Machine Learning in KM3NeT

Chiara De Sio[1] for the KM3NeT Collaboration

[1]Dipartimento di Fisica Università di Salerno and INFN Gruppo Collegato di Salerno, via Giovanni Paolo II 132, Fisciano 84084 Italy

Abstract. The KM3NeT Collaboration is building a network of underwater Cherenkov telescopes at two sites in the Mediterranean Sea, with the main goals of investigating astrophysical sources of high-energy neutrinos (ARCA) and of determining the neutrino mass hierarchy (ORCA). Various Machine Learning techniques, such as Random Forests, BDTs, Shallow and Deep Networks are being used for diverse tasks, such as event-type and particle identification, energy/direction estimation, source identification, signal/background discrimination and data analysis, with sound results as well as promising research paths. The main focus of this work is the application of Convolutional Neural Network models to the tasks of neutrino interaction classification, as well as the estimation of energy and direction of the propagating particles. The performances are also compared to those of the standard reconstruction algorithms used in the Collaboration.

1 Introduction

KM3NeT is a multi-km^3 network of neutrino telescopes [1] designed with the main goals of identifying sources of cosmic neutrinos [2] and establishing the neutrino mass hierarchy [3]. The detector is composed of a three-dimensional array of photomultipliers (PMT), hosted in spherical structures called Digital Optical Modules (DOM). In this setting, neutrinos are indirectly detected by collecting the Cherenkov light emitted by the propagation in the sea water of the particles produced in the interaction. KM3NeT events are then represented as a collection of signal *hits* on the photo sensors.

Convolutional Neural Networks (CNN) [4] are applied to the problem of identifying and studying neutrino interactions in order to classify KM3NeT events according to their topological features (i.e. space-time distribution of photomultiplier signals). Similarly, Deep Learning-based solutions have also been applied to estimate the interaction parameters of neutrino events for which complex reconstruction algorithms are usually needed.
The following tasks for the KM3NeT-ARCA project are presented:

- up-going/down-going neutrino event classification
- neutrino interaction (ν_μCC/ν_eCC) classification
- neutrino energy estimation
- neutrino direction estimation

Specific models have been designed according to the requirements of each task, by leveraging the properties of ad-hoc Neural Network architectures. A Deep Learning track/shower classifier for KM3NeT-ORCA is also described.

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

Results obtained from the Deep Learning models are then compared to the KM3NeT official track reconstruction algorithm for ν_μCC events, and/or other Machine Learning-based solutions.

2 Deep Learning applications for KM3NeT-ARCA

The data used as input for the Deep Learning models are KM3NeT triggered ν_μCC and ν_eCC events. The trigger conditions have been optimised to ensure the purity of event selection and the quality of event reconstruction. Events with at least 5 DOMs hit in a 20 ns coincidence time window have been selected as input for the deep learning analyses here reported. A KM3NeT event is represented as a collection of *hits* and times: (DOM-ID, t_i), where DOM-ID refers to the unique identification number of the Digital Optical Module (DOM) that detects the signal, whilst t_i is the corresponding time instant, with nanosecond granularity, of the i-th hit, within the 100 ms time-slice defined for the data acquisition.

A space regularisation is needed to prepare the data to be analysed with CNNs: a lattice, exactly 90 m and 36 m spaced in the (X, Y) and Z plane, is defined, which contains the whole ARCA detector, without altering the hit distribution. The event data are discretised in time and space with a fixed number of steps. The collection of hits over time is mapped to a 4D structure, according to the lattice dimensions.

A dataset of 258,879 ν_μCC and ν_eCC events is split into a training and a test dataset. The training set is then further split in a training and a validation subset, the latter being used for performance assessment during the training phase. All the code is written in Python. The Keras framework, a high-level Neural Network API [5] has been used, on the Tensorflow [6] backend. The training and test of the Neural Network models have been performed on a NVIDIA GEFORCE GTX-1080Ti GPU.

2.1 Up-going/Down-going neutrino classification

With the aim of classifying *up-going* and *down-going* neutrino events, according to the evolution of the Z coordinate of the propagating particle over time, a CNN model is defined to analyse the projection in the (Z,t) plane of events. The model architecture is a stack of three 2D Convolutional blocks (each composed by two 2D Convolutional layers followed by an Average Pooling layer [7]), and three fully-connected layers. The last fully-connected layer is equipped with a Softmax activation function [8] to perform the classification. Input tensors are reshaped by summing over the X and Y coordinates, and events are labelled according to the zenith angle of the neutrinos, as *up-going* or *down-going*.

