Comparing Performance of ML models to eratio algorithm for electron/jets classification

Lai Wei Sheng

Provided simulated data: *elefull.dat, jetfull.dat, jetfull_jz1.dat, jetfull_jz2.dat, jetfull_jz3.dat*

jetfull and jz1 are data with similar distribution.

Jz2 and jz3 are "fake" data to create jets with higher pt distribution.

	Number of data
Electrons	27524
Jets	45749

Figure 1: Table of simulated data

Pre-Selection Cuts

(Simply_Parameter_Cut.ipynb)

To have a fair comparison between ML algorithms and eratio, some easy cuts were made to full data. (ie: Reta, ws2, rhad, f3)

Below are the required signal efficiency of each parameter and their cut value:

eFEX Cut Parameter	Cut Signal Efficiency (%)	Cut Value
Reta	98	0.85215
Ws2	99	14604.6
Rhad	99	0.05543
f3	99	0.02539

Figure 2: Table of cut efficiency and value for each simple parameter

A final signal efficiency of 95.41% is achieved.

Remaining data:

	Number of data
Electrons	26261
Jets	6593

Figure 3: Table of data remained after parameter cut

ML Model Training

(ML_training.ipynb)

Remaining data is then mixed and split into train, test and validation dataset.

	Number of Data
Train	26282 (80%)
Validation	3286 (10%)
Test	3286 (10%)

Figure 4: Table of data split

ML models are trained with 3 different methods:

- Original image
- Normalised image using maximum energy deposited in 3x17 frame
- Normalised image using summation of energy deposited in 3x17 frame

(Model architecture/parameters/number of trainable parameters can be found in "Model Architecture" word file).

ML models are tested on 10% testing dataset and their rejection value at 90% signal efficiency is found.

1st Test (Overfitting Test)

(Overfitting_Test.ipynb)

Saved ML models are then tested on the full data after cut to look for signs of overfitting. (Note that for CNN, results are not reproducible because of the use of GPU Cuda.)

We are comparing the performance between models tested on 10% of full data which are unseen and models tested on full data.

		Rejection	
		10% Test	Jz sample
CNN	Original	0.1118	0.1059
	Normalised(max)	0.1025	0.1145
	Normalised(sum)	0.0994	0.1128
KNN	Original	0.1769	0.1563
	Normalised(max)	0.182	0.1338
	Normalised(sum)	0.176	0.1346
Decision Tree	Original	0.2343	0.171
	Normalised(max)	0.2845	0.2303
	Normalised(sum)	0.2583	0.2122
RandomForest	Original	0.1102	0.0306
	Normalised(max)	0.1351	0.0537
	Normalised(sum)	0.1211	0.0479
XGBoost	Original	0.1071	0.0174
	Normalised(max)	0.0932	0.0177
	Normalised(sum)	0.1413	0.0849
NeuralNet	Original	0.1537	0.1643
	Normalised(max)	0.1304	0.1257
	Normalised(sum)	0.1537	0.1445
AdaBoost	Original	0.1335	0.1292
	Normalised(max)	0.1351	0.1426
	Normalised(sum)	0.1366	0.1409

Figure 5: Table of rejection value @90% to check for overfitting

Models like RandomForest or XGBoost which has a drop in rejection value is suggesting overfitting. CNN, which is a parametric model, performs well without significant sign of overfitting.

2nd Test (Turn-on Curve)

(Turn-on Curve.ipynb)

With each model having their cut value at 90% signal efficiency, a plot of signal efficiency against electron truth pt is plotted to study the pt dependence of all models.

