IMPROVEMENTS FOR HEP DATA/DATASETS DATA COMPRESSION

EHIZOJIE ALIGBEH

[GSoC-application-baler]

INTRODUCTION

In this presentation, I will share my report on applying Baler to a given particle physics dataset (data).

Baler is a machine learning based compression tool for scientific data.

Baler is a tool used to test the feasibility of compressing different types of scientific data using machine learning -based autoencoders.

The goal is to minimize the difference between the mass calculated before and after compression (this value is found in ./projects/example/plotting/analysis.pdf after running the analysis), thereby improving the Baler tool

The Dataset here is a root data (DAA238E5-29D6-E511-AE59-001E67DBE3EF.root)

SETUP

```
cd GSoC-application-baler
poetry install
```

Procedure

```
poetry run python baler --project=example --mode=new_project
poetry run python baler --project=example --mode=preprocessing
poetry run python baler --project=example --mode=train
poetry run python baler --project=example --mode=compress
poetry run python baler --project=example --mode=decompress
poetry run python baler --project=example --mode=evaluate
poetry run python baler --project=example --mode=analysis
```

Place the working data (example.root) in the data directory[/data/example/]
[i.e. /data/example/example.rppt]

Configuration (configuration of interest):

Via _config.py(helper.py):

```
path_before_pre_processing = "data/example/example.root"
epochs                     = 10
early_stopping             = False
lr_scheduler               = True
patience                   = 20
min_delta                  = 0
model_name                 = "george_SAE"
custom_norm                = False
l1                         = True
reg_param                    = 0.001
RHO                        = 0.05
lr                         = 0.001
batch_size                 = 512
save_as_root               = True
test_size                  = 0.15

via utils.py
factor = 0.5
min_lr = 1e-6
```

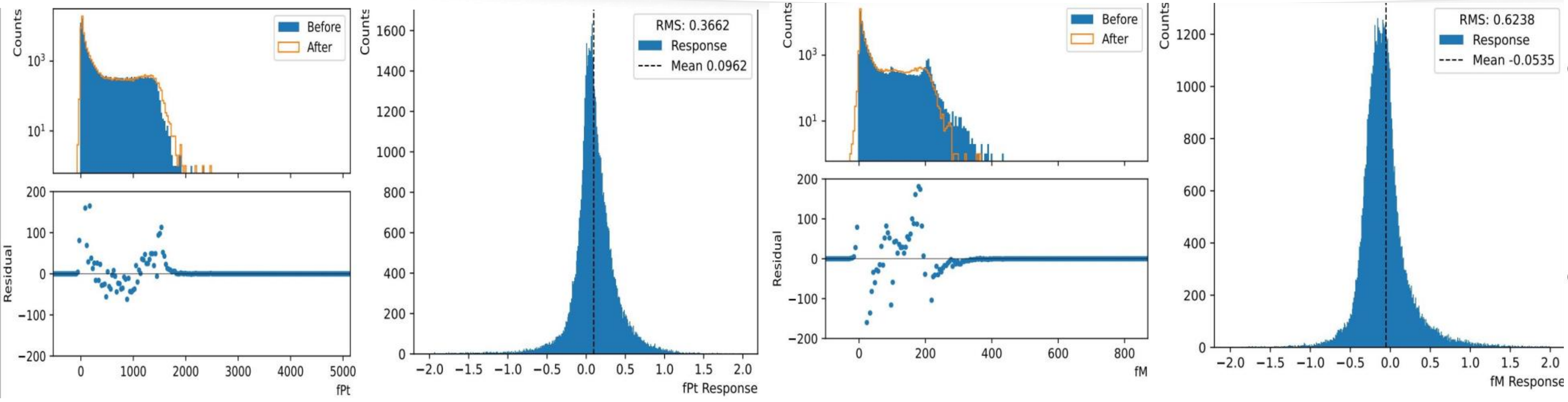
Via utils.py:

```
factor = 0.5
min_lr = 1e-6
```

IMPROVEMENTS

Initial Results;

from the initial setup, the results of the applying the baler compressor tool on the given data were;



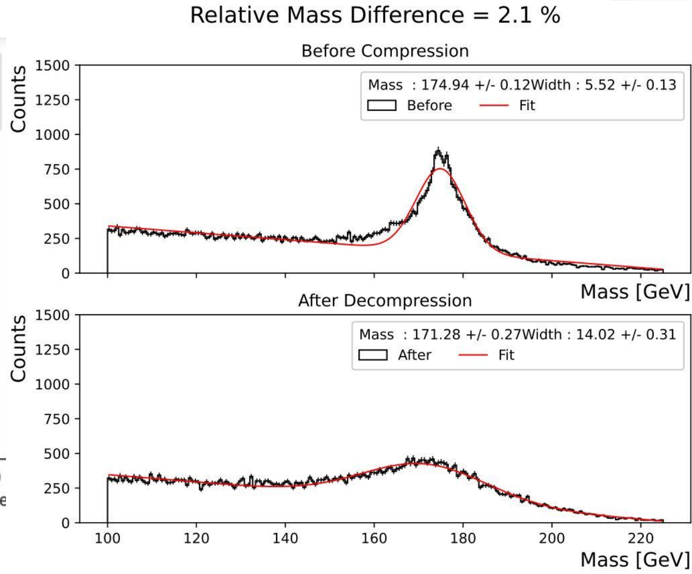


Fig: plots of pt(fpt) residuals and response, mass(fM) residuals and response and relative mass difference

I have considered some improvements which should achieve better when using the baler compressor tool.