On test events, a classification accuracy of 93.3% is reached. The percentage of misclassified events for the two classes is comparable (up-going events: 6.4%; down-going events: 7.1%). The higher percentage of misclassified down-going events is compatible with the lower detection efficiency for the events propagating from above, due to the DOM design, comprising a lower number of photomultipliers in the top hemisphere. The best performances are obtained for high-energy events (characterised by a larger number of hits) and for events developing at short distances (below 400 m) from the centre of the detector (Fig. 1a).

2.2 Neutrino interaction classification

To distinguish between ν_μCC and ν_eCC interactions, characterised by a different spatial evolution, two different views of the neutrino events are considered. Two branches in the network, analysing separately the (Z, t) and (X, Y) evolution, are defined (each composed by a stack of three 2D Convolutional blocks), concatenated and merged in the last three learning steps to extract common features. Events are labelled to indicate whether they have been generated by the charged current interaction of a muon or of an electron neutrino, which produce mostly “track-like” or “shower-like” events respectively. A Softmax activation function is used in the last fully-connected layer.

The test of the Neural Network model on unseen events shows an accuracy of 92.8%, with a higher classification error in the muon neutrino case (10.3%, compared to the 2.7% for electron neutrinos), reflecting the possibility of ν_μCC interactions to produce either a track or a particle shower. The classification performance shows the same dependence on the energy and on the distance from the detector centre as the previous case (Fig. 1b).

2.3 Neutrino energy estimation

A regression model is defined to produce an estimation of the energy of the neutrinos from the analysis of the signal produced by the propagating particle(s). A branched Neural Network model is used to take into account the evolution over time of the (Z, t) and (X, Y) view of the events. The architecture is composed of two branches of stacked 2D Convolutional blocks, concatenated and followed by four fully-connected layers. The energy estimation is performed by means of a fully-connected layer with a linear activation function, which calculates the estimated energy. The Monte Carlo simulated energy is used as reference truth for this learning task. A Mean Squared Error (MSE) [9] of 0.22 is achieved on test data, considering a dataset of both ν_μCC and ν_eCC events (Fig. 1c). An r^2 score (coefficient of determination) [9] of 0.84 is obtained.

2.4 Neutrino direction estimation

A partial neutrino direction estimation is performed by estimating the Z component of the neutrino direction, using a regression model to analyse the (Z,t) view of the neutrino events. The Convolutional Neural Network model used for this task is composed of a stack of three 2D Convolutional blocks, followed by three fully-connected layers. The last layer, with a cosine activation function, is used to estimate the cosine of the zenith angle of the propagating particle. The model, trained on both ν_μCC and ν_eCC events, results in an MSE of 0.03 (Fig. 1d). As a further investigation, a test on the two separate classes is also performed. As expected, the best estimations are obtained on ν_μCC events, since tracks have a more defined direction, compared to particle showers, mostly produced by electron neutrino events.

2.5 Comparison with the official reconstruction algorithm

The performances of the previously presented estimations are compared, when applicable, to the results obtained with the KM3NeT standard track reconstruction algorithm. Thus, only ν_μCC events are considered.

On the neutrino direction estimation, the Neural Network models, trained on a dataset comprising both track-like and shower-like events, produce an overall stable estimate in the whole energy range. On the other hand, the standard reconstruction produces, on average, a more accurate estimation for the Z component of the neutrino direction. The two algorithms give, as a result, an MSE of 0.002 and 0.001 respectively.

For the comparison on the classification of up-going/down-going events, the accuracies are of 0.998% and 0.987% for the standard reconstruction and CNN reconstruction respectively.

3 Deep Learning applications for KM3NeT-ORCA

With the aim of performing multi-purpose classification and regression studies on the ORCA configuration, multiple Convolutional Neural Network-based models have been designed. The tasks which are being considered are the energy and direction reconstruction, as well as the track-shower composition of event sets, and particle identification (PID).

A Deep Learning classifier called *OrcaNet* has been defined, by projecting the 6D information (XYZ, azimuth, zenith, t) of events to a 4D subspace (neglecting the PMT orientation). Space and times have been discretised to define the data structure suitable to be analysed with Convolutional Neural Networks and 4-weeks' training on a GPU Cluster has been performed. Two views (XYZ-t, YZt-X) of events have been analysed separately and merged just before the classification is performed.

The results obtained on track/shower classification outperform traditional Machine Learning models as RDF (Figs. 2a, 2b).

