CCE5225 – Assignment 1

Dr Ing Gianluca Valentino

Department of Communications and Computer Engineering
Faculty of Information and Communication Technology
University of Malta

MiniBooNE particle identification signal/background classification

Please submit a report in the form of **one pdf document** as well as the code in the form of a **Jupyter notebook** by not later than **20th December 2020** on VLE.

The assignment is worth 30% of the unit's final mark.

Objectives

To train 3 simple classifiers: vanilla neural network, SVM and random forest on a dataset taken from the MiniBooNE particle physics experiment to determine whether a detected particle is important (i.e. signal) or not (i.e. background).

The MiniBooNE experiment at Fermilab (Illinois, USA) was set up to observe neutrino oscillations. A neutrino beam consisting primarily of muon neutrinos is directed at a detector filled with 800 tons of mineral oil and lined with 1,280 photomultiplier tubes. An excess of electron neutrino events in the detector would support the findings of a previous experiment which would have conflicted with the standard model expectation of only three neutrino flavours.

Machine learning can be used to classify whether a detected particle is an electron neutrino (signal) or a muon neutrino (background), based on particle identification (PID) variables, such as the particle energy and the radius of the tracks that they form in the detector. For more information, please have a look at an example of such previous work:

https://arxiv.org/abs/physics/0408124

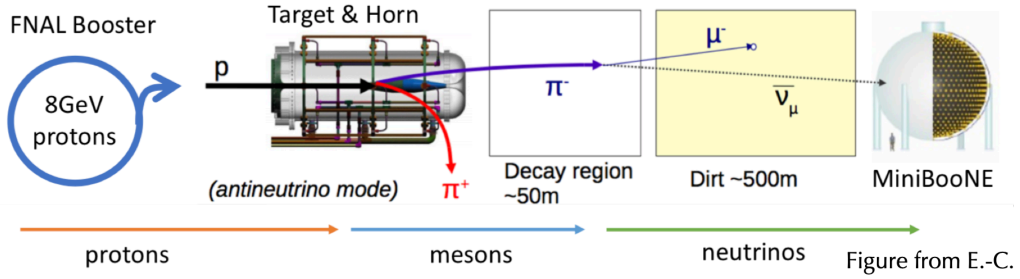

Figure from E.-C. Huang

Figure 1 – Generation of neutrinos from protons, and their path to the MiniBooNE detector.

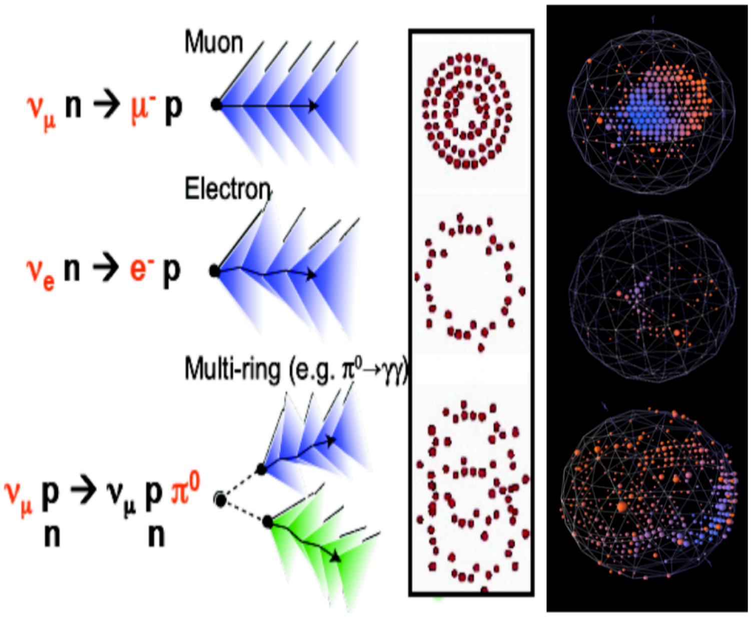
Muon
ν_μ n → μ^- p
Electron
ν_e n → e^- p
Multi-ring (e.g. $\pi^0 \to \gamma\gamma$)
ν_μ p → ν_μ p π^0
n n

Figure 2 – Examples of neutrino tracks generated in the MiniBooNE detector

Tasks

Download and prepare the dataset

- Go to https://archive.ics.uci.edu/ml/datasets/MiniBooNE+particle+identification and download the dataset.
- The dataset is set up as follows. The first line contains the number of signal events followed by the number of background events. The signal events come first, followed by the background events. Each line after the first line has 50 PID variables for one event.
- In order to train the ML models, you will need to develop some Python code to create a matrix of the input features, and a corresponding target vector (1s and 0s) for each row (event) in the dataset.

Training the 3 classifiers

For each of the above classifiers, follow the subsequent steps:

1. Scale the input features [10 marks]

2. Re-shuffle and divide the dataset into 80% for training, 20% for testing [10 marks]

3. Perform a grid-search over the hyperparameter space in order to obtain the hyperparameters (performing 5-fold cross-validation) that provide the best accuracy. Comment on whether the performance changes significantly amongst different hyperparameter values for each model. [30 marks]

4. Report the time required for training, the set of hyperparameters which were tested, and the final per-class accuracies achieved on the unseen test set (in the form of a confusion matrix). [30 marks]

5. Comment on the performance of each model, and provide an explanation as to why you believe the highest performing model gave the best results. [20 marks]