

Neural Networks for Electron Identification with DAMPE

David F. Droz*, **Andrii Tykhonov**, **Xin Wu**, for the **DAMPE collaboration**[†]

University of Geneva

E-mail: david.droz@unige.ch

The Dark Matter Particle Explorer (DAMPE) is a space-borne particle detector and cosmic ray observatory in operation since 2015, equipped with alongside other instruments a deep calorimeter able to detect electrons up to an energy of 10 TeV and cosmic hadrons up to 100 TeV. The large proton and ion background in orbit requires a powerful electron identification method. In recent years, the field of machine learning has provided such tools. We explore here a neural network based approach to an on-orbit particle identification problem. We present the issues that arise from the constraints of particle physics, notably the difference between training set based on simulated data, and the application set based on real unlabeled data, leading to a trade-off between performances and general usability. We finally compare the neural network discrimination power with the more traditional cut-based analysis.

36th International Cosmic Ray Conference

July 2019

Madison, WI, USA

*Speaker.

[†]for collaboration list see PoS(ICRC2019)1177

1. The Dark Matter Particle Explorer - DAMPE

The DARK Matter Particle Explorer (DAMPE) is a cosmic ray detector and gamma ray telescope in operation since December 2015 on a sun-synchronous orbit around the Earth[1][2]. It is constituted of four subdetectors (fig. 1): a plastic scintillator (PSD) for absolute charge measurement; a silicon tracker/converter (STK) for precise direction measurement and for enabling photon pair production; a bismuth germanate imaging calorimeter (BGO) of about 32 radiation lengths, made of 308 hodoscopically arranged bars in 14 layers and used for particle identification, energy measurement, and trigger; and a neutron detector (NUD) for improving the identification of hadronic showers.

Among the main scientific objectives of DAMPE is the study of cosmic electrons, leading notably to the direct detection of a spectral break in the TeV region as the first scientific result of the mission[3].

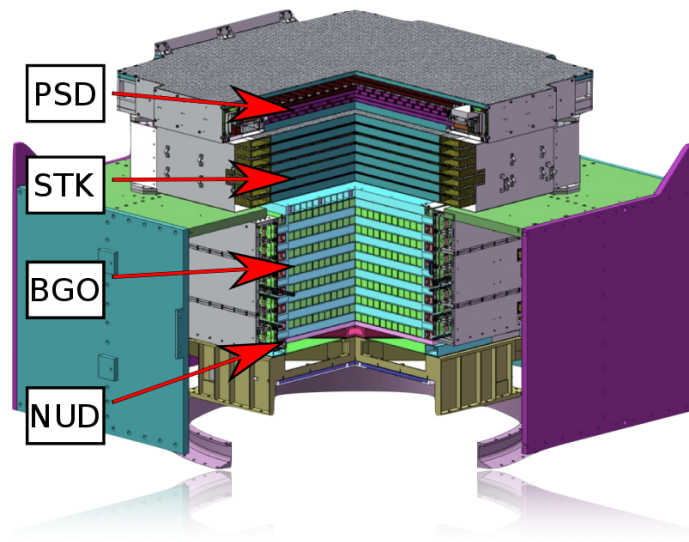


Figure 1: Layout of the DAMPE detector system

2. Electron identification

Measuring electrons in orbit requires a high rejection power for the main background source, protons. Protons have the same absolute electric charge as electrons and their flux is orders of magnitude stronger. The difference lies in the physical processes when interacting with matter: protons produce a wide and deep hadronic shower, electrons produce narrower electromagnetic showers. Classically, we can build high-level observables that quantify the shower shape inside a calorimeter, and use it for particle discrimination[4]. In the DAMPE experiment, said observable is a variable named ζ that combines the depth and spread of the shower inside the BGO calorimeter. The DAMPE collaboration has already detected a break in the cosmic electrons spectrum with an analysis that exploits the e/p discrimination capabilities of ζ [3]. While powerful, such a hand-crafted variable does not use the full information available in the detector, including the possibility

to exploit the strong correlations between topological variables used to describe the shower development in the imaging calorimeter, and shows its limits above the TeV range where the topological development of hadronic and electromagnetic showers in the detector is less pronounced. More powerful methods are therefore required.

3. Neural networks for electron identification

Over the last decade, the fields of artificial intelligence and data sciences have been experiencing an incredibly fast development, in no small part thanks to the advent of deep learning[5]. Deep neural networks are now the state-of-the-art in a wide range of data-driven applications, from computer vision to speech recognition, search engines or self-driving cars. They have been suggested for particle physics as well[6].

