

ML Ideas for SFGD

Sebastian Pina Otey

Supervisors: Vicens Gaitán, Federico Sánchez, Thorsten Lux, Anna Espinal

Grupo AIA/IFAE/SEA

February 22, 2019



GRUPO **AIA**



Institut de Física d'Altes Energies



Servei d'Estadística
Universitat Autònoma de Barcelona

- A proper Monte Carlo simulator is needed to perform future tasks, such as algorithm training.
- There are still many difficult steps to model, such as the cross-talk or missing signals at some spots.
- How can we assess the goodness of MC on simulating data close to real one once a model is proposed?

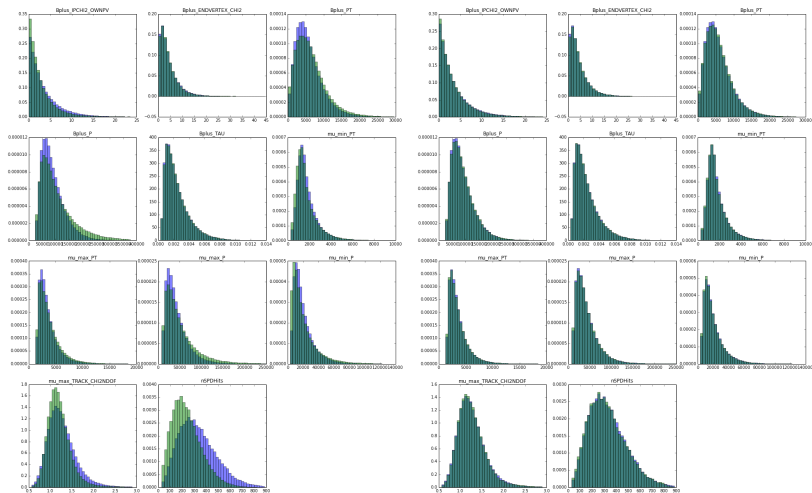
- A proper Monte Carlo simulator is needed to perform future tasks, such as algorithm training.
- There are still many difficult steps to model, such as the cross-talk or missing signals at some spots.
- How can we assess the goodness of MC on simulating data close to real one once a model is proposed?

Maybe Machine Learning can help!

Binary classification

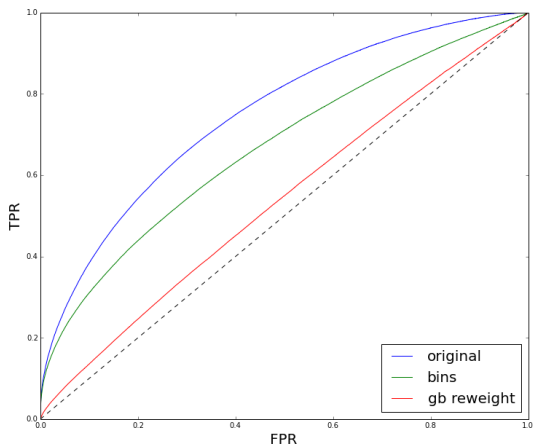
- We have simulated data and real data, two classes of data. There are many algorithms to try to distinguish data labeled as two different things. This task is called **binary classification**.
- Let us see how different data can look like.

Binary classification



Binary classification

The discrepancy can be measured by the receiver operating characteristic (ROC) curve for instance.



Binary classification

- To check if the classifier works good, we first turn off some feature of the simulator (such as cross talk) and then try to distinguish this bad simulation to the real data.
- If the classifier manages to distinguish well, we train it again on good simulations with all the features and check if it manages to distinguish it from real data or not.
- This can asses the goodness of the simulator model.

Cleaning data via AE

Going from the projection of the 3 planes to the 3D representation is not trivial due to missing registers and cross-talk, among other things. There is noise which has to be cleaned and is difficult to process.

Cleaning data via AE

Going from the projection of the 3 planes to the 3D representation is not trivial due to missing registers and cross-talk, among other things. There is noise which has to be cleaned and is difficult to process.

In a simulator, this noise can be turned on and off on command. Hence we can have pairs of data with noise and data without noise of the same events. Can we learn to clean this noise using this data and then apply the cleaning process to real data?

Cleaning data via AE

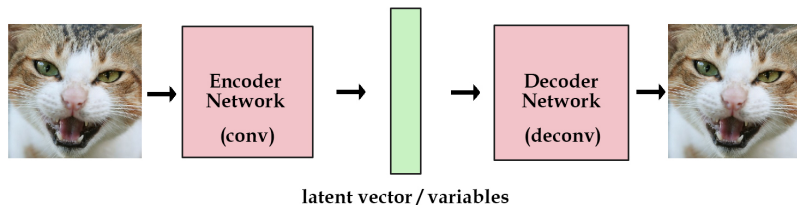
Going from the projection of the 3 planes to the 3D representation is not trivial due to missing registers and cross-talk, among other things. There is noise which has to be cleaned and is difficult to process.

In a simulator, this noise can be turned on and off on command. Hence we can have pairs of data with noise and data without noise of the same events. Can we learn to clean this noise using this data and then apply the cleaning process to real data?

Autoencoders can present such a framework.

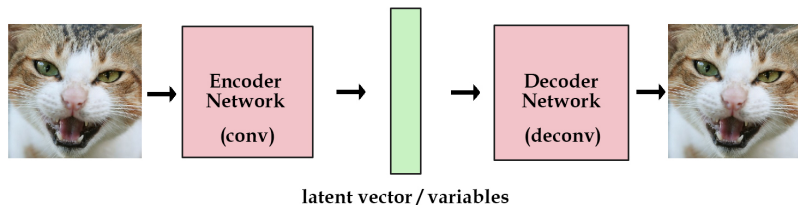
Cleaning data via AE

Autoencoders is a transformation to and from a latent space, in the form of an encoder and a decoder.



Cleaning data via AE

Autoencoders is a transformation to and from a latent space, in the form of an encoder and a decoder.



The algorithm is trained by minimizing a distance between the input and the output.

Cleaning data via AE

Autoencoders can be used to clean noise too. For this, the input is a noisy set, where the output can be compared to the clean set. The goal is then to produce a noiseless output of the input.

Cleaning data via AE

Autoencoders can be used to clean noise too. For this, the input is a noisy set, where the output can be compared to the clean set. The goal is then to produce a noiseless output of the input.

