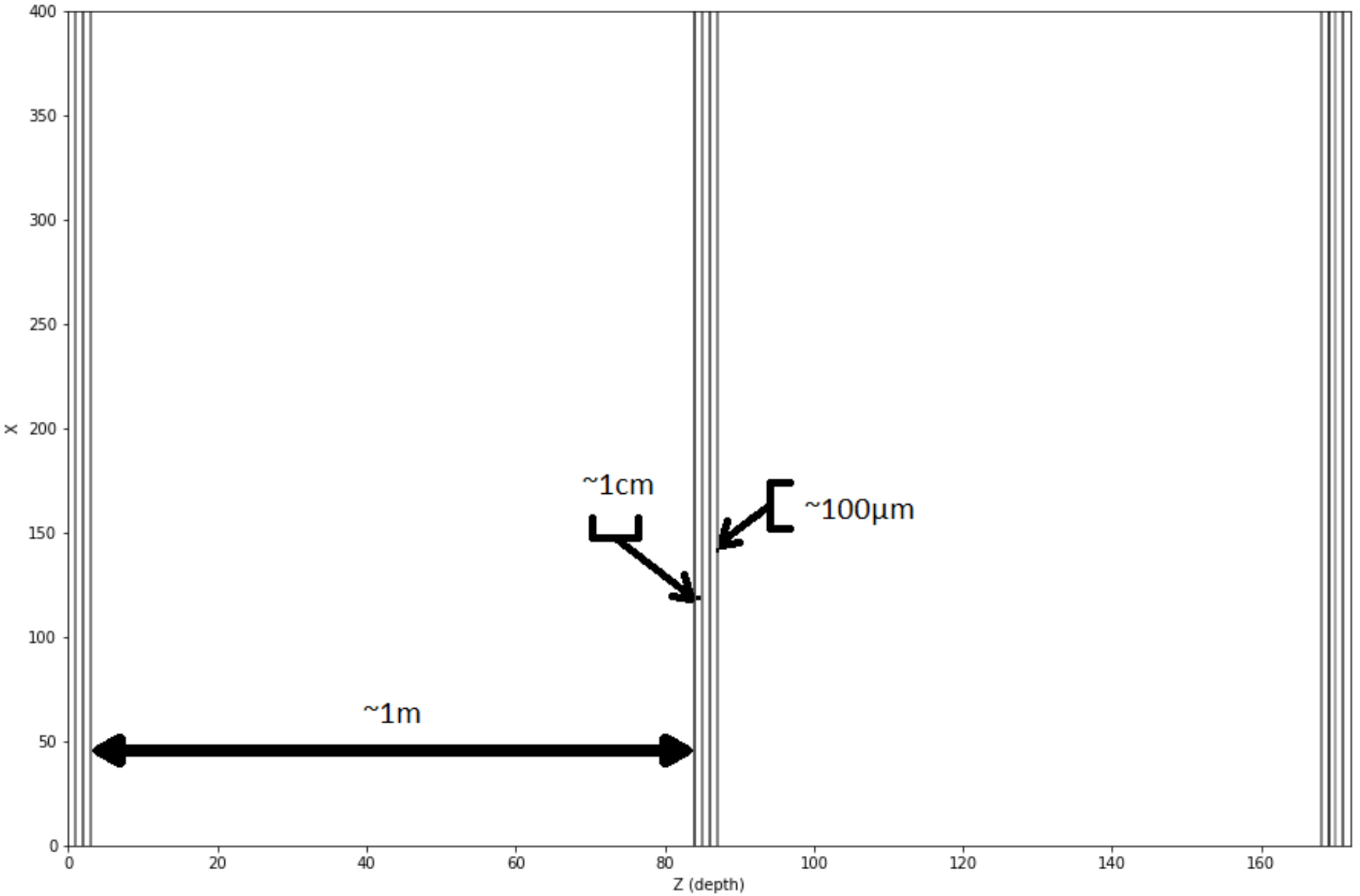


- Only one track parameters are given to the network during training
- Works well with variable output size (number of tracks may differ)

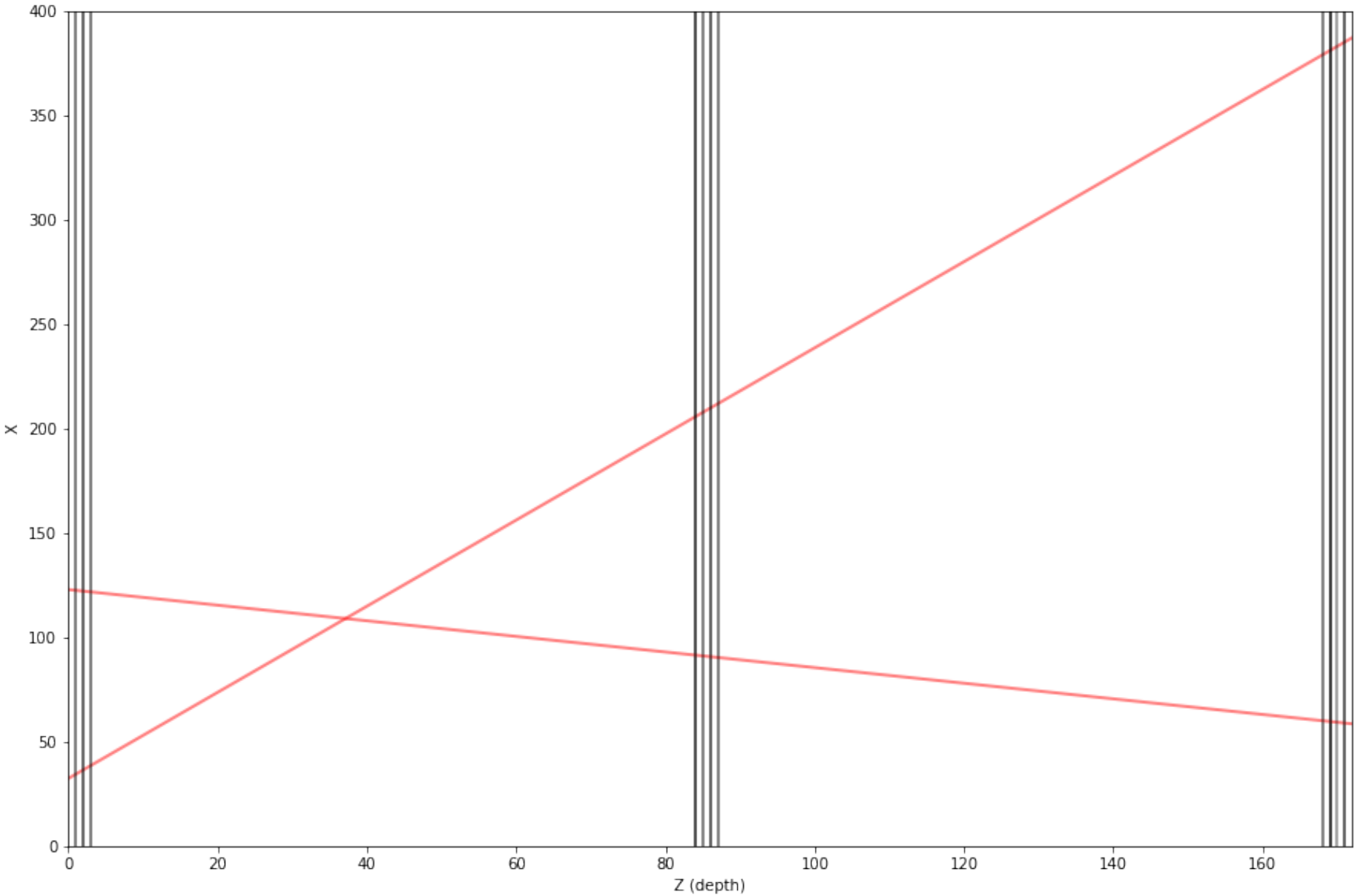
Goal - reconstruction of particle tracks



