Project Report for Course 14P053: Physics Applications of AI

Arshia Ruina

University of Geneva

June 2022

Contents

1 Introduction

The report presents the author's work on the project topic titled

Incidence Angle Prediction of Astro-Particles passing through a Detector

where machine learning techniques introduced in the course, especially regression methods, have been studied and used to make the predictions. This section very briefly introduces the experiment and detector setup on which the provided data is based, regression analysis in machine learning and its use in this work. The methods used to achieve the aim of this project, the corresponding results, challenges and future prospects are detailed in the sections that follow.

1.1 The DAMPE Detector

The DArk Matter Particle Explorer (DAMPE) is a space-based particle detector that is orbiting the Earth in a sun-synchronous orbit at an altitude of about 500 km [1]. It has been in operation since 2015 and is aimed at studying high-energy gamma rays and cosmic nuclei fluxes. The detector system, as shown in figure 1, comprises a plastic scintillator detector (PSD) for charge measurement, a silicon-tungsten tracker-convertor (STK) for reconstructing the trajectories of incident particles, a bismuth-germanium oxide (BGO) calorimeter for energy measurement and a neutron detector (NUD) to further aid in hadron identification.

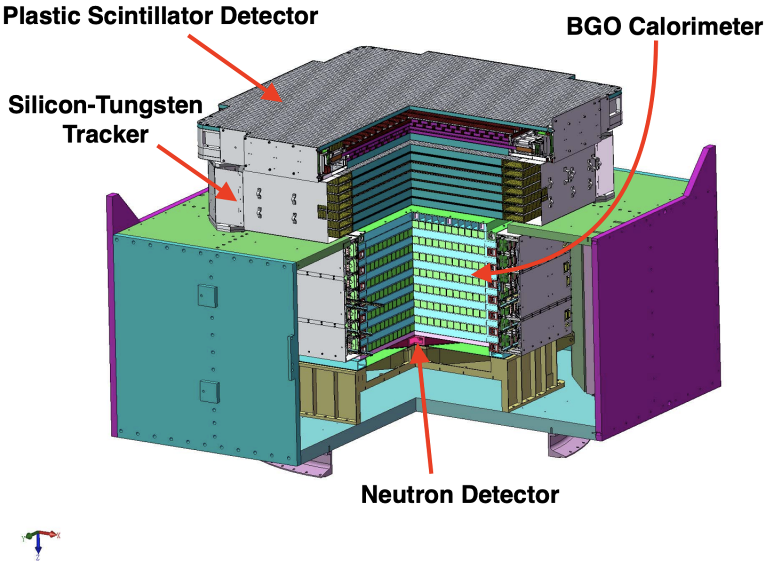


Figure 1: A schematic of the DAMPE detector system [2]

As an incoming particle hits the detector, it passes through the different sub-detectors (depending on its trajectory and energy) and gets slowed down when it starts interacting with the material of the detectors. These interactions usually give rise to what are known as particle showers inside the detector, that are characteristic to the incident particles. The BGO calorimeter [3] records such showers in such a way that we are able to reconstruct the shower topology, the energy deposited by the incoming particle and its direction. It is made up of 14 layers, where every layer has 22 BGO crystal bars, each of which has a dimension of 25mm x 25mm x 600mm (see figure 2). Scintillation light is detected at both the ends of a crystal bar by two photo-multiplier tubes. The layers are arranged orthogonal to one another which makes it possible to reconstruct the shower shape in all three dimensions.

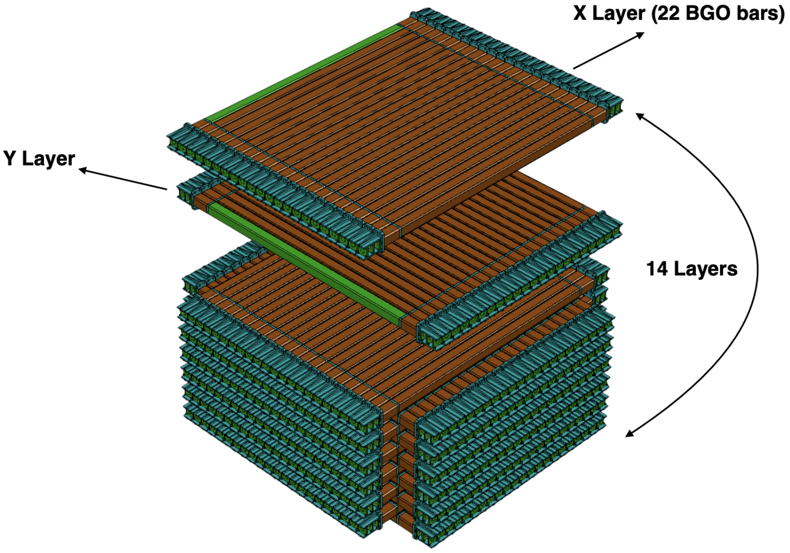


Figure 2: A schematic of the BGO calorimeter [2]