We propose to use deep learning for electron identification. In machine learning terms, this is a binary classification task: discriminate between a signal class (electrons) and a background class (protons). We explore two techniques: a multivariate analysis using a multilayer perceptron (section 3.1) and a pattern recognition using convolutional neural networks (section 3.2).

3.1 Multilayer perceptron (MLP)

The first approach is a regular feed-forward neural network, sometimes named *multilayer perceptrons* (MLP). The architecture is a stack of fully connected layers of neurons. A neuron is a mathematical unit that applies a non-linear function to the combination of its inputs and sends the output value to the next layer. Mathematically, if a neuron receives as input a set $\{X_i\}$, then its output y is:

$$\{X_i\} \longrightarrow y = f\left(\sum_i w_i X_i\right) + b$$

where f is the non-linear activation function, w_i the weights and b the bias. Both w_i and b are determined during the training procedure: the network is exposed to a set of labelled data (*training data*), and uses gradient descent to minimise an error metric. In the case of classification, the canonical choice for the error metric is the cross-entropy[7].

Such a network takes as input a one-dimensional set of variables $\{X_i\}$ and outputs a value in the $[0; 1]$ range that can be interpreted as the probability for an electron to produce $\{X_i\}$.

Both the choice of our input variables and the determination of our MLP hyperparameters are the result of an extensive optimisation and gridsearch[8] campaign using Nvidia GPUs. We settled with a 4-layers network (fig. 2) using rectified linear units (*ReLU*)[7] activation and regularised with dropout[9], with a total of 66k trainable parameters. Our model was trained over 100 iterations (*epochs*) with the Adam optimiser[10]. The input variables are selected to be as descriptive as possible of a shower interaction inside the calorimeter: the energy deposited in every layer of the calorimeter, the spread of said deposition, the total reconstructed energy, and the inclination of the reconstructed particle trajectory. We also add the ζ variable, used for the classical electron identification. The motivation is to give the network our current best estimator along with the low level quantities required to improve it.

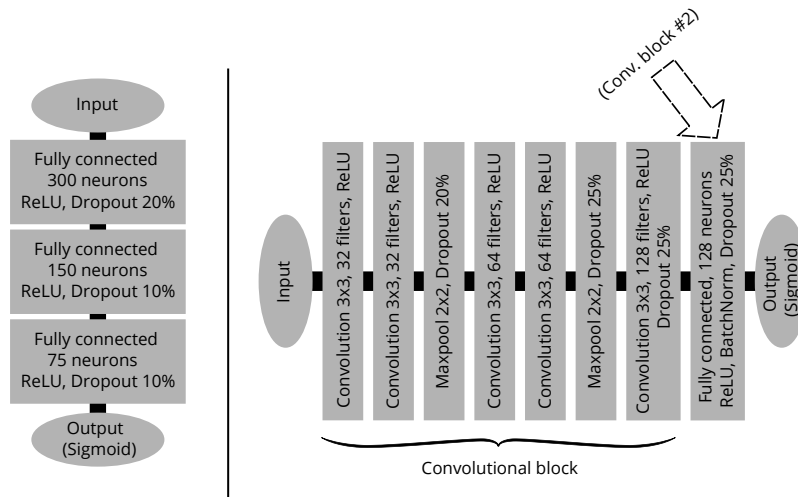


Figure 2: Schematics of the different neural networks, left the MLP and right the CNN. The second convolutional block is added for the case of separate XZ/YZ views of an event. Details of each technique are presented in [5][7]

3.2 Convolutional neural networks (CNN)

The second approach is using pattern recognition techniques. The interaction of a particle inside a calorimeter can be represented as an image, where the pixel values are the energy deposited in every section of the detector (fig. 3). The BGO calorimeter is composed of 14 layers of 22 bars each, which would hence result in a picture of 14x22 pixels. A complication however arises in this approach: as described in section 1, the bars have an alternating orientation: there are 7 layers with bars in the "X" direction, and the remaining 7 in the perpendicular "Y" direction. To build an image out of an event, we can then either make two separated 7x22 pictures, or ignore this alternating orientation and make a combined 14x22 picture. We choose to explore both options.