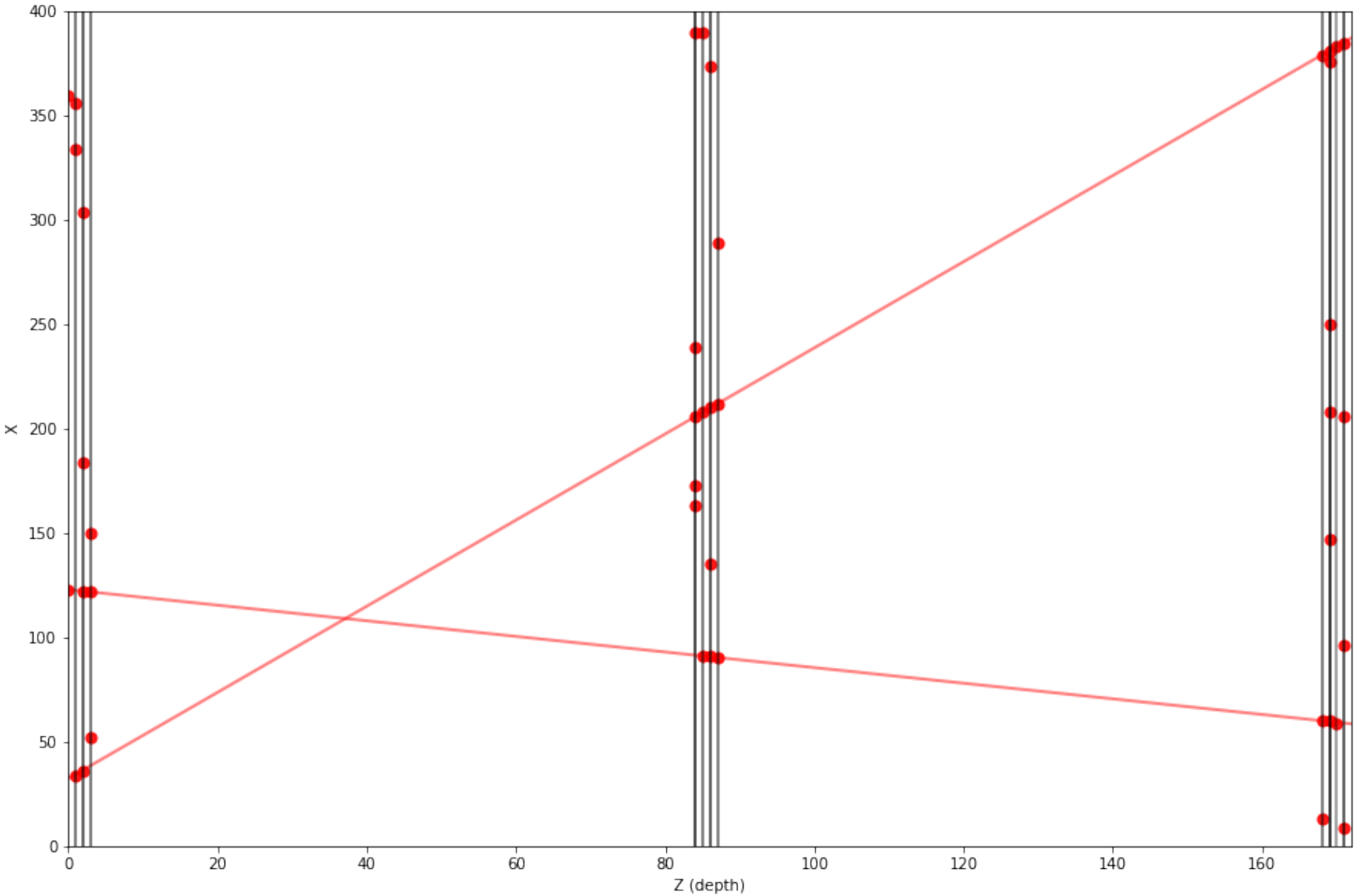
Goal - reconstruction of particle tracks



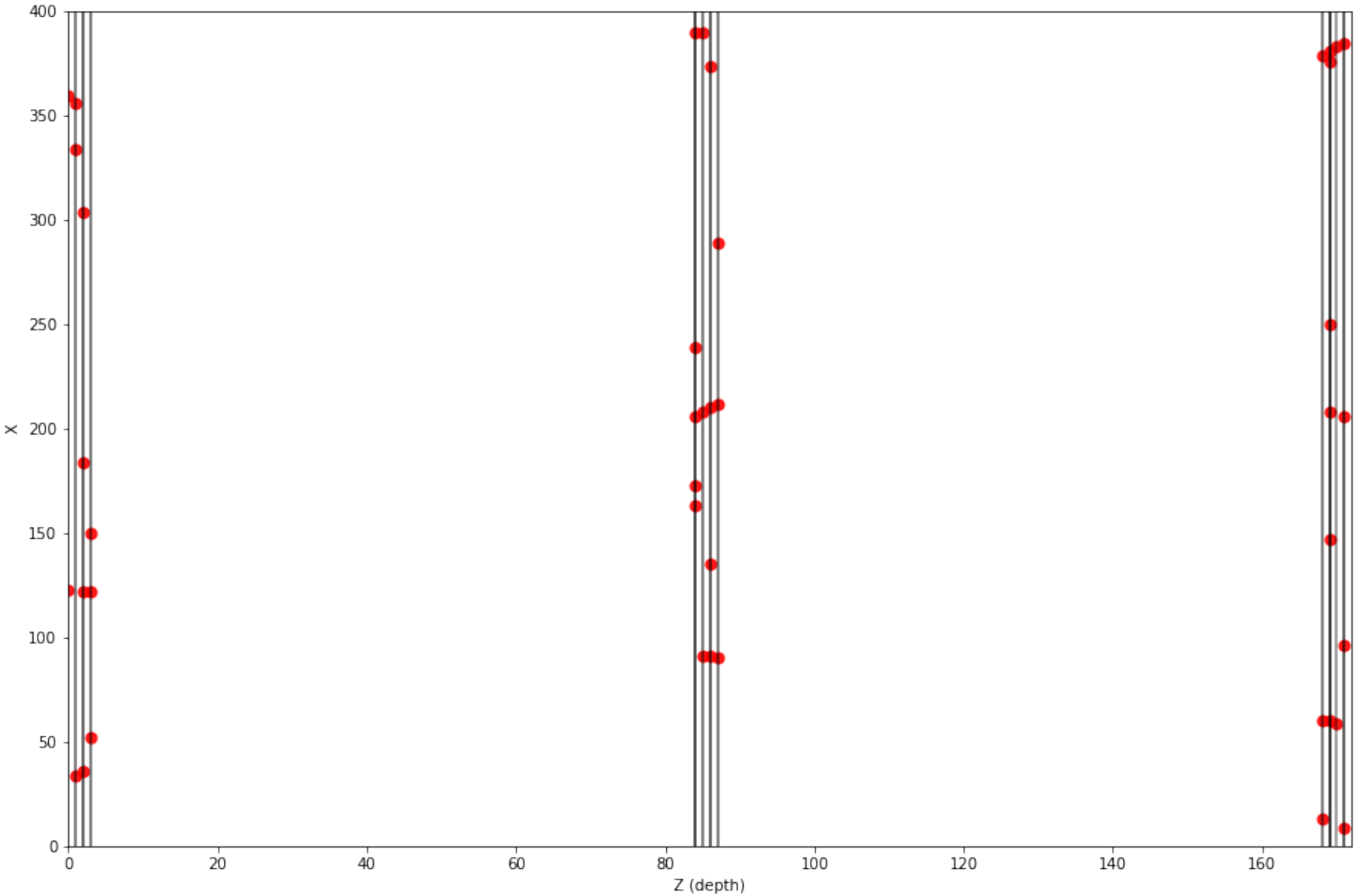
Goal - reconstruction of particle tracks



