UNIVERSITÉ DE GENÈVE

Physics Applications of AI

DArk Matter Particle Explorer (DAMPE) Particle Trajectory Reconstruction with Deep Learning

---

Student:
Kyril Kaufmann

Professor:
Prof. Tobias Golling

17 June 2024

Contents

1 Introduction

DAMPE stands for 'DArk Matter Particle Explorer'. The mission was launched on December 17, 2015 and is accumulating 2 billion cosmic-ray events per year.

The main scientific objective of DAMPE is to measure electrons and photons with much higher energy resolution and energy reach than achievable with existing space experiments in order to identify possible Dark Matter signatures.

The aim of this project is to reconstruct the particle trajectory in the BGO calorimeter. This will be done by looking at simulated readout of the calorimeter, treat those as images and apply deep learning algorithms for the regression task.

The actual cosmic particle trajectory reconstruction is performed by the Silicon-Tungsten Tracker-Converter (STK) detector. The STK detector is using the reconstruction of the calorimeter shower axis as a seed. The projection of the STK track onto the Plastic Scintillator Strips (PSD) detector is used for the calculation of the path length.

For the regression task of the shower axis estimate, different models, all based on convolutional neural networks will be developed and compared.

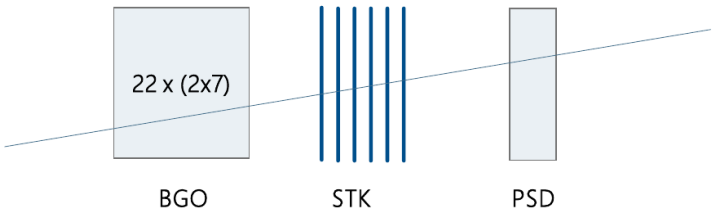

Figure 1: Schematic representation of the DAMPE sensors. BGO : Bismuth Germanium Oxide calorimeter. STK : Silicon-Tungsten Tracker-Converter. PSD : Plastic Scintillator Strips Detector.

2 Data and Processing

The dataset consists of simulated hits on the DAMPE calorimeter. Each cell of the calorimeter can be visualized as a pixel of an image. Together with the calorimeter readout we get the particle energy. Because of the hodoscopic arrangement of the calorimeter detector, following downward the z direction, the calorimeter readout alternates between x and y coordinates. A simulated particle hit on the calorimeter is called a shower. An example is given in figure 2.

As the data is being simulated, the true direction of the particle is known. The labeled for each particle shot is stored in a 4-dimension vector

$$X_{true} = \begin{pmatrix} x_b \\ x_t \\ y_b \\ y_t \end{pmatrix}, \tag{1}$$

where x_b, y_b correspond to the x, y of the particle at the bottom of the detector and x_t, y_t to the x, y of the particle at the top of the detector. Notice that the particle can come from any of the 4π sr direction. The goal is to infer from the available data the 4-dimension vector $\hat{X}$ corresponding to the x and y entry point and exit point of the particle.

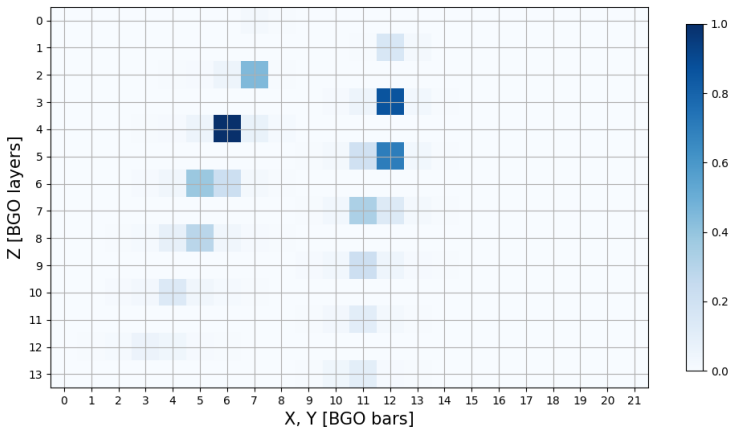

Figure 2: Example of a simulated shower. Following the z direction, figure x-axis is alternating calorimeter x and y readout. The pixel intensity is related to the particle energy reading. It is encoded on 256 bits and is normalized to 1 for each shower.