1. Normalization Techniques: I have implemented a Custom Normalization technique in-place of the default standard Min-max scaler. The improved normalization technique produces a better relative error in the mass calculated difference, as this technique has a better fitting/performance in normalizing the data

2. Autoencoder Model Variation: I have implemented a Sparse Autoencoder Model (george_SAE) as opposed to other Neural Networks which are unable to generate a good model (The choice of which activation function to use can have a big impact on the performance)

3. Modified Training Procedure and Modified Training Utilities: In implementing my improvements, I have configured some procedures such as the utilization of early stopping, adjusted discount_factor and learning rate

IMPLEMENTATION:

I observed maximum improvement by optimizing the normalization function; manually scaling each variable improving the data distribution and in-turn the optimizing the mass calculated difference after compression

Also, improvement to Autoencoder model was applied; this involved modifying the input layer to a larger one, and reducing the number of dropouts

Improvement to the training procedure and training utilities were; adjusting the discount_factor to a value of 0.9, early_stopping and epochs value of 50.

RESULTS

Improvements applied Results;

IMPROVEMENTS:

Normalization Techniques; changes applied to "normalize" function (preprocessing.py). Autoencoder Modification; changes applied to "george_SAE" model. Training and Configuration: changes applied to utilities.py. Also changes applied to helper.py to allow for the aforementioned changes

SETUP(modifications)

Configuration (configuration of interest):

Via _config.py(helper.py):

```
custom_norm         = True
```

Via utils.py:

```
factor = 0.5
min_lr = 1e-6
```

From the improvements, the results of the applying the baler compressor tool on the given data were;