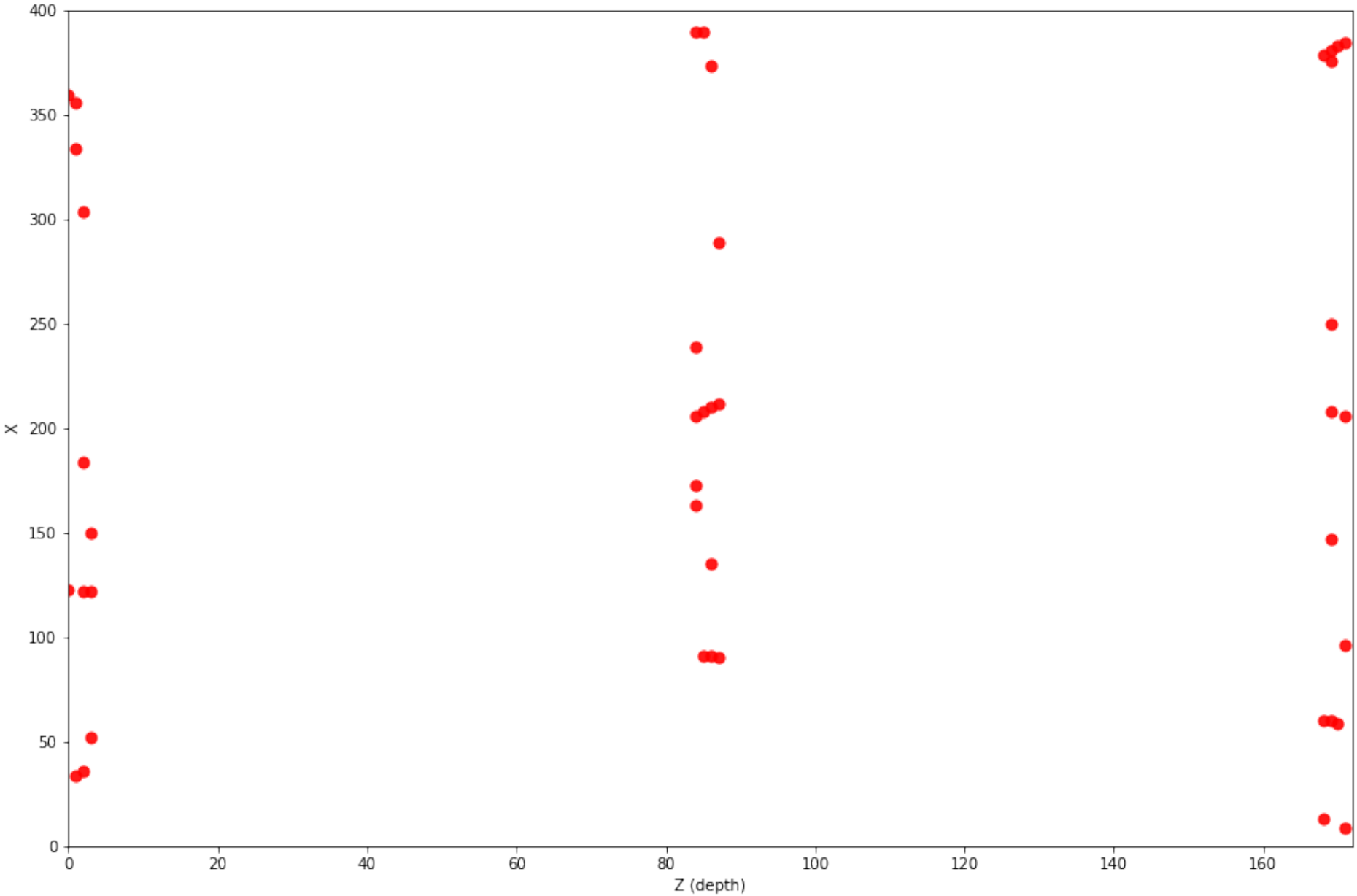
Goal - reconstruction of particle tracks



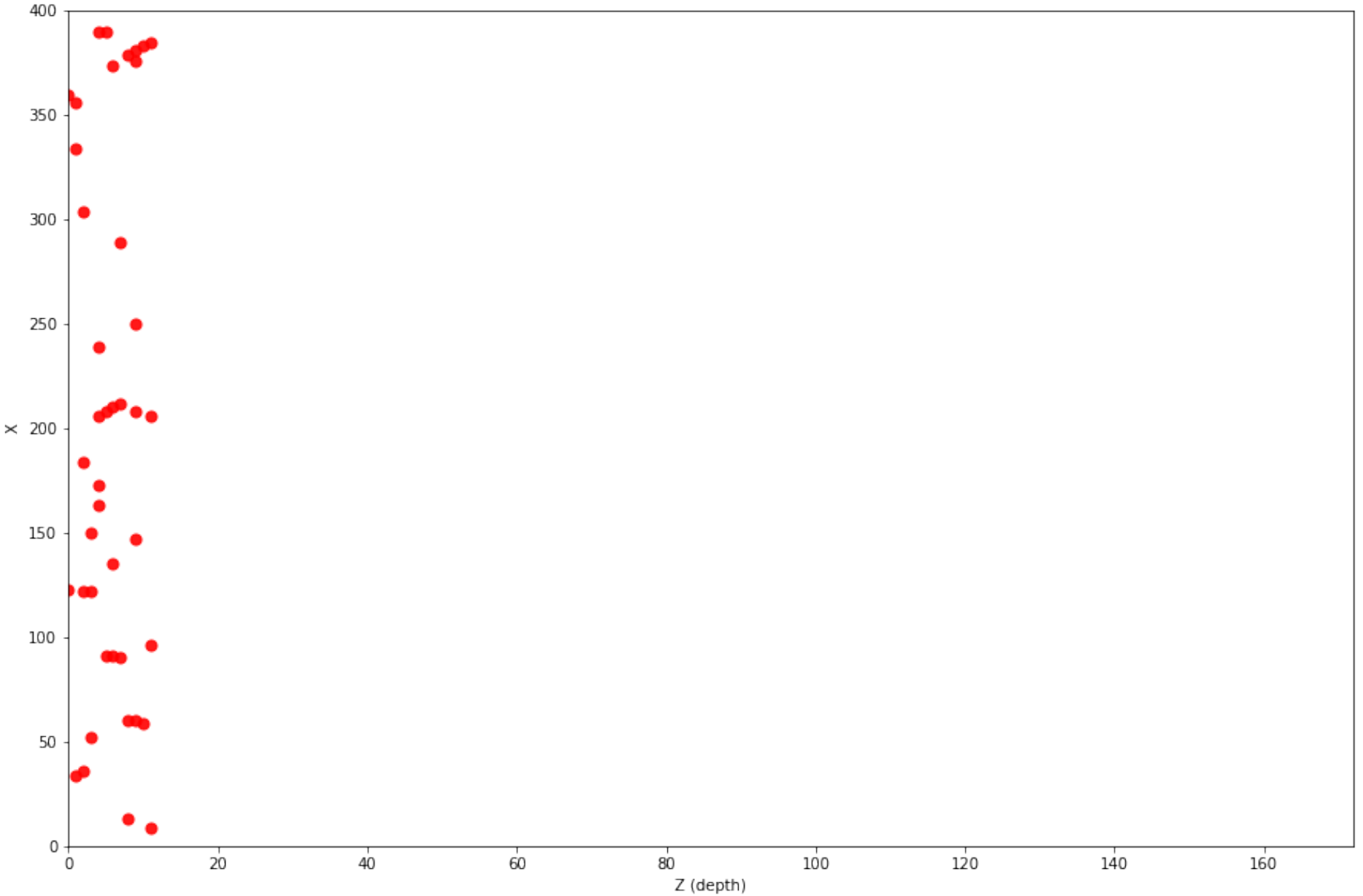
Goal - reconstruction of particle tracks