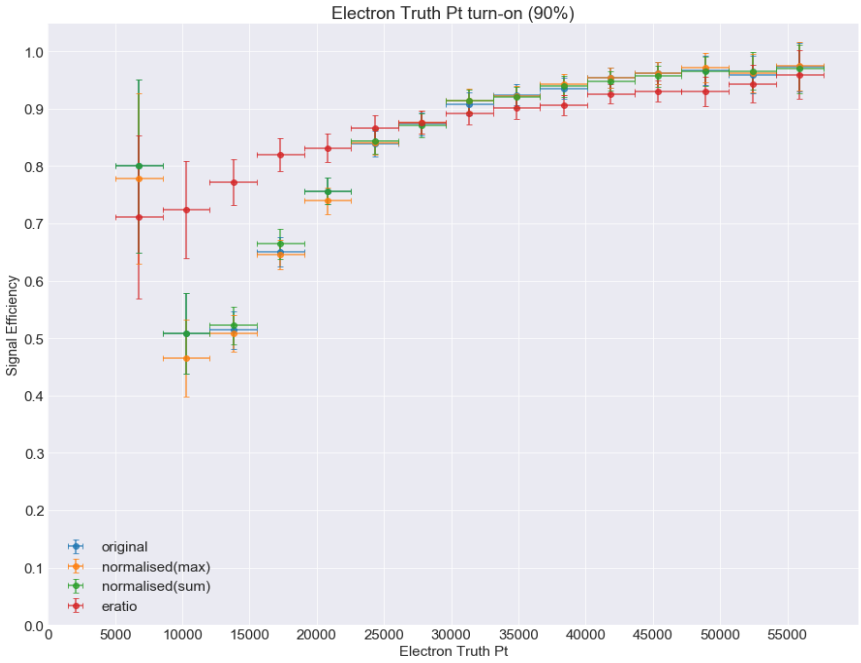

Figure 6: Graph of signal efficiency of CNN model against electron truth pt

As seen in Fig. 6, for CNN, all training method have a higher drop in signal efficiency than eratio algorithm at low pt region. This is suggesting that CNN is doing a better job than eratio at rejecting jets by sacrificing electrons at lower pt region. (CNN has higher Pt dependence).

Trigger Tower Pt Cut

(Simple_Parameter_Cut.ipynb)

(Turn-on Curve after Tower Cut.ipynb)

We notice that after 30GeV, CNN and eratio both have a relatively flat turn-on curve.

For a fair comparison of performance, an extra cut is added (trigger tower pt cut) as an attempt to removing electrons and jets with truth pt lower than 30GeV.

A distribution of trigger tower pt for electrons lower than 30GeV and higher than 30GeV is plotted. A cut value of 30GeV to trigger tower Pt is found to remove most electrons with truthpt lower than 30GeV.

	Number of data
Electrons	11978
Jets	1607

Figure 7: Table of data remained after trigger tower pt cut

The models are then tested on the remaining cuts to obtain a new rejection value with their turn-on curve plotted.

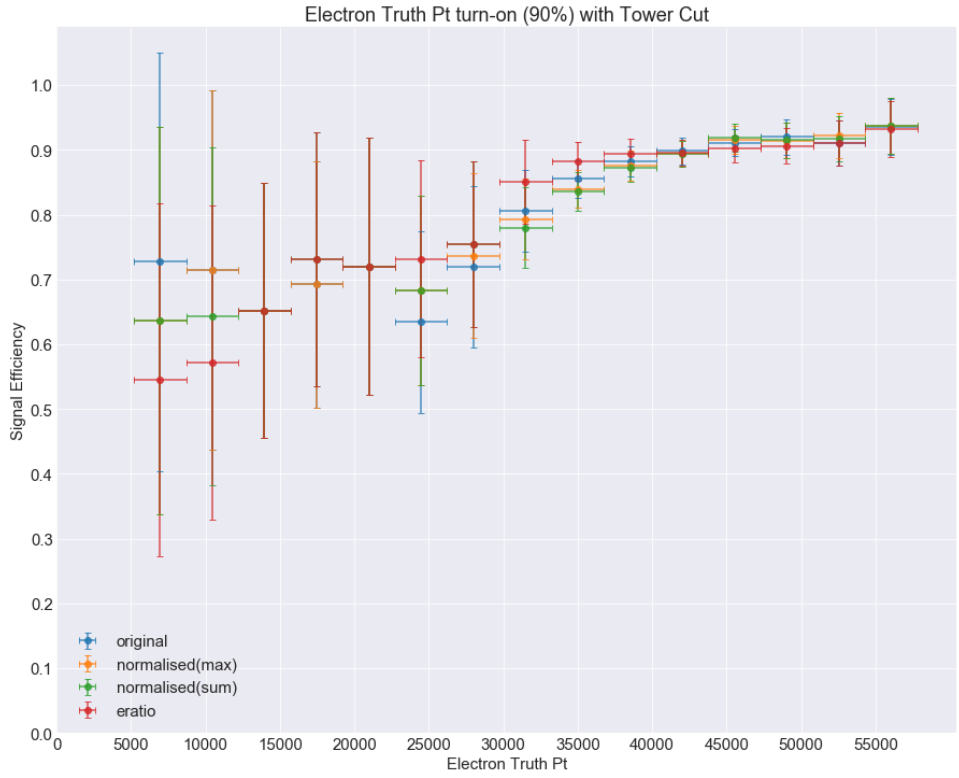

Figure 8: Graph of signal efficiency against electron truth pt on data after trigger tower pt cut for CNN.