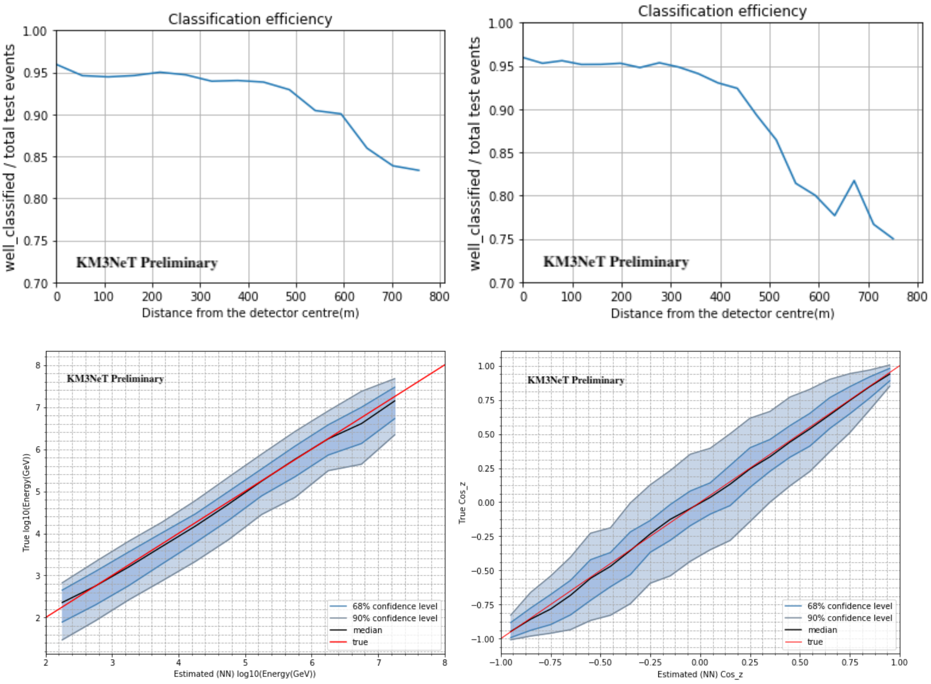

Fig. 1. (a,b,c,d from left to right): up-going/down-going classification efficiency (a) and νμCC/νeCC classification efficiency (b) as a function of the distance from the detector centre; regression performance of energy estimation with 68% and 90% intervals (c); regression performance of direction (Z component) estimation with 68% and 90% intervals (d).

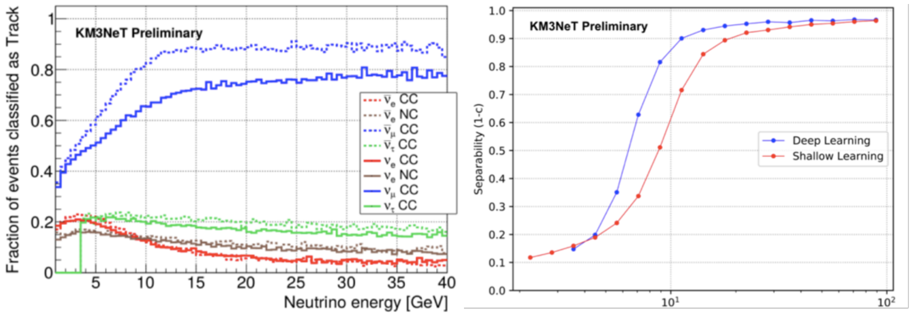

Fig. 2 Left*. fraction of events classified as tracks for different neutrino interactions; Right: separability of track and shower events, comparison between deep and shallow neural networks.*

4 Conclusions and outlook

The results of the described Deep Learning applications, and in particular the possibility to perform estimates by automatically extracting the needed features, are quite promising. Machine Learning models provide stable estimations, comparable and in some cases possibly better than other reconstruction algorithms. Furthermore, Machine Learning does not depend on any reconstruction, allowing for an independent event study to be performed directly from raw data.

There is room for improvement, starting from a more detailed detector description, a complete direction reconstruction and a complete neutrino flavour identification. The KM3NeT Collaboration is expected to produce a Deep Learning toolset for CNN applications in ASTERICS (H2020), possibly portable to other event-based experiments.

References

1. S. Adrián-Martínez et al. (KM3NeT Coll.), J. Phys. G. 43 1-131 (2016)
2. S. Aiello et al. (KM3NeT Coll.), https://arxiv.org/pdf/1810.08499.pdf (2018)
3. S. Adrián-Martínez et al. (KM3NeT Coll.), J. High Energy. Phys. 2017 8 (2017)
4. Y. LeCun et al., Convolutional networks and applications in vision, IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems, pp. 253-256, (2010)
5. F. Chollet et al., Keras, https://github.com/keras-team/keras (2015)
6. M. Abadi et al. https://arxiv.org/abs/1603.04467 (2015)
7. V. Dumoulin and F. Visin, A guide to convolution arithmetic for deep learning, ArXiv e-prints (2016)
8. C. Bishop, Pattern Recognition and Machine Learning. Information Science and Statistics, Springer (2006)
9. S. Raschka, Python Machine Learning. Packt Publishing (2015)