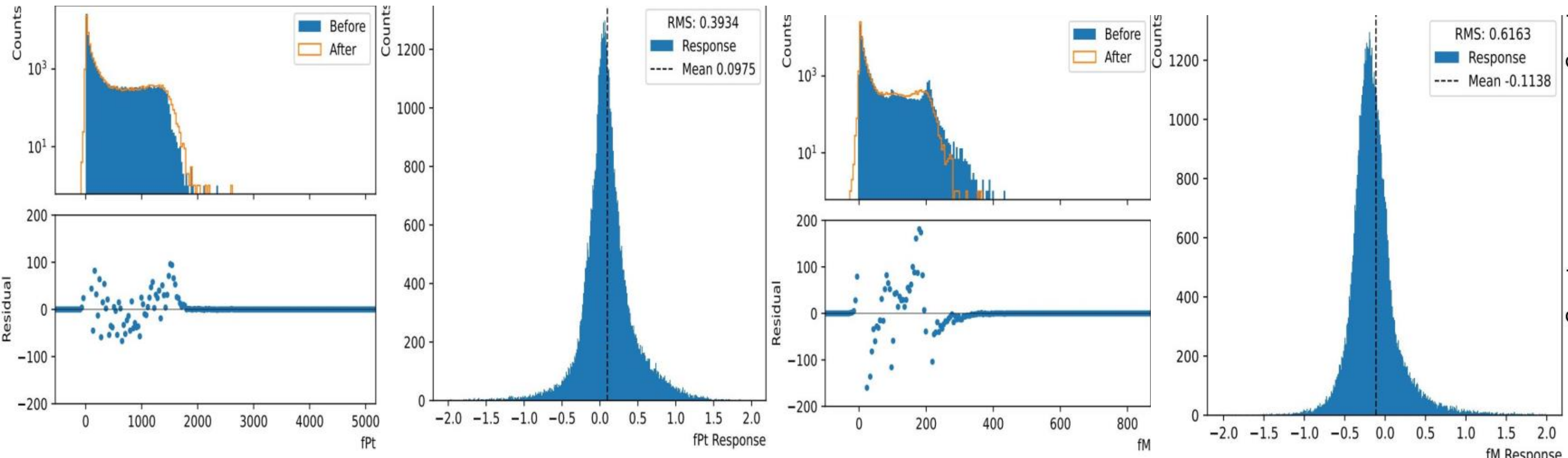
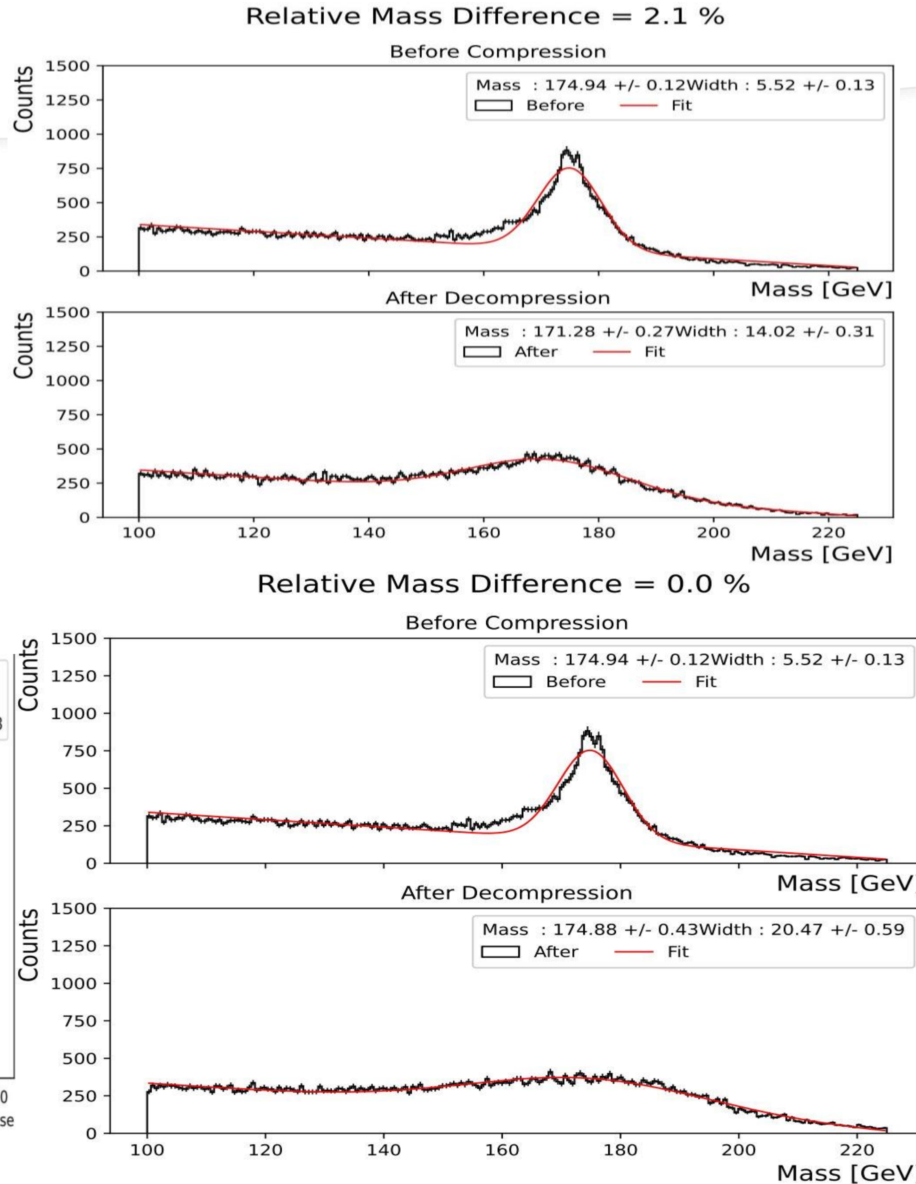


Fig: plots of pt(fpt) residuals and response, mass(fM) residuals and response (improvements)

Comparison of Initial Results to the Improvements



DISCUSSION

Improvement vs Initial Result:

Reviewing the Initial Results and the Improvements;

Custom Normalization techniques applied significantly improved the results of mass calculated difference. Utilizing the standard normalization techniques would have resulted in mass calculated difference of about 3%, whereas, the applied custom normalization technique resulted in mass calculated difference of less than 0.5%

The modification the Autoencoder results in much less significant improvements in the mass calculated difference. Adjusted training resulted in much longer training times with less significant results in mass calculated difference

Why do the Improvements work?

Given the compression techniques, the particles data of the jet-trained model are sensitive to the normalization approach utilized, thus producing significantly better results when appropriate normalization techniques are applied.

While modification of autoencoder models and training procedure produce improvements, the significance of the improvements are subtle across various Standard Autoencoder models and training configuration

What could be improved further?

Further research on more appropriate normalization techniques would be effective, and utilization of custom autoencoder models would be worth exploring (models such as the Adversarial Autoencoders and VAEs)

CONCLUSION

Evaluation(score) Vs Good Analysis result:

In the course of the analysis, it was observed that extensive analysis provided much more valuable results than an evaluation/error score. Despite having great mass calculated difference, some configurations did not provide expected performance after compression. Extensive analysis(including graphs) showed that these configurations did not appropriately fit the particle features and were invalid despite have great evaluation score. Indicating that evaluation score alone is not sufficient in determining performance of the modifications made

Fundamental Flaw with Baler:

- Inherent flaw from Standard Autoencoder models, flexibility of analysis methodology

Comparison of Initial Results to the Improvements