1.2 Regression Techniques in Machine Learning

Regression analysis is the prediction of continuous outputs that is dependant on certain input variables (as opposed to assigning discrete labels in classification problems). In machine learning, the aim is to let the network predict this relation (between the inputs and outputs) and to obtain the best possible results by minimising the losses (which could be, for example, the squared differences between the input and predicted values).

1.3 The Project

In this project, the provided data consisted of (simulated) calorimeter images and energies measured with DAMPE. Each image is made of 14x22 pixels so each pixel represents a hit in a calorimeter bar. The pixel values range from 0-255, representing a single channel where the intensity is proportional to the hit energy. The aim is to design a regression network to predict the x- and y-coordinates of the incident particle at the top and bottom of the calorimeter using the given information.

2 Methods

2.1 Understanding the Data

Before blindly embarking on designing a neural network to throw out some predictions from a given set of numbers, it is essential that the data itself is well-understood by the developer, as this can affect the design of the network, its efficiency and accuracy. A considerable amount of time was devoted to understand the provided data and be able to visualise it. Figure 3 shows the first four calorimeter images. On these images, or plots, the y-axis goes from 0 to 13, indicating the 14 layers of the calorimeter while the x-axis ranges from 0 to 21 that represent the bars in each layer. Each pixel value corresponds to the ratio of the energy deposited in that bar to that of the maximum-energy bar for the given event. This keeps the range of values between 0 and 1, with an 8-bit precision, which means that each pixel can hold a value between 0 and 255 (inclusive).

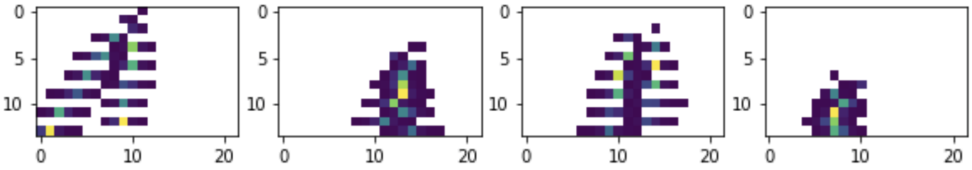


Figure 3: The first four calorimeter images in the provided dataset

However, it is crucial to note that these are 2D images of a 3D calorimeter – the XZ and YZ views are superimposed on each other. Put together in this way as a single image, they do not provide a good picture of a particle shower or direction. Here, we already see an example of how understanding our data will help us in providing more sensible input values to a neural network. It becomes easier to visualise the shower topology when this combined view is broken down to separate XZ and YZ views as shown in the figure 4. It is expected that the maximum energy deposits will be along the particle trajectory. This is evident in these figures where, in a given layer, the centers of all the clusters of hits have pixel values closer to the maximum value (255, yellow) indicating that higher energies were deposited in those bars as the particle passed through each layer.

The total and maximum energies for each calorimeter image are also provided in the dataset. A distribution of these quantities over the entire dataset is shown in figure 5. This information can also be an important input to the network as, from physics, we know that if the particles with higher incident energy will undergo fewer interactions in the detector material, resulting in a narrower shower profile while for lower incident energies, the showers are expected to be more spread-out. This relation can also be quantified by using the ratio of the maximum energy to the total – the higher this value is, the narrower is the shower and vice versa. However, these two variables will not be used as direct inputs to the network.

Also provided in the dataset are the true values (they exist because the data is simulated) of the target variables – the x- and y-coordinates in the top and bottom planes of the calorimeter (figure 6).