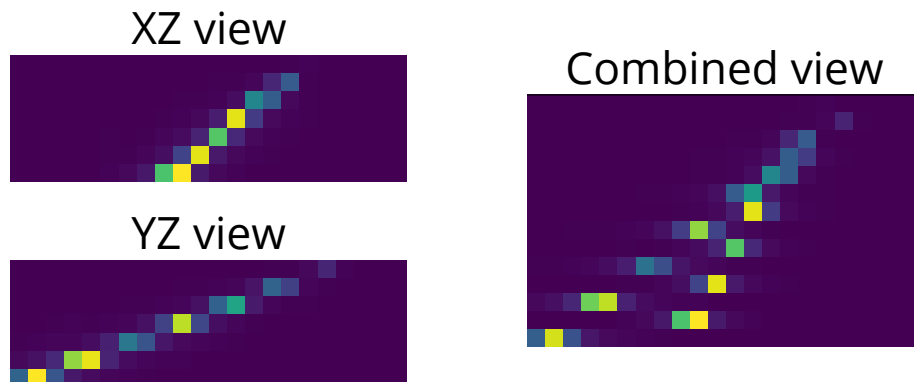


Figure 3: Image representation of a simulated 2 TeV proton inside the BGO calorimeter.

The current state-of-the-art in pattern recognition is achieved through *convolutional neural networks* (CNN)[5]. The idea is to have convolutional filters scanning through an image and learning to extract relevant patterns and features, interleaved with pooling layers that gradually down-sample the image while providing translation invariance. The extracted features are then fed into

a fully connected neural network for the final classification. Instead of going with a very computationally expensive optimisation campaign, we pick an off-the-shelf model[11] that exhibits good performances on the low-dimensionality MNIST task[12]. The architecture (fig. 2) consists of 9 layers regularised with dropout and batch normalisation[13], with a total of 730k trainable parameters. It is trained with the Adam optimiser over 50 epochs. For the case of two 7x22 pictures, each image is fed into parallel convolutional blocks which are then merged into the dense layers.

3.3 Results

All models were trained on protons and electrons Monte Carlo (MC) events simulated with the Geant4 package. The training data was prepared with a set of cleaning cuts to replicate the analysis chain applied on real data, with in particular cuts to remove Helium nuclei[3]. We kept a balanced set of roughly 1 million events per class, split 60/40 between training and testing.

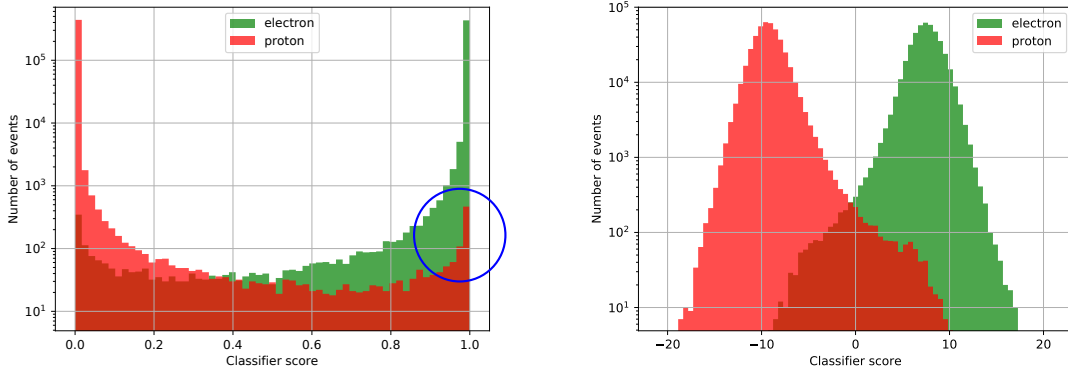


Figure 4: Log-scale histogram of the CNN output on Monte Carlo. **Left:** With a vanilla model, values are crunched in the $[0; 1]$ range, resulting in a background peak at 1.0 exactly as shown by the blue circle. **Right:** By removing post-training the last activation function, we get a more monotonic behaviour for both classes.

A first feature we noticed is that the neural network output values are either very close (or equal to) 0.0 or 1.0, with only few events classified in-between. This holds true for false positives and false negatives as well: figure 4 (left) shows that the histogram of MC protons exhibits two peaks: one at 0.0 (true negatives) and the second, much smaller, at 1.0 (false positives)¹. The overall distribution is therefore non-monotonic, which can be problematic: in further stages of the analysis, we would like to estimate the performances and uncertainties directly from the flight data, for example using a baseline background extrapolation. If said background has a non-monotonic distribution, extrapolation methods become tricky if not outright impossible.

The cause is a feature of neural networks for classification: the very last operation is a logistic sigmoid function that maps the output to the $[0; 1]$ range:

$$f(x) = \frac{1}{1 + e^{-x}}$$

¹These events are likely protons that transfer most of their energy to one or several π_0 , starting E.M. showers while the remaining energy starts only negligible hadronic showers. They are therefore the most difficult background to distinguish from electron-induced showers.