Goal - reconstruction of particle tracks

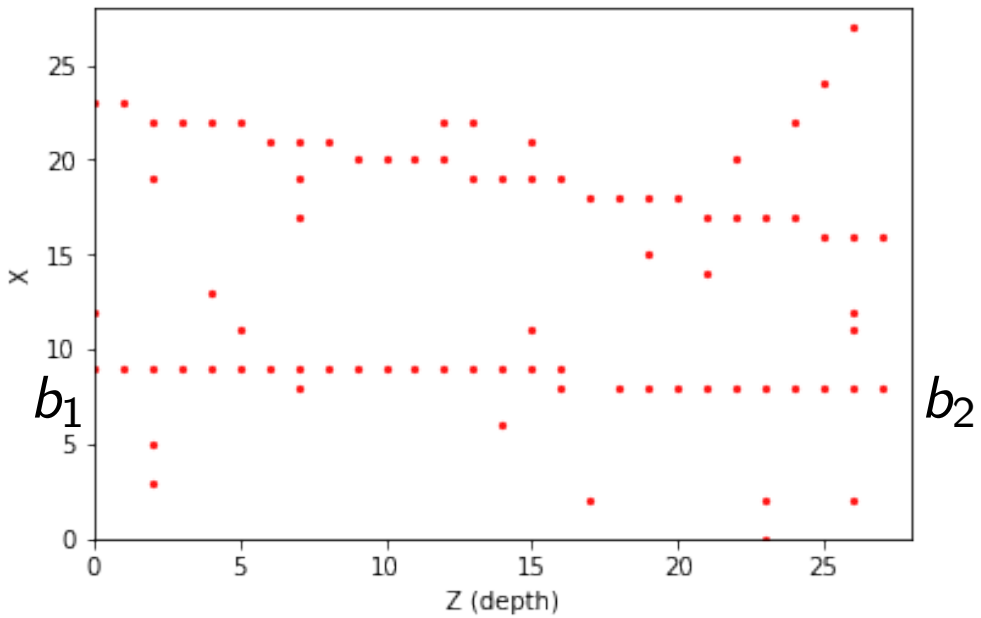


Goal - reconstruction of particle tracks



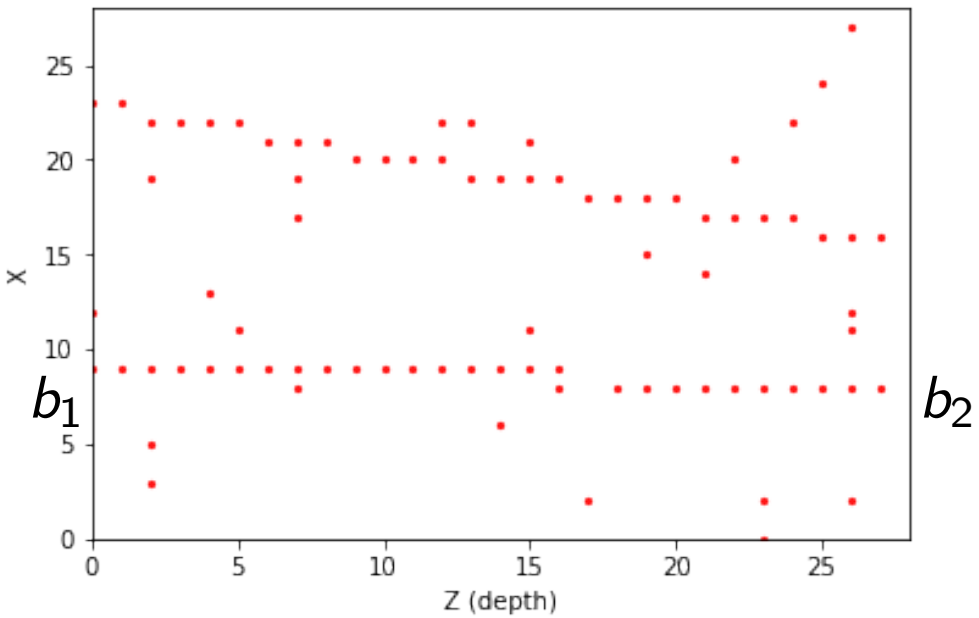
Toy Problem - Generating Data

- Generating images of size 28x28 pixels with straight lines and some random noise (matrix of 0s and 1s)



Toy Problem - Generating Data

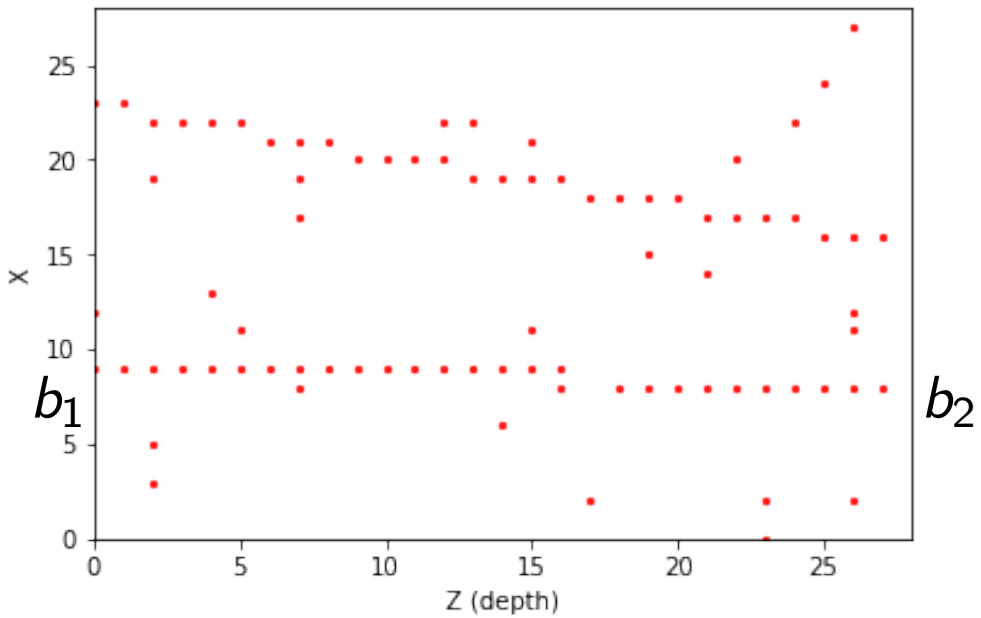
- Generating images of size 28x28 pixels with straight lines and some random noise (matrix of 0s and 1s)



- Training vector (single track): $[b_1, b_2]$

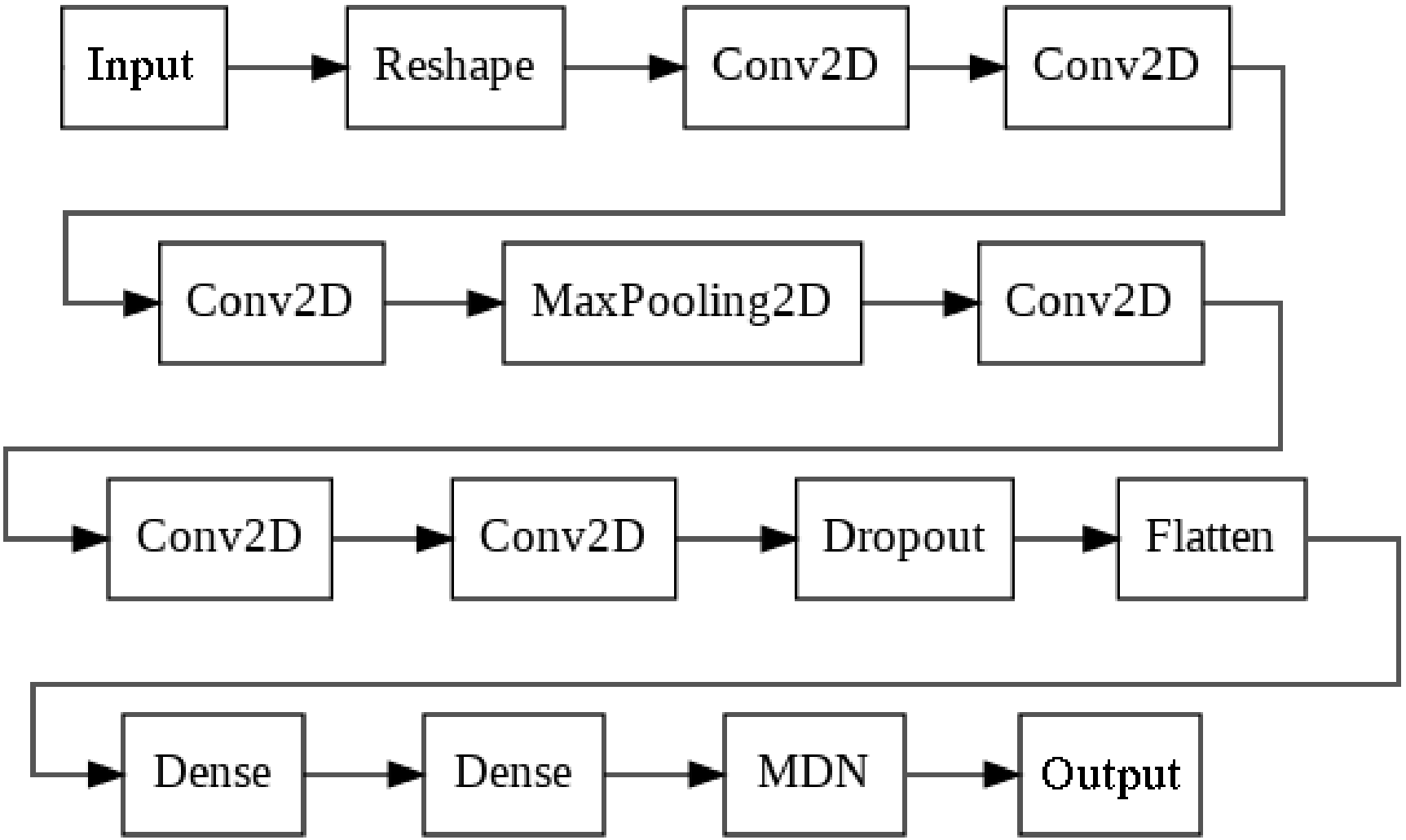
Toy Problem - Generating Data

- Generating images of size 28x28 pixels with straight lines and some random noise (matrix of 0s and 1s)