The distributions of these coordinates over the entire dataset is shown in figure 7. Noticeably, the distributions are narrower for the top plane than for the bottom. The reason behind this is two-fold: in the DAMPE detector geometry, the positive z-axis points in the direction of the NUD from the STK (see 1), so the top layer is the one closest to the NUD while the bottom layer is

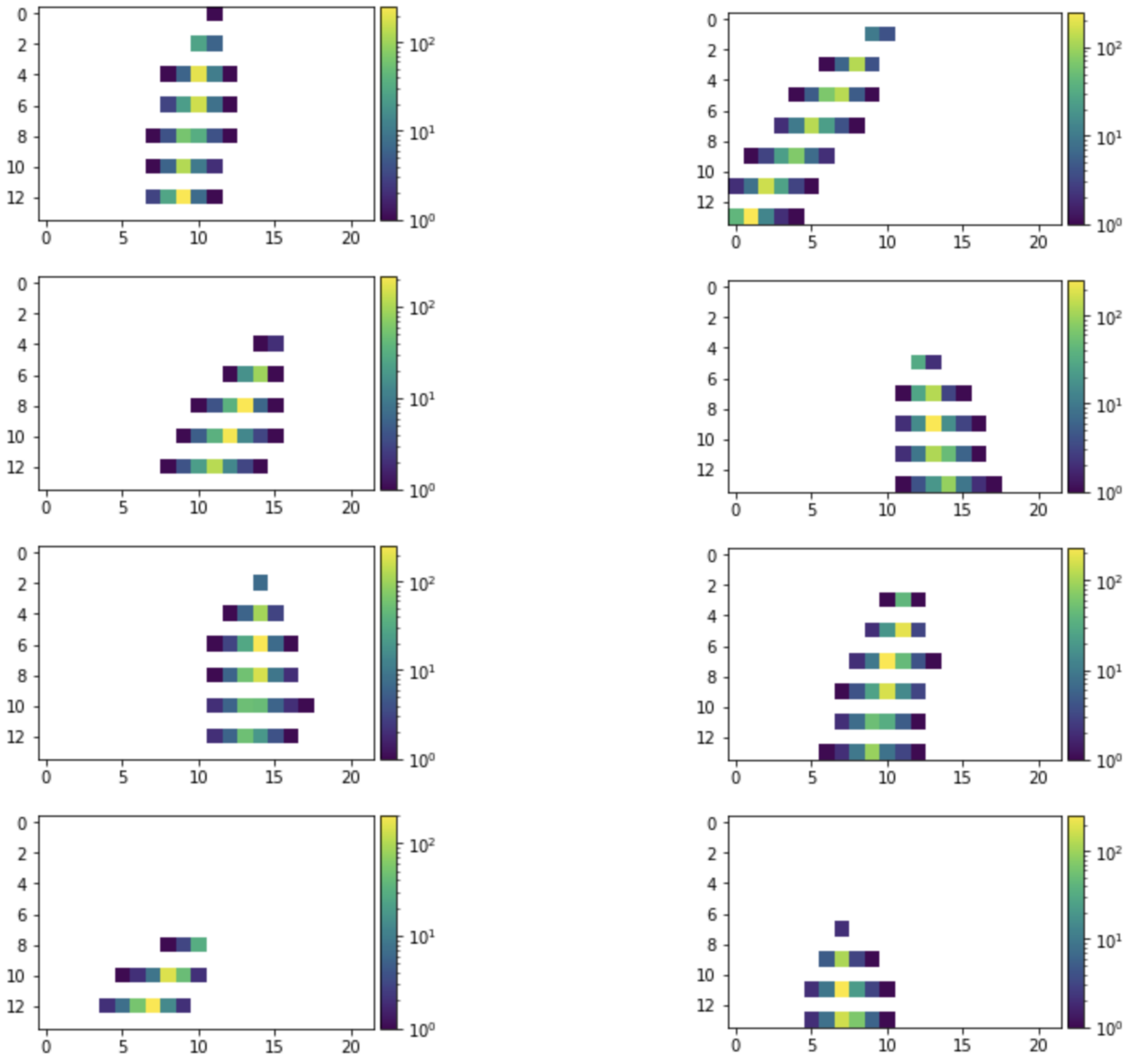


Figure 4: YZ (left) and XZ (right) views of the calorimeter images

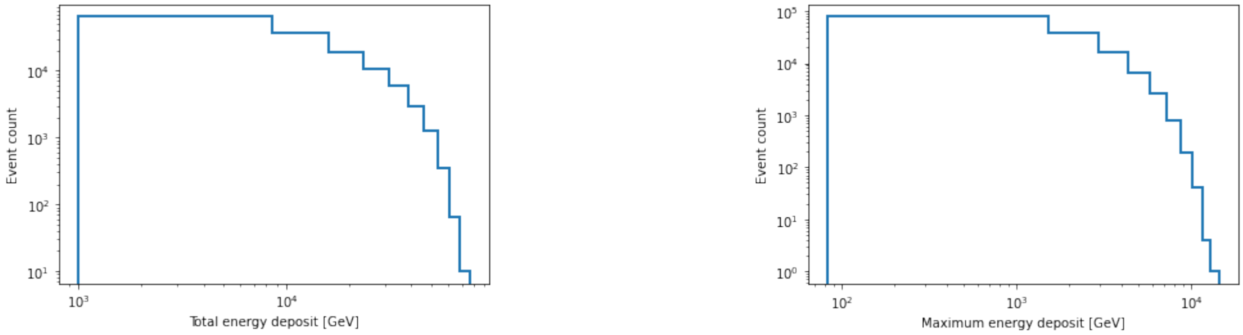


Figure 5: Distributions of the total (left) and maximum (right) energies as deposited in the calorimeter

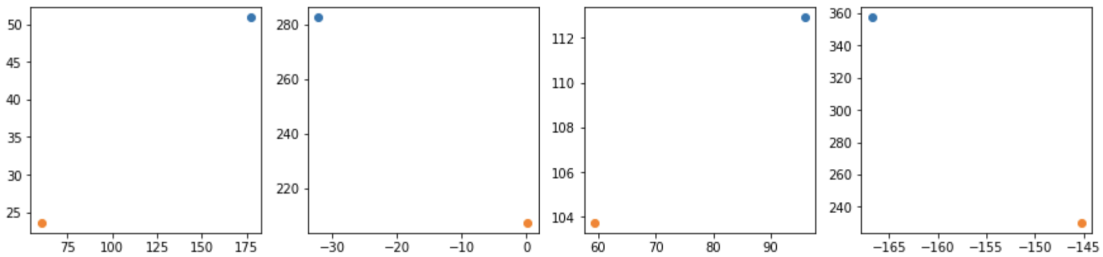