Values $x \gg 0$ are mapped to $f(x) \simeq 1.0$, effectively compressing the output into a limited, finite space. Computer floating point accuracy also has an influence: for a 16-bits float, $f(18) = f(20) = 1.0$ exactly. Therefore the function is not bijective anymore.

A possible workaround is to remove the logistic sigmoid from the output layer. This must be done after training since the metrics being optimised (binary crossentropy) assume an output between 0 and 1. The result is shown on figure 4 (right): the compression and non-monotonic features are gone, resulting in a distribution much easier to use for baseline extrapolation methods, without altering the classification performances. Another important gain is getting rid of the seemingly irreducible background: without the sigmoid we can select a small sample with 0% background, by e.g. cutting away events with a score $\lesssim 12$. Whereas with the sigmoid, we still have protons passing even a cut at 1.0.

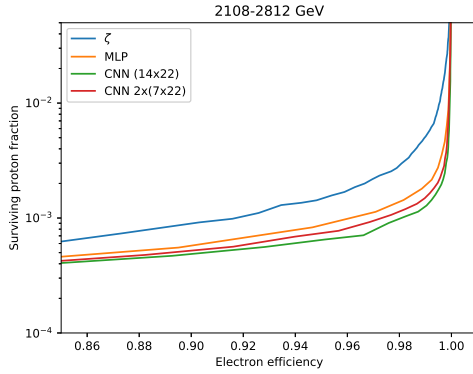


Figure 5: Sample ROC curve on Monte-Carlo for the MLP, CNN with separated images, CNN with combined images, and the classical ζ method, at an energy around 2 TeV. The lowest curves have the lowest background for a fixed efficiency.

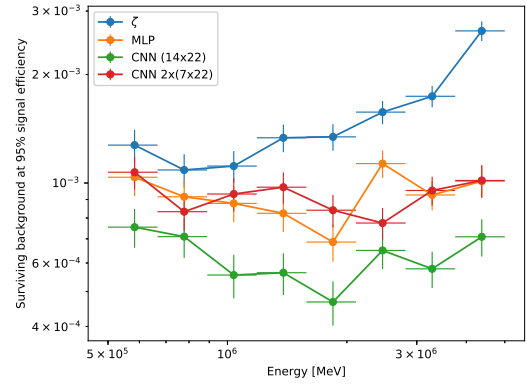


Figure 6: Energy dependency of the surviving background fraction for a fixed signal efficiency of 95%, for all neural networks and the classical ζ method.

In order to quantify the classifier performances, we report on figure 5 a sample Receiver Operating Characteristics (ROC) curve for all our neural networks, in comparison with the classical ζ method, in an energy bin around 2 TeV. ROC curves are obtained by computing classification metrics at various discrimination thresholds, on the test sample. In our case we settle to plot the signal efficiency against the remaining background. On the figure, classifiers with the lowest curves have the smallest background for a set efficiency. We first see that all neural networks outperform the classical method. To better quantify said improvement, we report on figure 6 the background when we set the discrimination threshold such as to have a 95% signal efficiency, as a function of the energy reconstructed from the BGO calorimeter. This second plot shows first that the performances gained by neural networks are more significant at higher energies, where ζ reaches its limits.

Curiously, we see that the combined image approach described in section 3.2 leads to a better discrimination than the separated images, despite ignoring completely the X/Y orientation of the BGO bars. This combination seems in fact necessary to fully exploit the power of pattern recognition, as otherwise the CNN isn't any more powerful than a much simpler MLP.

While figure 6 designs CNNs as the best method, an extra check is necessary: the compat-

ibility between MC data and real, flight data. Figure 7 shows the output distribution for both methods, comparing flight and MC data on a representative energy range. Both samples have been cleaned such that contributions from other cosmic species, notably Helium nuclei, is negligible[3]. The matching between flight data and MC for the MLP classifier is better than that for the CNN classifier. In the latter, the real electron peak is closer to protons than the MC one, resulting in effectively lower performances than previously estimated. The apparent discrepancy in this case can be accounted for by uncertainties related to the electronic response simulation of the BGO photomultipliers, at very low energies of about few MeV. Due to the inner mechanisms of neural networks and their high sensitivity to training data, in particular at low energies, even tiny differences can get amplified up to high levels

This apparent decrease in CNN performances would need to be quantified, and raises the need for a MC correction which would be a source of systematic uncertainties. MLP thus stands out as an attractive compromise between raw classification performances and usability on real data.