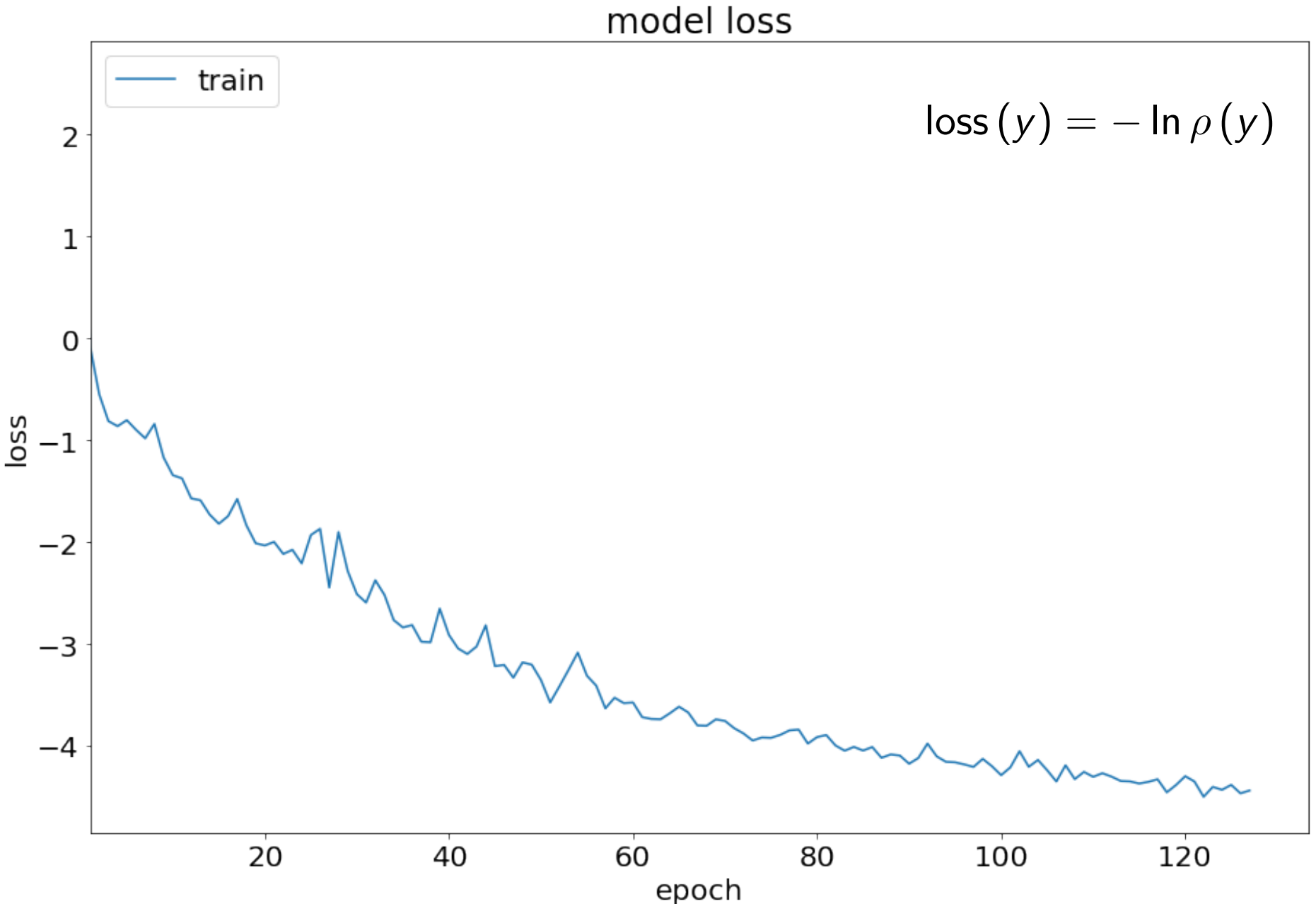


- Training vector (single track): $[b_1, b_2]$
- Prediction vector (3 mixtures):
  $\left[b_1^1, b_2^1, b_1^2, b_2^2, b_1^3, b_2^3, \sigma_1^1, \sigma_2^1, \sigma_1^2, \sigma_2^2, \sigma_1^3, \sigma_2^3, w^1, w^2, w^3\right]$