		90% Signal eff	
		Rejection before 30GeV cut	Rejection after 30GeV cut
CNN	Original	0.1059	0.1157
	Norm(max)	0.1145	0.1437
	Norm(sum)	0.1128	0.1425
Eratio		0.3044	0.2004

Figure 9: Table showing comparison of performance between CNN and E-ratio algorithm

Finally, we can make a fair comparison of performance between CNN and eratio.

“CNNoriginal” would perform twice better than eratio at rejecting jets.

		95% Signal eff	
		Rejection before 30GeV cut	Rejection after 30GeV cut
CNN	Original	0.1612	0.1543
	Norm(max)	0.1728	0.1904
	Norm(sum)	0.1805	0.1917
Eratio		0.4048	0.2558

Figure 10: Table of rejection value for CNN and E-ratio at 95% signal efficiency

It is worth noting that for 95% signal efficiency, we have a looser cut for jet thus a higher fake rate is obtained.

Training with Flat Pt Distribution

(ML_training_flat.ipynb)

To lower the Pt dependence of trained ML models. A new dataset is made so that electrons and jet training dataset would have a same trigger tower pt distribution.

	Number of data
Electrons	5281
Jets	5314

Figure 11: Table of data with a flat pt distribution

All models are then trained and saved in a new folder.

1st Test (Overfitting Test)

(Overfitting Test (Flat).ipynb)

		Rejection	
		10% Test	Flat Jz sample
CNN	Original	0.2079	0.1771
	Normalised(max)	0.1891	0.1639
	Normalised(sum)	0.1742	0.1823
KNN	Original	0.297	0.2771
	Normalised(max)	0.2616	0.214
	Normalised(sum)	0.2379	0.2063
Decision Tree	Original	0.3355	0.2336
	Normalised(max)	0.3206	0.2411
	Normalised(sum)	0.321	0.2254
RandomForest	Original	0.2097	0.0194
	Normalised(max)	0.2255	0.0294
	Normalised(sum)	0.2397	0.028
XGBoost	Original	0.2097	0.023
	Normalised(max)	0.2041	0.0183
	Normalised(sum)	0.1929	0.0198
NeuralNet	Original	0.2622	0.2416
	Normalised(max)	0.1985	0.1818
	Normalised(sum)	0.221	0.1989
AdaBoost	Original	0.2266	0.1897
	Normalised(max)	0.2378	0.2038
	Normalised(sum)	0.2341	0.1908

Figure 12: Table of rejection value @90% to check for overfitting

Overfitting test was done on data with flat distribution. As notice again, RandomForest and XGBoost are believed to overfitted to training dataset.

2nd Test (Test on non-flat data after cut)

The saved model are tested on jz sample after cut (non-flat) to obtain a rejection value

		Cut Value	Jz sample	
			Signal efficiency	Rejection
CNN	Original	0.5258	0.9	0.1599
	Normalised(max)	0.6013	0.9	0.1185
	Normalised(sum)	0.6695	0.9	0.127
KNN	Original	0.6695	0.9083	0.206
	Normalised(max)	0.8	0.9246	0.2201
	Normalised(sum)	0.7667	0.9208	0.2087
Decision Tree	Original	0.5351	0.9072	0.2319
	Normalised(max)	0.531	0.9364	0.2771
	Normalised(sum)	0.5909	0.9078	0.2425
RandomForest	Original	0.5757	0.9	0.0458
	Normalised(max)	0.5867	0.9	0.0599
	Normalised(sum)	0.5839	0.9	0.0582
XGBoost	Original	0.6104	0.9	0.0461
	Normalised(max)	0.6557	0.9	0.041
	Normalised(sum)	0.6466	0.9	0.0435
NeuralNet	Original	0.4223	0.9	0.2231
	Normalised(max)	0.6116	0.9	0.1538
	Normalised(sum)	0.6183	0.9	0.1696
AdaBoost	Original	0.5002	0.9	0.154
	Normalised(max)	0.501	0.9	0.1559
	Normalised(sum)	0.5011	0.9	0.1486
Eratio		0.8358	0.9	0.3044

Figure 13: Table of rejection value for model tested on full jz sample.