Figure 6: The target *x*- and *y*-coordinates (in units of mm) for the first four events in the provided dataset. The blue points are on the bottom layer and the orange points are on the top layer.

the one closest to the STK. As such, the fiducial volume of the detector (where a track has passed all the sub-detectors) excludes the very edges of the top plane. This is also visible in the 2D representations of these distributions as shown in figure 8.

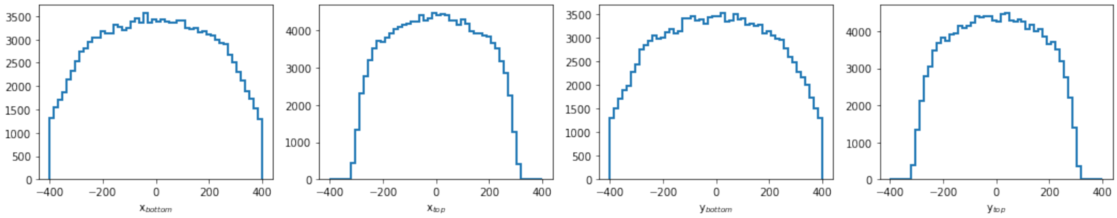


Figure 7: Distributions of the target *x*- and *y*-coordinates (in units of mm) in the top and bottom planes of the calorimeter

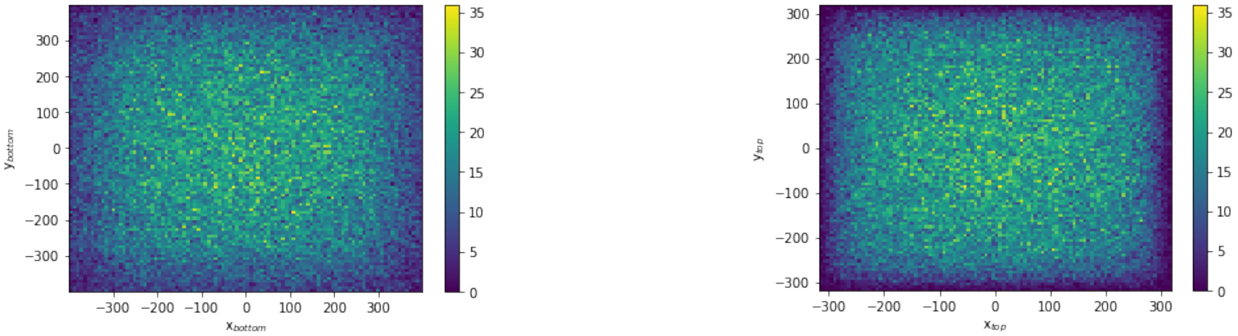


Figure 8: Distribution of the target *x*- and *y*-coordinates (in units of mm) in the bottom (left) and top (right) planes of the calorimeter