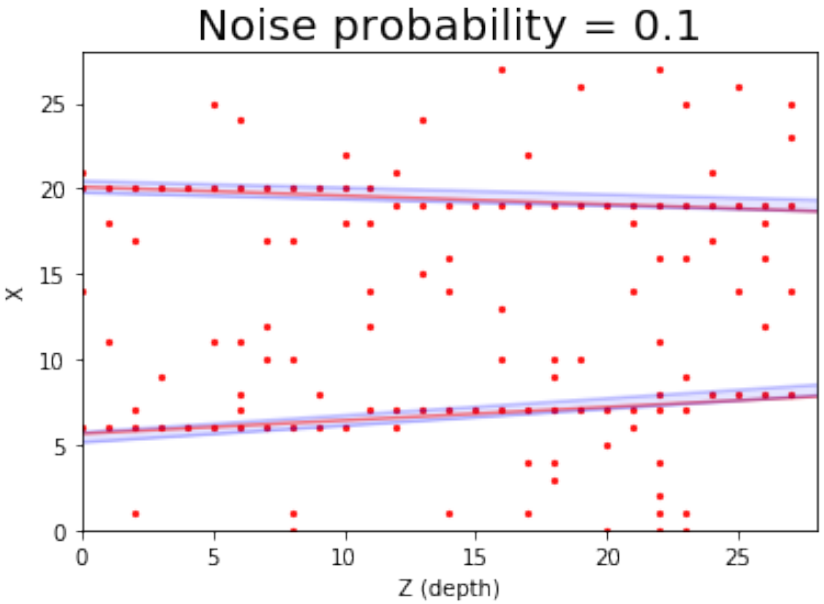
Neural Network used

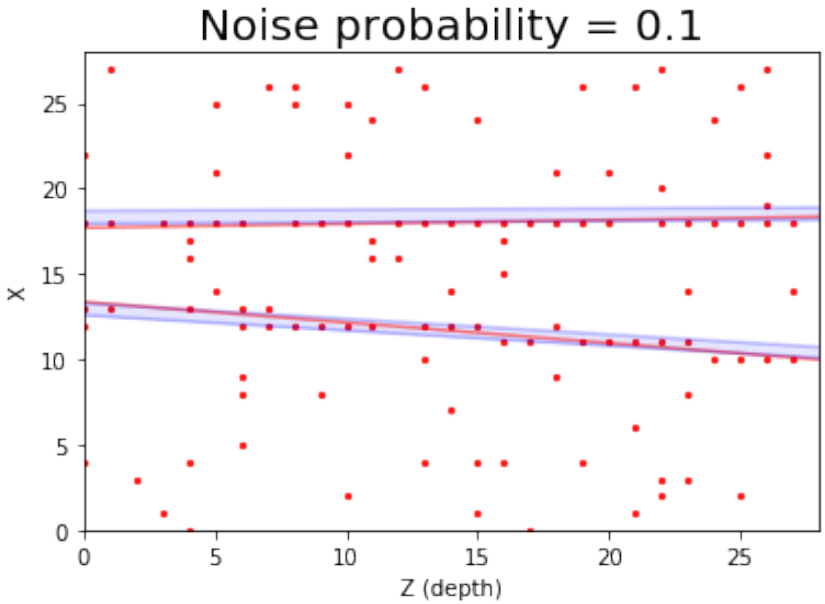


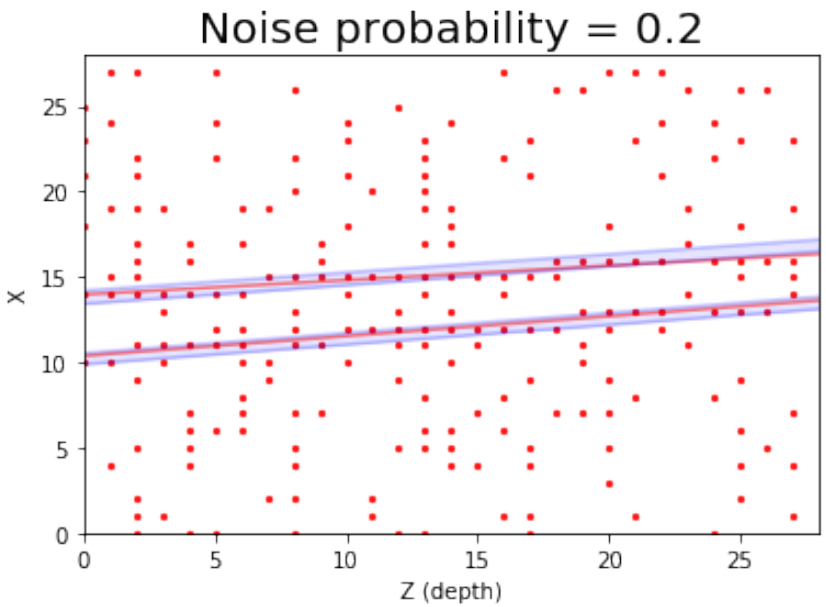
Learning process

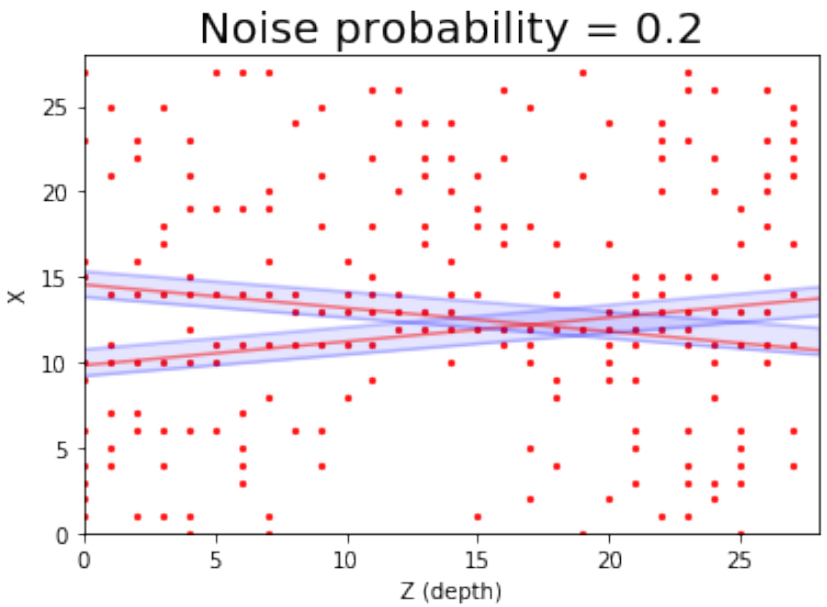


Toy Problem - Results

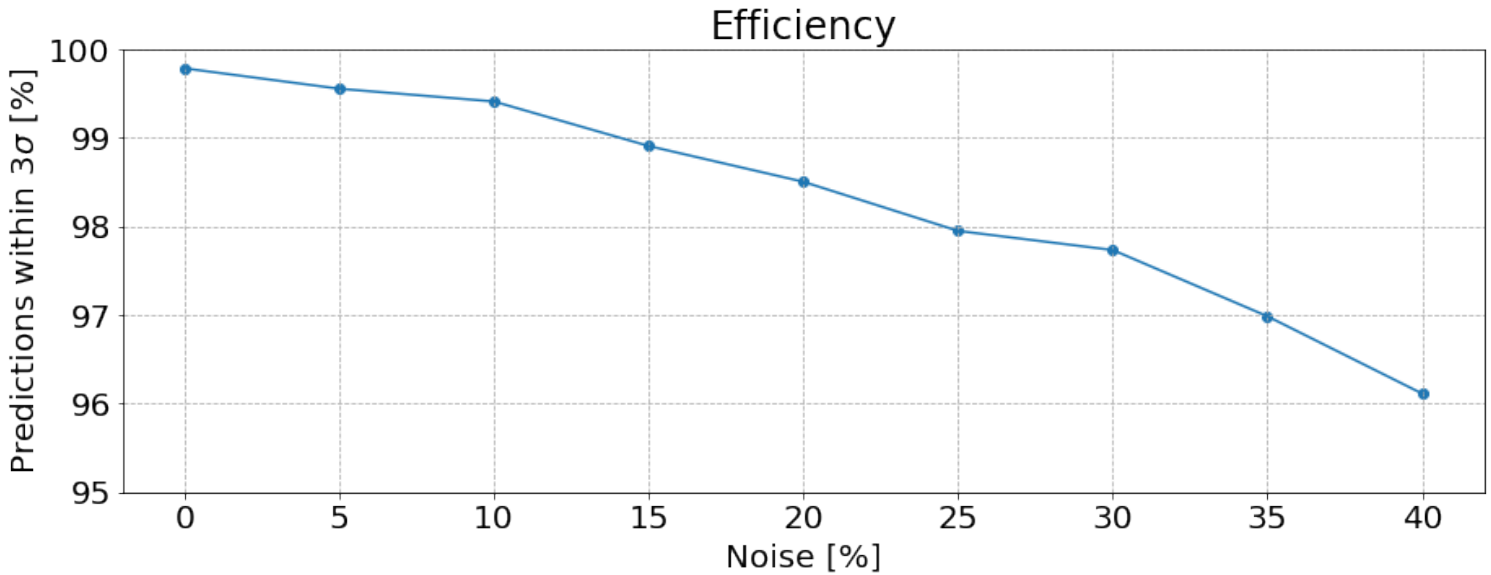