Together with the labeled points we get the benchmarked entry / exit $\hat{X}_{bench}$ from standard regression methods. This will be useful to asses the goodness of the developed models. A summary of the available data is given in table 1. When using the term data without specification, it is meant all available data and labeled output.

In ordered to try different modeling approach, the calorimeter image data is transformed

Data Type	Number of Items	Dimension
Calorimeter Image	141'946	14x22
Energy	141'946	2x1
Labeled Coordinates	141'946	4x1
Benchmarked Coordinates	141'946	4x1

Table 1: Data summary. All data element is red as python numpy arrays. The Data Type energy is encoding both the total energy and the detector maximum readout.

as to gather the detector x readout at the top of the image and y readout at the bottom of the detector. This can be clearly visualized in figure 3.

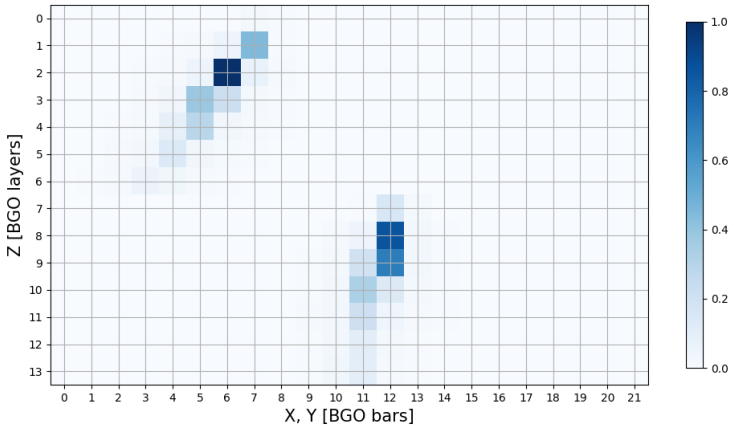

Figure 3: Ordered shower of figure 2. The first half (reading from top to down) corresponds to the x readout and the second half to the y readout of the detector.

3 Neural Network Models

As per standard machine learning regression convention, the data is shuffled numerically uniformly and then split in training (80%), validation (10%) and test (10%) sample sets.

The image data inputs being encoded on a $14 \times 22 = 308$ 8-bits pixel correspond a significantly big input. Combined with a relatively small dataset this suggest that direct feed-forward neural network would most likely not perform well while taking a lot of computational power. Moreover visual analysis of the particle shower shows that only a few pixels are colored and that those colored are close to each others, following the line of the particle trajectory. This closeness and feature aspect brings to the choice of convolutional neural networks (CNN).

The choice of the CNN is not obvious. Different architecture where tried and each will be presented in this section. For all the models we tried to keep the number of parameters around 40'000 - 50'000. Not more than twice less than the training data set size. Although there is no precise rule to support that fact, this should ensure no over-fitting.

The target for the training used for all the developed models is the mean squared error

$$L\left(\hat{X}, X\right) = \frac{1}{N} \sum_{i=1}^{N} \left\| X_i - \hat{X}_i \right\|^2, \tag{2}$$

with the sum going over all the training labeled trajectory.

In theory, all models would work with the calorimeter images given raw (figure 6) or reordered on x and y (figure 3). In practice, models 1 to 4 are trained on the raw images while model 5 is trained on the reordered image. For models 1 to 4, the first convolutional layer filter is taken of size (5,4). This is done in order for the filter looking at a pixel representing the x projection (respectively y projection) to convolve with neighboring x (resp. y) pixels. Whilst when the image is ordered (Model 5), the initial convolutional filter can be taken smaller.