3rd Test (Turn-on Curve)

(Turn-on curve (Flat).ipynb)

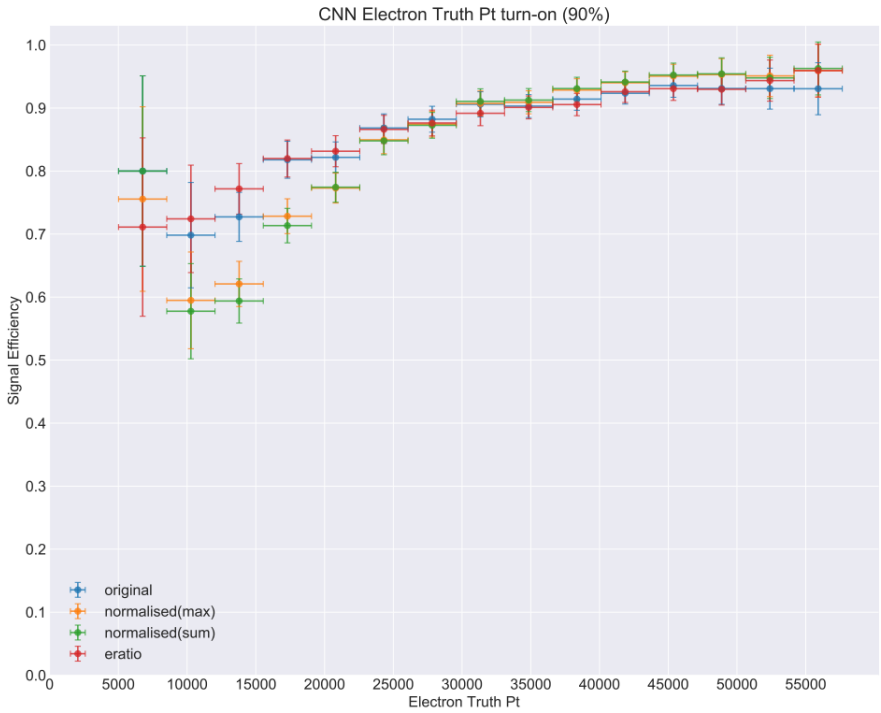

Figure 14: Graph of signal efficiency of CNN model against electron truth pt

CNN turn-on curve is found to have a smaller drop in signal efficiency compared to model trained with non-flat distribution. Take note that for CNN original, turn-on curve is very similar to eratio algorithm. Thus, we can say that this is a fair comparison of rejection power between CNN original and eratio.

It is found again that CNN original performs twice better than eratio at rejecting jets.

Summary

In our attempt to use machine learning techniques to identify jets and electrons, specifically CNN. We have found out that firstly, training ML models using original data or normalised data does not have much effect on performance of models. Also, it is proven that CNN performs better than e-ratio algorithm in rejecting jets while maintaining a signal efficiency of 90%. After a study of turn-on curve, we notice that CNN outperforms e-ratio by sacrificing electrons at low pt region (<30GeV). After removing most events recorded at low pt, a comparison of rejection value once again showed that CNN performed twice as better than e-ratio algorithm. Also, an approach of training machine learning models using electron and jet data with same trigger tower pt distribution have shown to reduce the pt dependence of ML models while still performing twice as better than eratio.

Improvements

- A hyperparameter scan could be done for better optimization of model structure.
- More simulated data could be produced to better train models.
- Original input data frame of 3x17 could be increased to include more cell information.
- Pt dependence of other machine learning models could be studied (turn-on curve similarly to CNN). This could be done by changing the model used in "*Turn-on curve.ipynb*".
- Other machine learning models could be used (Graphical Neural Network).
- CNN could be used to study rejection power of individual parameter to search for any other simple parameter cut.