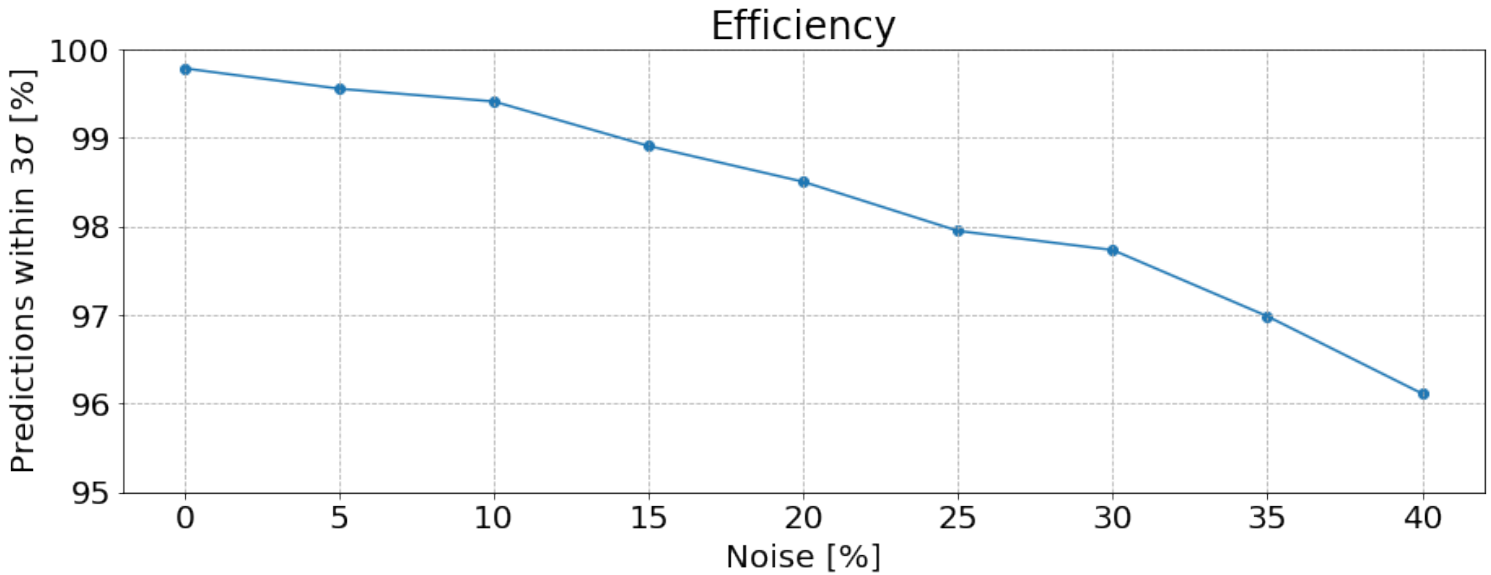


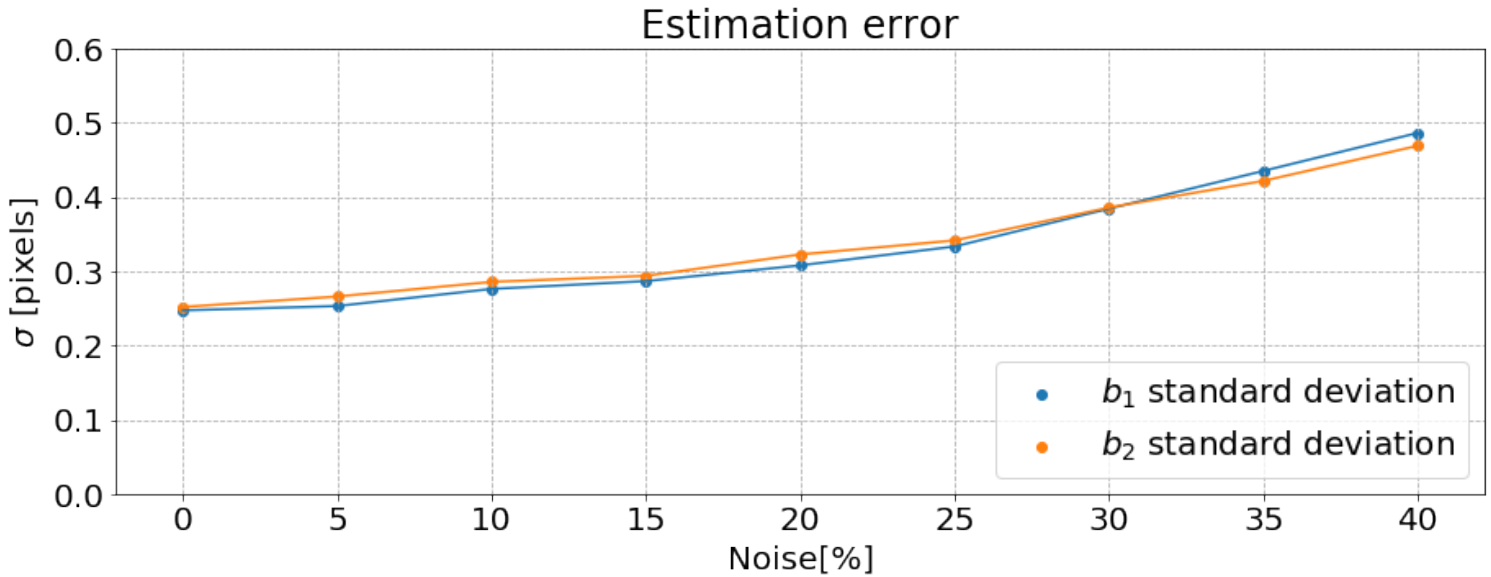


How does random noise affect prediction ability?

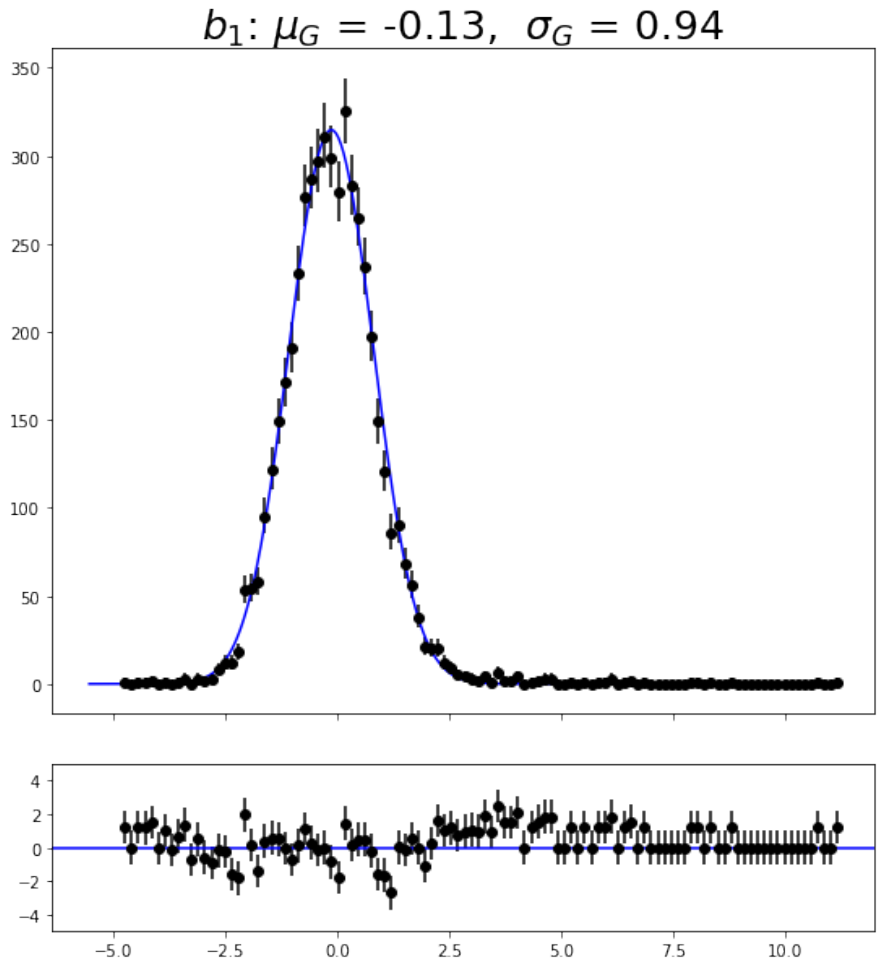


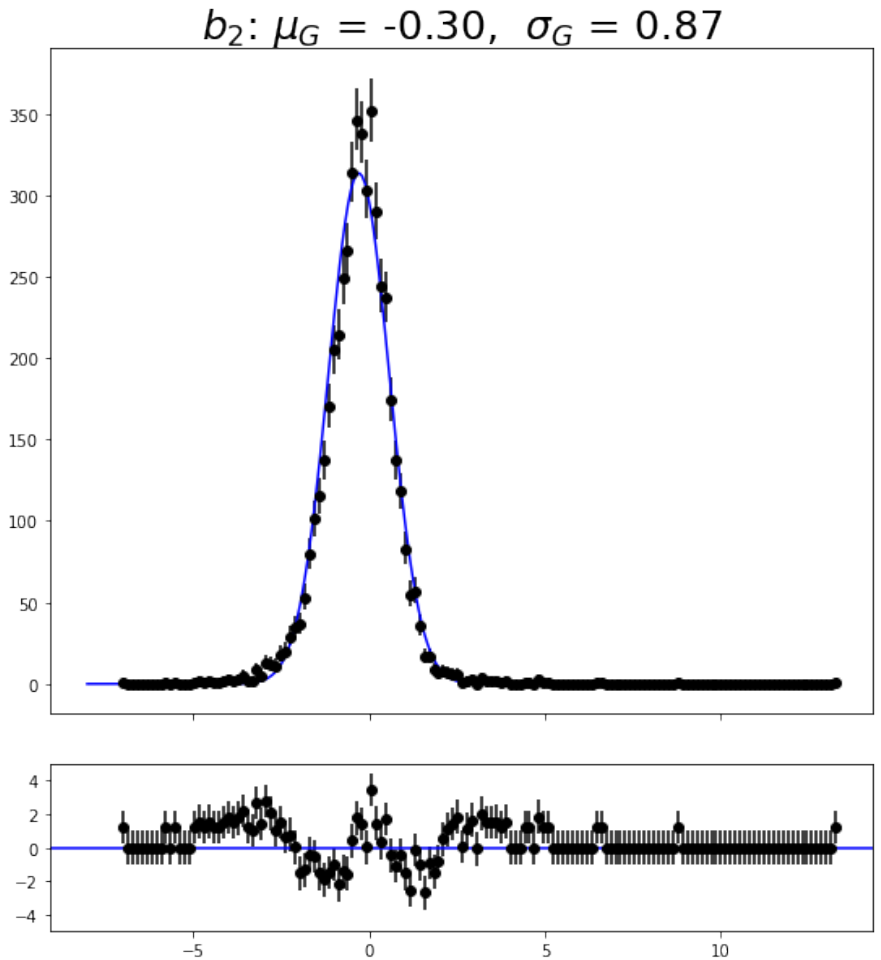
How does random noise affect prediction ability?