2.2 Network Architecture

Initially, a network architecture comprising two convolutional layers, one pooling layer and one final dense layer with four outputs was designed. The total dataset size is 141946, 80% of the data was used to train the network and 20% was set aside for validation. This architecture took as input the combined calorimeter images (with no separation between XZ and YZ views). For the

Conv2D layers, the number of filters was chosen as 50 and 20 with a kernel size of 4x4 and strides of 1x1. Varying the values of these three parameters with everything else the same did not yield any substantial improvements in the validation loss.

Loss functions. A comparison was made among the different loss functions – mean squared error (MSE), mean absolute error (MAE) and logcosh[1]. The latter two had very similar performances, albeit had much lower absolute values than MSE (figure 9).

Optimizers. The Adam optimizer was used with a learning rate of 0.001. Referring to an online resource [4] further supported its use, it is one of the most reliable adaptive optimizers available where a fine tuning of the learning rate by hand is not necessary. It updates the learning rate according the the parameters and also takes into account the history of past gradients (stores an exponentially decaying average of past gradients).

Epochs and batch-size. The number of epochs was kept well below 20 as the network was being trained on simple Jupyter notebook on a local machine (with no GPU cards or HPC access). The batch-size, which defines the number of samples that would be propagated through the network, is also an important parameter which was set to somewhat unreasonable values.

Early stopping. Despite using a low number of epochs at this initial stage, the early stopping callback was explored [5]. This is used to stop the training when a given metric has stopped showing improvements. The validation loss was monitored with a very low `min_delta` requirement of 1 and a `patience` value of 3, which meant that if the reduction in validation loss was less than 1 for 3 consecutive epochs, the training would be stopped.

A minimally optimized network design, as the one described above, gave promising results as far as the validation loss performance was concerned (figure 10). Over 20 epochs, the validation loss followed the training loss quite well, with no signs of overfitting and eventually plateaued out.

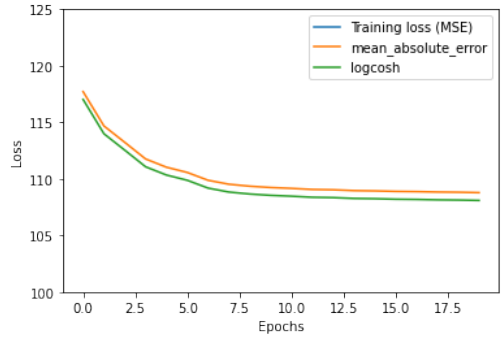


Figure 9: Mean absolute error and logcosh as loss functions (MSE is way beyond the limits of the plot)

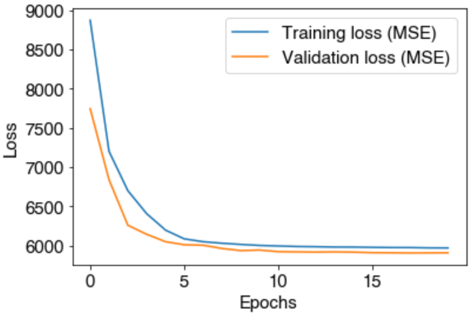


Figure 10: Training and validation losses from a very initial network architecture

However, to judge the performance of a model by just this parameter, or the 1D distributions

[1] Huber loss was not explored at all, it was, somehow, not available with the TensorFlow version that was being used

(as shown in figure 11) would be quite naive. Indeed, it became evident when the predicted x- and y-coordinates were plotted in 2D (figure 12) and compared with figure 8.

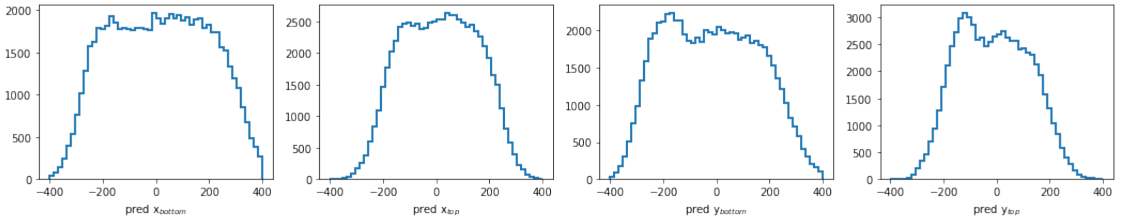


Figure 11: Distributions of the predicted x- and y-coordinates (in units of mm) in the top and bottom planes of the calorimeter with an initial model

