HotEncoding and Tagging of the Single μ Tracks

Dinupa

NMSU Update October 11, 2022

MLP Neural Network



Charge, position1 (x, y, z), momentum1 (px, py, pz), position3 (x, y, z), momentum3 (px, py, pz)

MLP classification block

3 hidden layers (size = 64) ReLu activation function Learning rate 0.001 Batch size = 64

Predicted hot id

Input features:

Charge, position1 (x, y, z), momentum1 (px, py, pz), position3 (x, y, z), momentum3 (px, py, pz) + predicted hot id

MLP regression block

3 hidden layers (size = 100) ReLu activation function Learning rate 0.001 Batch size = 64

Output features:

Target position (x, y, z), target momentum (px, py, pz)

One Hot Encoding

collimeter | 1

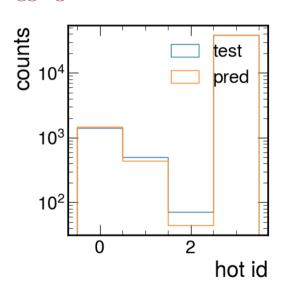
target other beam dump

- We used the OneHotEncoder class in sklearn.
- Encode categorical features as a one-hot numeric array.
- By default, the encoder derives the categories based on the unique values in each feature.

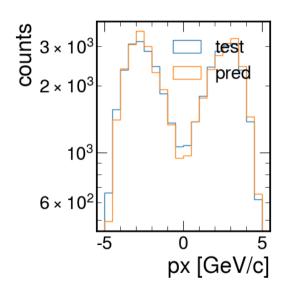
```
position | label | int
```

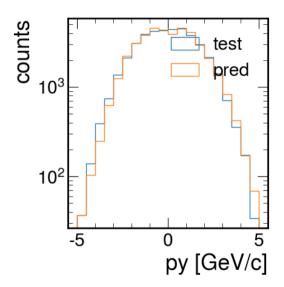
- -800. < z < -305. | collimeter | 0 -305. < z < -295. | target | 1 -295. < z < -1. | other | 2 -1. < z < 500. | beam dump | 3

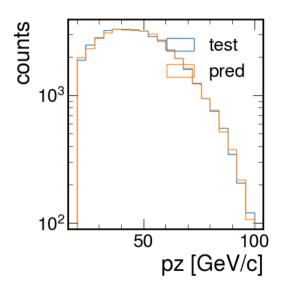
Tagging Task



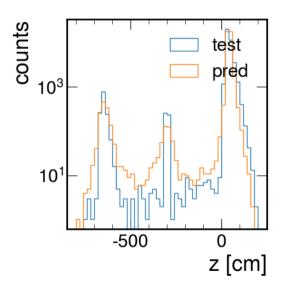
Classification layer almost predict
bins except for bin with
hot_id = 2, with
classification score = 0.9895



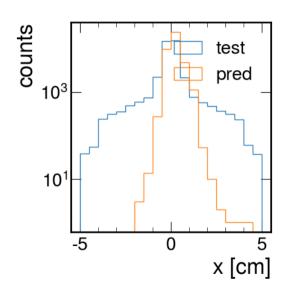


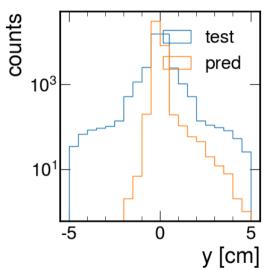


■ vpx, vpy, vpz has a good prediction.



■ With Tagging we can increase the vtz prediction.





Summary

- With tagging we can get a better z-vertex prediction?
- This module was build using MLPClassifier, MLPRegressor classes in sklearn library. Therefore, saving the trained module and getting the loss values after ephocs is not straight formward.
- Since no GPU support is not included it is hard to scale up the module. Better to build the neural network with pytorch. (working on this)