Model 1 - Including Maximum pooling

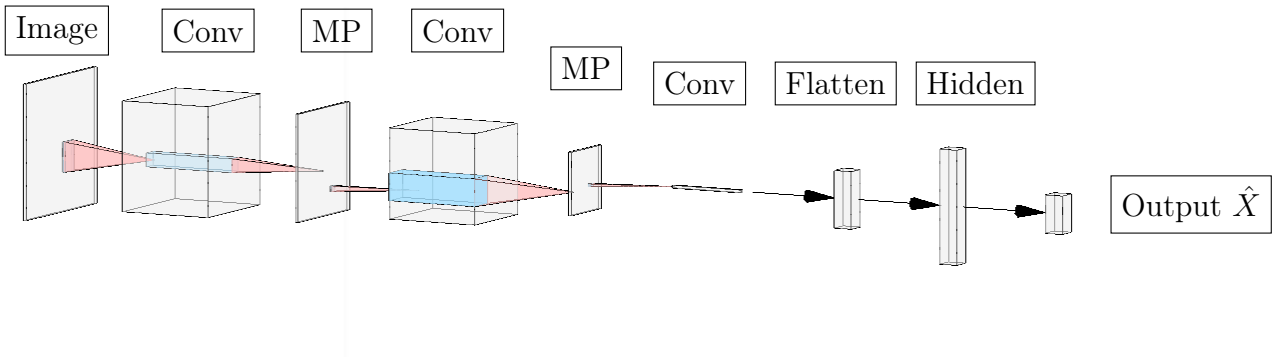

Figure 4: Schematic representation of Model 1

The first model consist of a succession of convolutional layers and Max Pooling (MP) layers. The last layer of the convolutional part of the network consist of applying a wide filter and then flatten the data to a single vector representation. The filter is chosen wide in order to reduce the amount of inputs to the dense network. The flatten vector is then fed to a feed forward MLP network with a single hidden layer of 64 neurons. Two 64 neurons layers were tried resulting in poorer performance.

A representation of the model is given in figure 4. The idea of using maximum pooling is to reduce the dimensionality of the intermediate filtered image while applying bigger filters at the convolutional layers.

Model 2 - Without Maximum pooling

The second model doesn't make use of maximum pooling. Instead the filter width is chosen bigger to keep the layer fed to the feed-forward MLP network to a reasonable size. A representation of the model is given in figure 4.

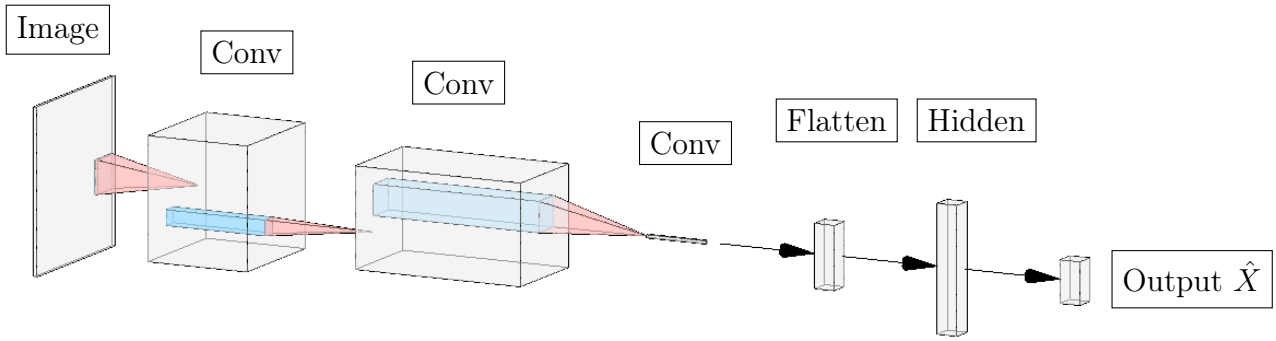

Figure 5: Schematic representation of Model 2

Model 3 - Splitting the data based on total energy

In figure 6 one can see the distribution of the particle total energy and calorimeter bar maximum energy. The wide distribution of energy suggest to use different model at different range of energy. For simplicity we choose to split the total energy at the median and build a model for showers with lower energy and another one for higher energy.