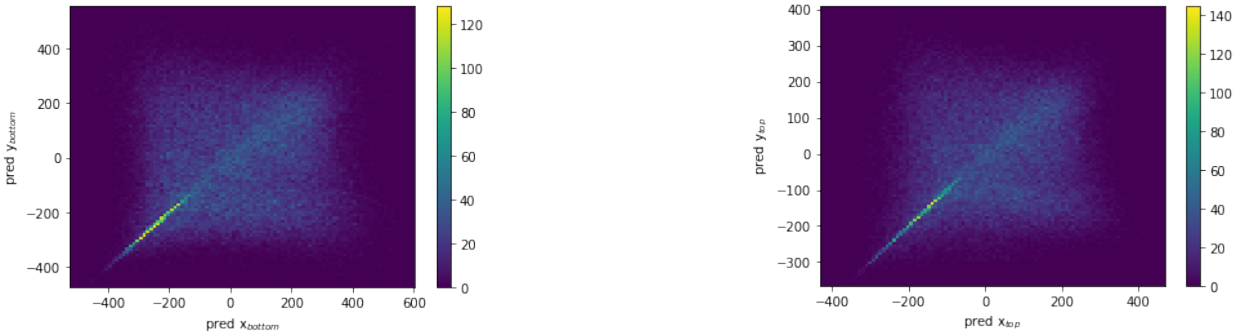


Figure 12: Distribution of the predicted x- and y-coordinates (in units of mm) in the bottom (left) and top (right) planes of the calorimeter with an initial model

Over the next couple of iterations, a number of changes were made to the network architecture and final[2] graph is shown in figure 13.

Datasets. The dataset was divided into 60-20-20 for training-validating-testing. Validating a model is not the same as testing it. After validation, the model should not be tested on the data that it has already learnt from as this might lead to biases. The final model, after all the fine-tuning procedures and iterations, must be tested on completely unseen data. However, the testing of the model has not been presented here.

Inputs. The input calorimeter images were separated into the XZ and YZ components. Then, a concatenation layer is added to merge them together before the application of convolutional layers.

Convolutional layers. The number of filters was increased on each layer and and extra third layer was added. The kernel-size was kept at 4x4. No change was made to the pooling layer.

Epochs and early stopping. With the availability of GPU cards, albeit for a limited period of time, the number of epochs was raised to 150-200. Consequently, early stopping was now showing its effects and most models stopped training at epochs less than 150. The `patience` value was increased to 5 and the `min_delta` was increased to 10.

Batch size. The batch size was reduced to 1000 on the suggestion of course assistants.

[2] It is , of course, misleading to say “final" here because this is only the best network obtained thus far, with many limitations and constraints playing a role!

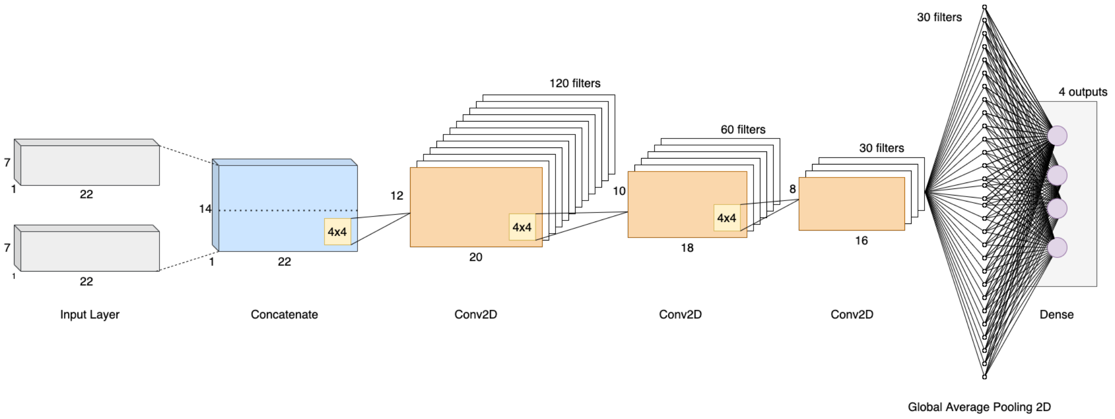


Figure 13: The network architecture (drawn by author using an online tool [6])

3 Results and Discussion

After fine-tuning the hyper-parameters, the final predictions on the validation data are shown in figures 14 and 15. When comparing the latter two with their target counterparts shown and referred to earlier, please keep note the change in plotting ranges.

