Artificial Intelligence Research and Development
T. Alsinet et al. (Eds.)

© 2024 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA240417

Machine Learning for Particle Identification in LHCb

Sergi BERNET ANDRÉS [a,1], Míriam CALVO GÓMEZ [a,2]
Álvaro GARCÍA PIQUER [a,3] and Xavier VILASÍS CARDONA [a,4]
[a] *La Salle - Universitat Ramon Llull*
ORCiD ID: Sergi Bernet Andrés https://orcid.org/0000-0002-4515-7541, Míriam Calvo Gómez https://orcid.org/0000-0001-5588-1448, Álvaro García Piquer https://orcid.org/0000-0002-6872-4262, Xavier Vilasís Cardona https://orcid.org/0000-0002-1915-9543

Abstract. LHCb is one of the four largest high-energy physics experiments at CERN focused in high precision measurements of particle physics. The LHCb detector has undergone a recent upgrade [1] implying changes at subdetectors, data taking conditions and data processing model. Information from subdetectors is processed at 30MHz at a first trigger phase builded entirely with GPUs to reduce this rate down to 1MHz. Afterwards, the same information is processed in a second trigger phase that runs in CPUs, performing a complete reconstruction and identification of particles. This upgrade implies an evolution of the algorithms used at trigger level. In order to keep performance and speed up processing time, some of them have been replaced by machine learning algorithms. To perform particle identification, one of the LHCb approaches uses a neural network using the information from all subdetectors. In this paper we explain the advantages of this method and the capabilities that machine learning brings to LHCb focused in the global particle identification and throughput improvement achieved with it.

Keywords. LHCb, particle identification, machine learning, neural networks

1. Introduction

LHCb at CERN is one of the four main experiments, being the only focused in high precision measurements. To achieve this task, the data is processed in real time at first level trigger called HLT1 where a partial reconstruction is conducted to select potential information. Then, the second level trigger named HLT2 conducts the precise reconstruction using filtered information. This information is stored in files called events that contains tracks information (particles path) used to reconstruct particles. To do this task, LHCb relies in particle identification (PID) algorithms that combine information from subdetectors specialized in identification such as RICH, calorimeter and muon systems.

[1] sergi.bernet@salle.url.edu
[2] miriam.calvo@salle.url.edu
[3] alvaro.garcia@salle.url.edu
[4] xavier.vilasis@salle.url.edu

Due to recent update, algorithms focused on this task need an update to maintain PID performance. At the same time, data taking conditions have also evolved with a larger number of collisions forcing an adaptation of some algorithms. To conduct this task, LHCb utilizes 2 approaches to identify charged particles, the traditional one that relies in likelihood comparison and a neural network approach called probNN.

Taking each track information from the events, probNN combines this information and performs a classification into one of the 6 particles considered: electron, pion, kaon, muon, proton and ghost, being this last one the result of combinatorial effect that looks like a real particle.

This probNNs perform a one-vs-all approach for each particle, implemented in a machine learning inference engine developed for LHCb that uses PyTorch training together with ONNX inference. This inference engine is focused on mimicking the existing machine learning algorithms in special multilayer perceptrons (MLP). On this specific problem, the outcome will be the probability of the given track characteristics correspond to a particle type.

2. Current solutions

LHCb has two procedures for charged particle identification, a traditional one based on a likelihood [2] and probNN implemented using TMVA [3] package. This solution uses almost 50 variables as input with 2 hidden layers that contain between 55 and 70 neurons each one and a single neuron output for binary classification. All together gives place to specialized networks with architectures over 7.5k parameters.

3. Data

Data used comes from LHCb simulation framework [4]. To ensure the data is as realistic as possible, latest conditions to emulate real data taking for this year have been used, getting between 50k and 100k samples for each particle type looking for a balance to improve networks convergence. Then, a first sample process to homogenize distribution coverage and avoid biases is applied to ensure networks generalize during training. All variables are described by quantitative values such as the energy stored in the calorimeter system associated to a track, but some others are described by discrete or binary values.

3.1. Variable Selection

In order to properly select variables, a first filter has been applied to some of the variables to avoid biases with the sample used. This is the case for the ones related to energy or geometry.

After this reduction, an analysis mixing different methods is performed. First, mutual information is used to determine which variables describe better each particle type. Kolmogorov-Smirnov test is applied to determine if variable distributions between the particle type and the rest are separable. Finally, a feature extraction from fundamental machine learning algorithms such as decision trees and random forest has also been used.