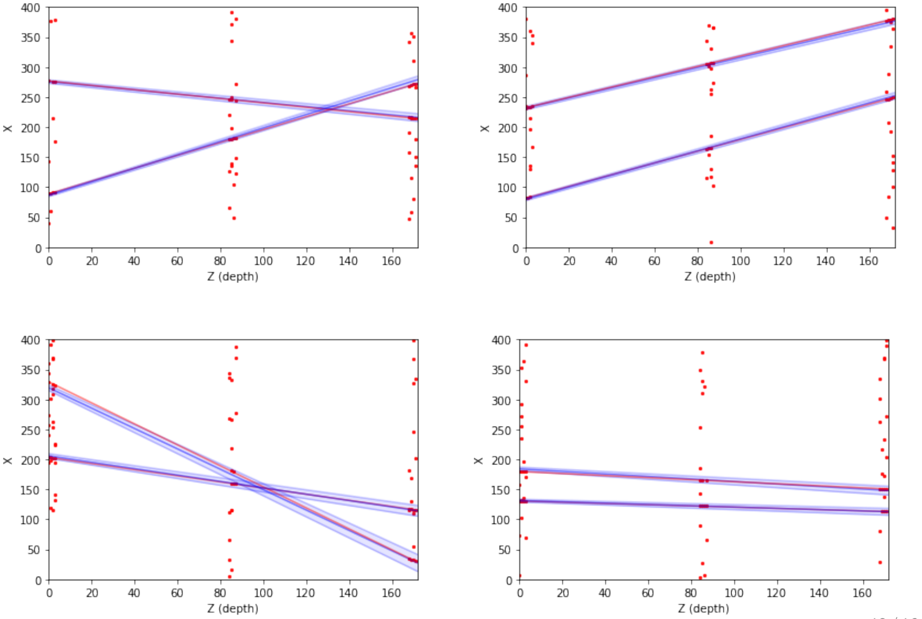
Pull plots - error estimation



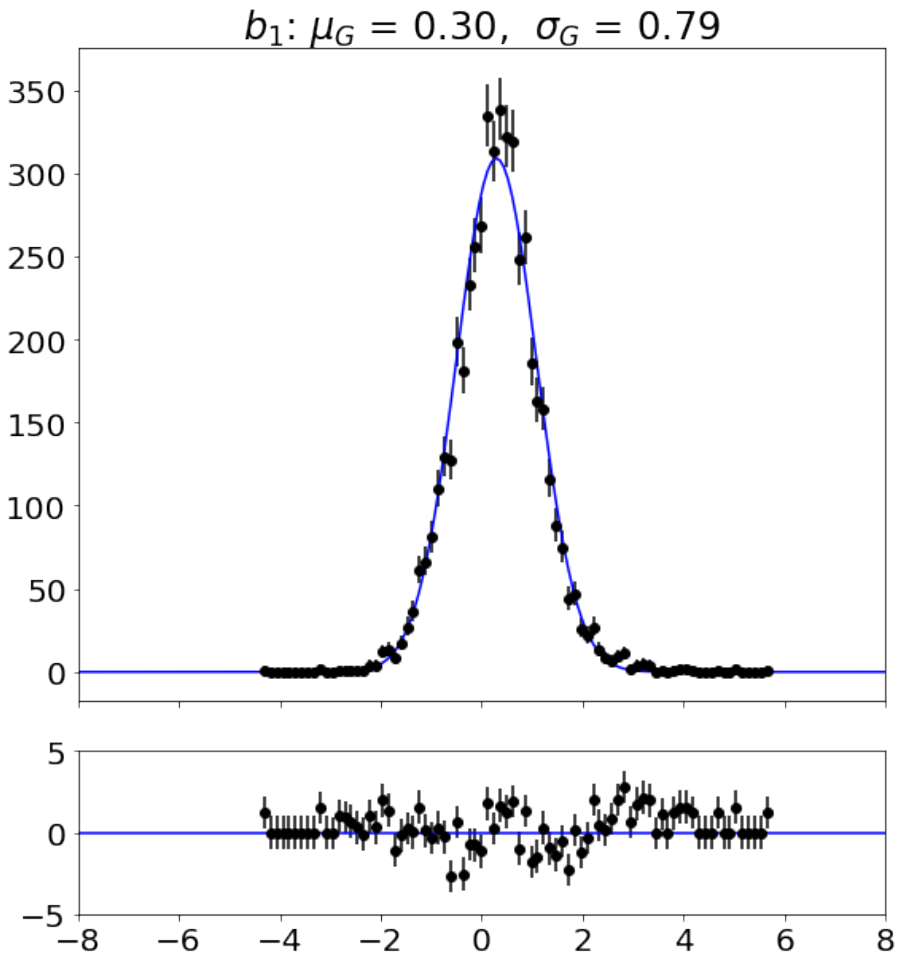


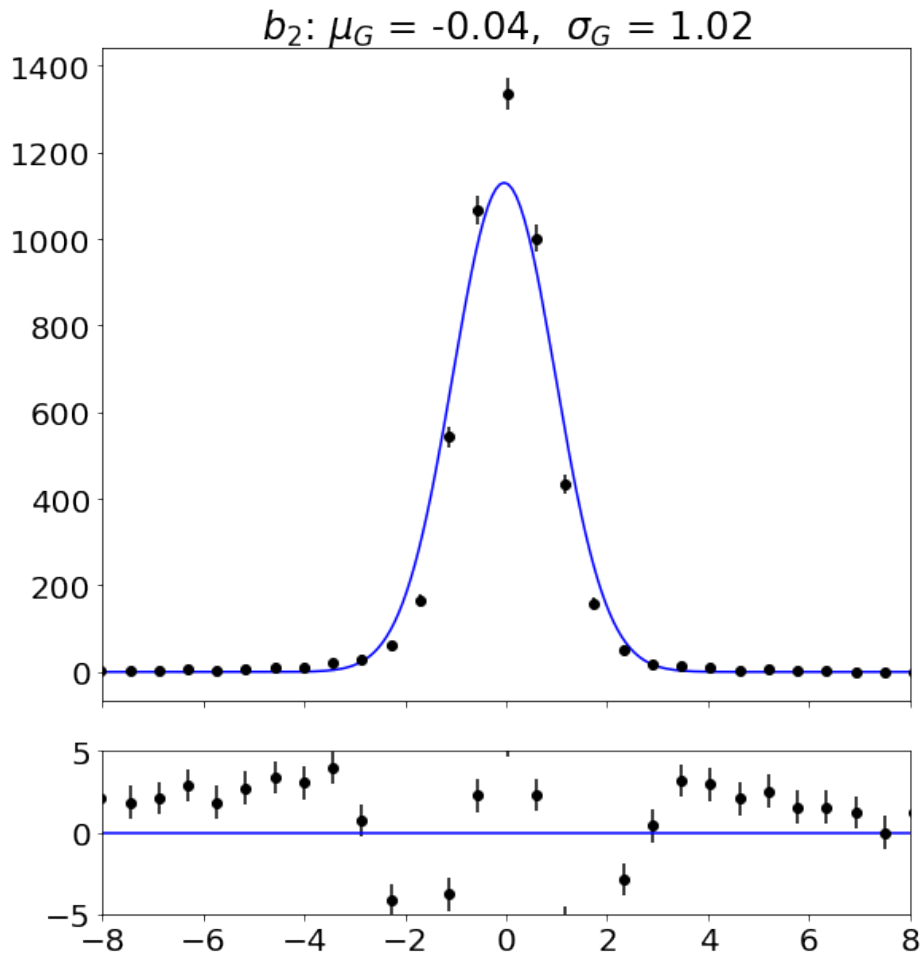
$$\frac{y_{true} - \mu}{\sigma}$$

Results - Realistic Data



Pull plots - realistic case





Potential future use

- There are plans to use neural network tracking in MuonE experiment operating at the SPS accelerator at CERN

Fig. 2: MUonE experiment logo[2]

Potential future use

- There are plans to use neural network tracking in MuonE experiment operating at the SPS accelerator at CERN

Fig. 2: MUonE experiment logo[2]

- Including magnetic fields $\rightarrow$ more layers

Potential future use

- There are plans to use neural network tracking in MuonE experiment operating at the SPS accelerator at CERN

Fig. 2: MUonE experiment logo[2]

- Including magnetic fields $\rightarrow$ more layers
- Extending the model to 3D

Technologies used

Keras

TensorFlow

python™

colab

Link to PPSS-Particle-Tracking Repository:
https://github.com/Matek1731/PPSS-Particle-Tracking

Questions?

Bibliography


Convolution arithmetic tutorial.


1st muone collaboration meeting at cern.


Christopher M Bishop.
Mixture density networks.
1994.


Cpmpercussion.
cpmpercussion/keras-mdn-layer.


Oliver Borchers.
A hitchhiker's guide to mixture density networks, Feb 2019.


Shaked Zychlinski.
Predicting probability distributions using neural networks, Jan 2019.


Guillaume Rabusseau and François Denis.
Learning negative mixture models by tensor decompositions.
CoRR, abs/1403.4224, 2014.

Backup

Loss - 400x12