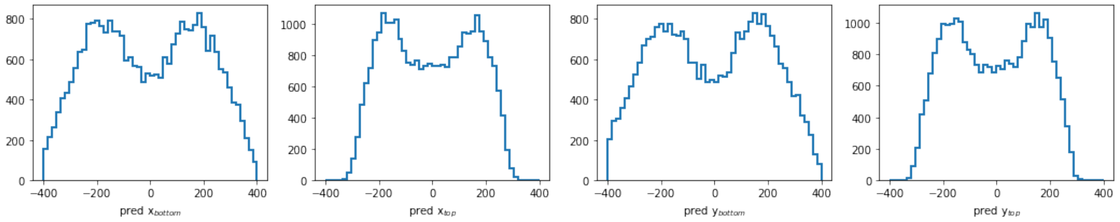
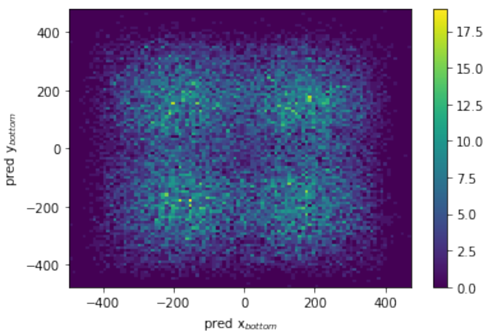


Figure 14: Distributions of the predicted x- and y-coordinates (in units of mm) in the top and bottom planes of the calorimeter with the final model



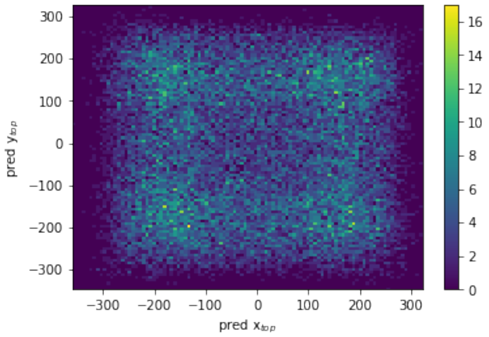


Figure 15: Distribution of the predicted x- and y-coordinates (in units of mm) in the bottom (left) and top (right) planes of the calorimeter with an final model

These predicted coordinates are a big improvement from the initial ones in their overall distribution but the work far from completion. The dataset is sparse but there seems to be a clear bias towards certain regions in both the calorimeter planes, namely the four corners. Ideas on how to tackle this have not yet been explored as part of this work.

In fact, another idea was to do regress the four coordinates individually, separately, and see how that changed the performance of the model. This too has remained an unexplored area.

Since this is simulated data, we have information to reconstruct the true trajectories of the incoming particles and compare them with the predicted trajectories. This will also be sensible way to quantify the accuracy of the predictions given by the model. A few such comparisons are shown in figure 16.