The model architecture is chosen the same as for the model 2 (figure 5) except that the convolutional filters are chosen smaller in order to keep the number of parameters twice lower than the number of input. Indeed, splitting the data set comes with the drawback of reducing the amount of training data available to the model.

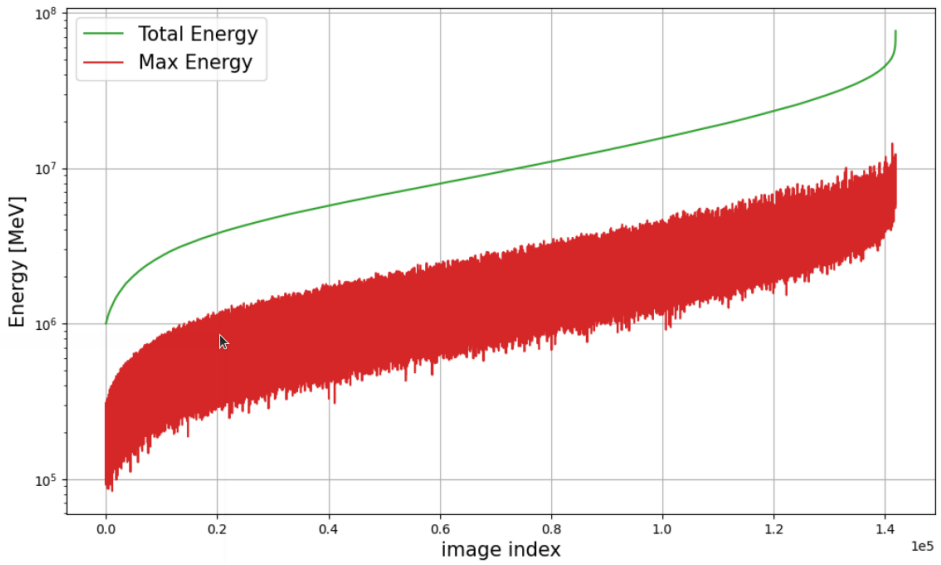

Figure 6: Calorimeter sample energy distribution. Green - Shower total energy. Red - Calorimeter bar maximum energy.

Model 4 - Including energy

Rather than splitting the data on higher and lower energy shower, the energy of the particle can be directly fed to the network. The network architecture used for Model 4 is the same as for Model 2 with the only difference that the total particle energy and the calorimeter bar max energy are concatenated with the flatten output of the convolutional network and fed to the feed-forward MLP network. Both the total energy and bar maximum energy are normalized to 1 with respect to the overall data maximum. This is to keep the same order of magnitude as the calorimeter images inputs.

Model 5

The last model has an architecture without maximum pooling similar to the Model 2. The main difference being that the model is applying a convolutional filter of width $(3, 5)$ at its first layer. The model is meant to work with the reordered data.

4 Calorimeter Particle Trajectory Reconstruction

The evolution of the mean squared error in training for the 5 developed neural network are shown in figure 7. The training was performed on 110 epochs with mini-batch stochastic gradient descend with batch size of 32 on a 8-core intel i5 8th generation CPU.

The training didn't reach a global minimum. Increasing the number of epochs would be necessary. However with the quality of the computational power at hand we thought not wise to do so.