Each method provided different variables to be used, but after carefully cross-validate them, none of the individual methods has proven a significant improvement

amongst the rest. Finally, a ranking method that combines all the different selections has been implemented. This method is prepared to reward those variables that have been stable on top of each method, bringing a slight improvement in PID performance around 1.5% against individual selection methods.

4. Solution

TMVA has been the default machine learning framework to inference models in LHCb, requiring the training be performed independently and introduced afterwards. The new inference engine avoids this procedure allowing both train and inference in LHCb framework and reduces 10% the time inference with same sized networks.

We take advantage from the update and perform a proper exploratory analysis, reduce the needed variables to the minimal expression and a genetic approach to reduce architectures while maintaining the PID performance.

To do so, a genetic algorithm is implemented to find a reasonable architecture. Taking care of population's genotype that describes the architecture of the networks we can control the maximum size of the networks. This is done for networks with 1, 2 and 3 hidden layers and then, based on the metric defined (AUC in this case) we select the architecture providing best best performance with least parameters.

5. Results

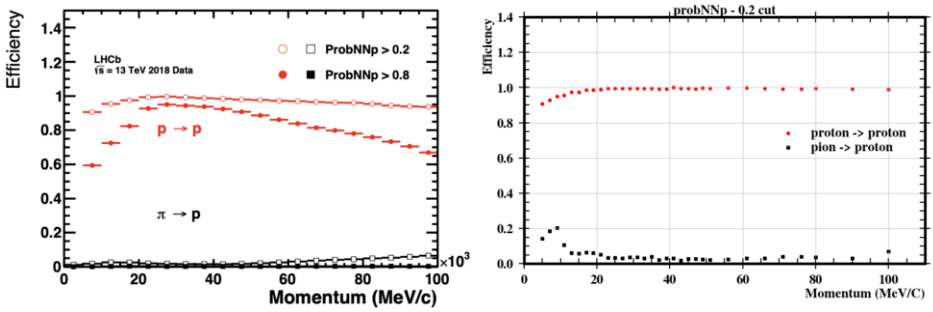
Results are consistent for all particle types as seen in table 1. PID has been maintained compared with original probNN while architecture sizes have been drastically reduced. The least reduced architectures have been for electrons, muons and kaons with a factor 4. At the same time, the greatest reduction is achieved for kaons with a factor 14. Finally, for protons and ghosts the reduction factor has been a factor 10, all disposed in table 2. This reduction is derived from the huge reduction in input variables that allowed the exploration of several architecture combinations upon 3 layers, improving LHCb throughput almost a 2%. A final comparison between proton efficiencies can be seen in figure 1.

probNN	Electron	Pion	Muon	Kaon	Proton	Ghost
AUC	0.996	0.994	0.998	0.993	0.994	0.945

Table 1. ProbNNs AUC results on validation data

probNN	Electron	Pion	Muon	Kaon	Proton	Ghost
Original inputs	48	48	47	49	49	48
Original layers	57, 57	76	56, 65	78, 78	58, 68	76, 57
Original size (params)	6000	3700	3800	10000	6800	8000
Optimized inputs	16	15	14	14	14	15
Optimized layers	32, 21, 7	34, 20, 6	32, 13, 3	21, 16, 5	25, 10	26, 10
Optimized size (params)	1300	1300	900	700	600	700

Table 2. ProbNNs architectures and size comparison



(a) probNNp previous efficiency

(b) probNNp: current efficiency

Figure 1. probNNp efficiency comparison

6. Outcome

This study has demonstrated that proper data exploration to perform dimensionality reduction together with an architectural optimization by using genetic algorithms is the key to improve real time applications using neural networks in the LHCb experiment.

References

[1] Aaij R, et al. The LHCb upgrade I. 2023 5.
[2] LHCb detector performance. International Journal of Modern Physics A. 2015 Mar;30(07):1530022. Available from: http://dx.doi.org/10.1142/S0217751X15300227.
[3] Therhaag J. TMVA Toolkit for multivariate data analysis in ROOT. PoS. 2010;ICHEP2010:510.
[4] Clemencic M, Corti G, Easo S, Jones CR, Miglioranzi S, Pappagallo M, et al. The LHCb simulation application, Gauss: Design, evolution and experience. J Phys Conf Ser. 2011;331:032023.