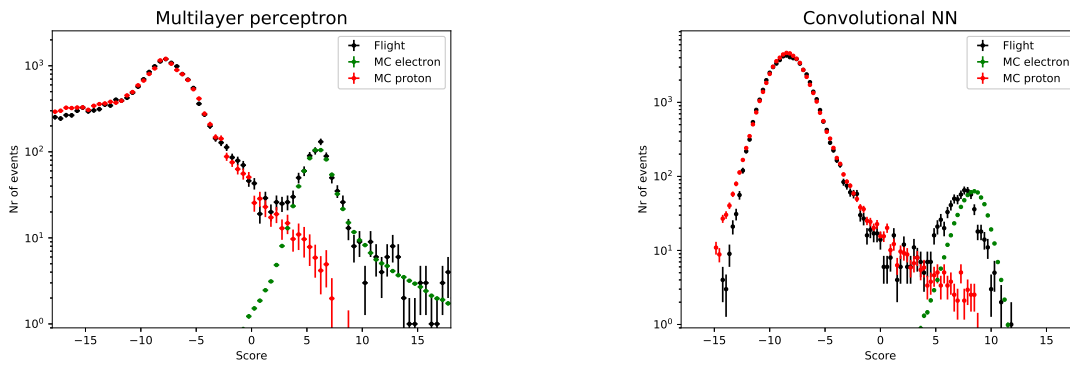


Figure 7: Comparison between real (flight) and Monte-Carlo (MC) data for the multilayer perceptron (left) and convolutional neural network (right), in the 1-2 TeV energy range.

4. Conclusion

The field of machine learning is ongoing a booming development thanks in no small part to deep learning. While deep neural networks are only sparsely used in cosmic rays physics, we demonstrate here that they hold the potential of reaching even more precise measurements as they significantly outperform more classical methods, a welcome gain in an era of increasingly smaller signals in particle physics. Yet we also demonstrate that they aren't some sort of plug-and-play miracle box and that instead some complications arise when moving away from naive benchmarking towards practical application: their sensitivity to training data can throw them off when switching to the real dataset. Nevertheless, the neural networks presented here will allow DAMPE to refine its cosmic electrons measurements by improving the background rejection at the highest energies.

Acknowledgements

The DAMPE mission is funded by the strategic priority science and technology projects in space science of Chinese Academy of Sciences. In China the data analysis is supported in part

by the National Key Research and Development Program of China, the National Natural Science Foundation of China, the strategic priority science and technology projects of Chinese Academy of Sciences, the 100 Talents Program of Chinese Academy of Sciences, and the Young Elite Scientists Sponsorship Program. In Europe the activities and the data analysis are supported by the Swiss National Science Foundation (SNSF), Switzerland; the National Institute for Nuclear Physics (INFN), Italy.

The computations presented in this document were performed at University of Geneva on the Baobab cluster. We used the Keras framework[14] with the Theano deep learning library[15].

References

- [1] J. Chang *et al.*, “The DArk Matter Particle Explorer mission,” *Astropart. Phys.*, vol. 95, pp. 6–24, 2017.
- [2] G. Ambrosi, Q. An, R. Asfandiyarov, P. Azzarello, P. Bernardini, M. Cai, M. Caragiulo, J. Chang, D. Chen, H. Chen, *et al.*, “The on-orbit calibration of dark matter particle explorer,” *Astroparticle Physics*, vol. 106, pp. 18–34, 2019.
- [3] G. Ambrosi *et al.*, “Direct detection of a break in the teraelectronvolt cosmic-ray spectrum of electrons and positrons,” *Nature*, vol. 552, pp. 63–66, 2017.
- [4] J. Chang *et al.*, “Resolving electrons from protons in ATIC,” *Adv. Space Res.*, vol. 42, pp. 431–436, 2008.
- [5] Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [6] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for Exotic Particles in High-Energy Physics with Deep Learning,” *Nature Commun.*, vol. 5, p. 4308, 2014.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
<http://www.deeplearningbook.org>.
- [8] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.
- [9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [11] A. Soni, “Mnist with keras.”
<https://www.kaggle.com/adityaecdr/mnist-with-keras-for-beginners-99457>, 2018.
- [12] Y. LeCun, “The mnist database of handwritten digits,” <http://yann.lecun.com/exdb/mnist/>, 1998.
- [13] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [14] F. Chollet *et al.*, “Keras.” <https://keras.io>, 2015.
- [15] Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions,” *arXiv e-prints*, vol. abs/1605.02688, May 2016.