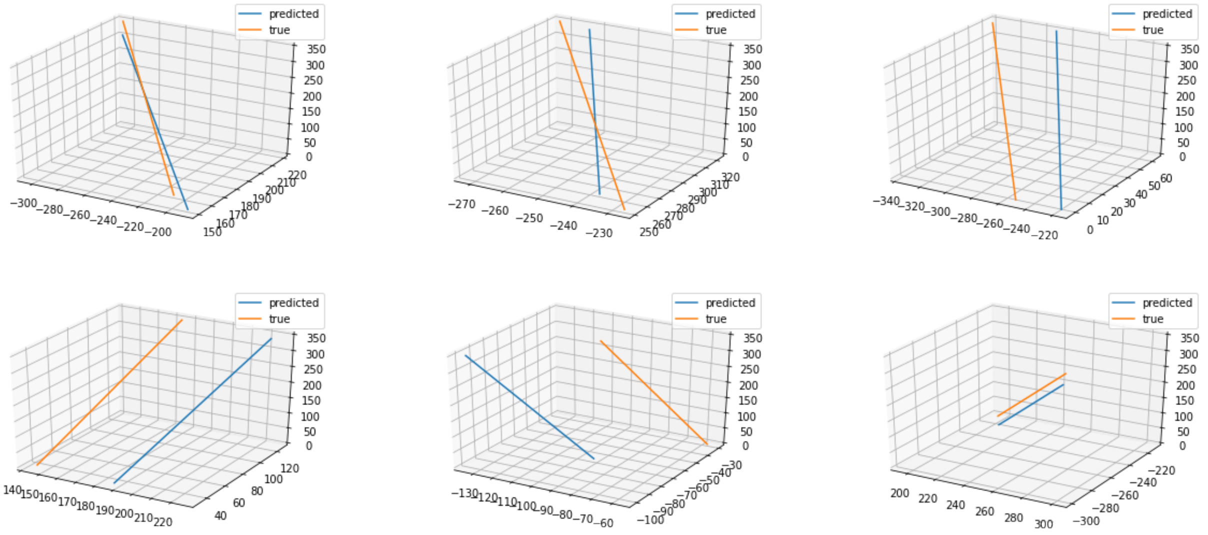


Figure 16: Some true and predicted trajectories (from the validation dataset) of incoming particles in the calorimeter space

4 Challenges and Future Prospects

Using neural networks for regressive analyses in physics is quite interesting as well as challenging. A very basic regression model in this project produced some physical results but the real task lies in going further on from there – using clever techniques and the knowledge of physics to modify the inputs and to design a model that is efficient, reproducible and as general as possible. In this project, the author, being a complete beginner in the world of machine learning, has managed to scratch the surface of what is a vast field of research. As such, there remains a lot of ideas that are yet to be explored.

One such idea was to use the maximum and total energies of an event as an input parameter to the model. Given the dimension of the `calorimeter_data` variable, an attempt was made to add it separately at the end, after the pooling layer gave a 1D output. However, this did not work as expected – the output values tended to peak very close to 0 (figure 17).

Testing with a high number of epochs in the model's training was also a challenge as this is

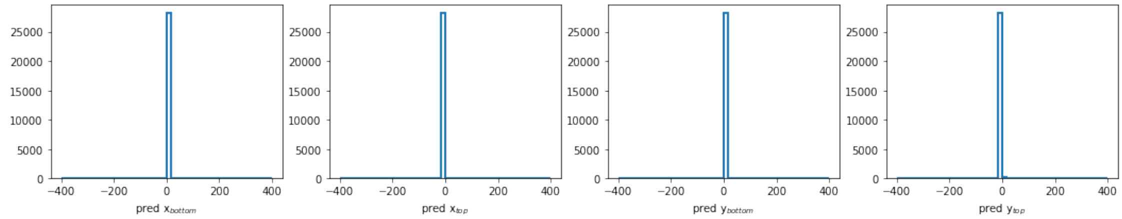


Figure 17: Distributions of the predicted x- and y-coordinates (in units of mm) in the top and bottom planes of the calorimeter with a model that attempt to include the maximum and total energies

quite time-consuming without GPUs. After several attempts, using the GPU nodes on the university's HPC cluster (Baobab) turned out to be not as trivial as anticipated and the only other option remaining was Google Colab, which, though quick is unreliable. The code used is still only at a testing stage (Jupyter notebooks) and a well-structured python script for easy reproducibility and model evaluation is missing as of now and can be worked on in the near future. Besides, severe time constraints led to a shortage in discussions with assistants and classmates, which, in turn, resulted in a limited progress in the work. The author realises that not all the targets of this project were met but, nevertheless, enjoyed working on it and is happy to present whatever little has been.

References

[1] J. Chang *et al.*, “The DArk Matter Particle Explorer mission,” *Astroparticle Physics*, vol. 95, pp. 6–24, oct 2017. [Online]. Available: https://arxiv.org/abs/1706.08453

[2] Y. Wei *et al.*, “On-orbit performance of the DAMPE BGO calorimeter,” *Proceedings of 37th International Cosmic Ray Conference, PoS(ICRC2021)*, vol. 395, p. 081, 2021.

[3] ——, “Performance of the DAMPE BGO calorimeter on the ion beam test,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 922, pp. 177–184, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168900218318333

[4] “Towards Data Science,” https://towardsdatascience.com/7-tips-to-choose-the-best-optimizer-47bb9c1219e, [Online; accessed 26-May-2022].

[5] “Keras Callbacks API,” https://keras.io/api/callbacks/early_stopping/, [Online; accessed 26-May-2022].

[6] “Diagrams.net,” https://app.diagrams.net/, [Online; accessed 11-June-2022].