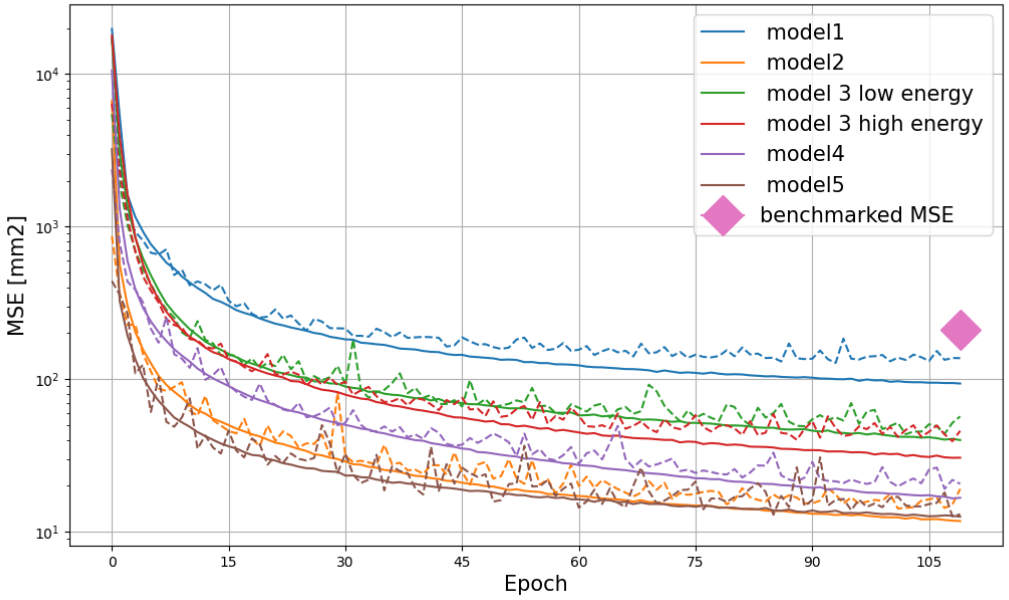

Figure 7: Models training history comparison. The training was performed on 110 epochs using mini-batch stochastic gradient descend. Full lines are corresponding to the training data error. Dashed lines to the validation data error. Diamond correspond to the mean squared error computed on the benchmarked data.

5 Conclusion

Deep learning based models for calorimeter particle reconstruction were successfully developed. The reconstruction is reaching error bellow 15 mm^2 on the combined 4-parameter x, y particle calorimeter entry/exit point.

The 110 epochs training didn't reach a global minimum. The project having educational purpose and due to the lack of computational power, we didn't aim to push it further.

All the developed models are performing better than the model used for the benchmark data.

Model 1 that works with maximum pooling shows the poorest performance. This is not surprising as the calorimeter images being coarse are showing very sharp edges with high pixel intensity where the particle passed. The maximum pooling operation basically destroys this sharp edge information.

It is surprising to see that Model 4 which is making use of the particle energy directly as an input to the MLP network is not performing better. Some potential explanation would be that the energy normalization was done improperly, or that the energy would require some weighted regularization in order to account differently for the energy direct input and the CNN output. Last, it could be that the MLP network was not designed with enough layers to properly train on the direct energy inputs.

Model 3 that is splitting the particles in lower and higher energy group shows to perform less good. It is not clear weather the splitting of energy is itself not making any improvement or the poorer performances are due to lesser parameters in the model and smaller training database.

Generally, as further improvement we make the following suggestions :

- The calorimeter image dimension is of size $14 \times 22 = 308$. So the total number of input is of size $308 \times 141'946 \approx 43.7 \cdot 10^6$. This suggests that the models developed have a total number of parameter quite small compared to the byte-wise input size and that they might underfit. It would be interesting to try CNNs with more parameters. This was not done due to the lack of computational resources.
- The particle hitting the calorimeter forms a 3-dimensional problem (x, y, z). Especially the y trajectory is correlated to the x one. The use of transformers could add attention between x and y projection leading in improved performances.