GSoC 2021 ATLAS Autoencoder Project

- Evaluation Exercise -

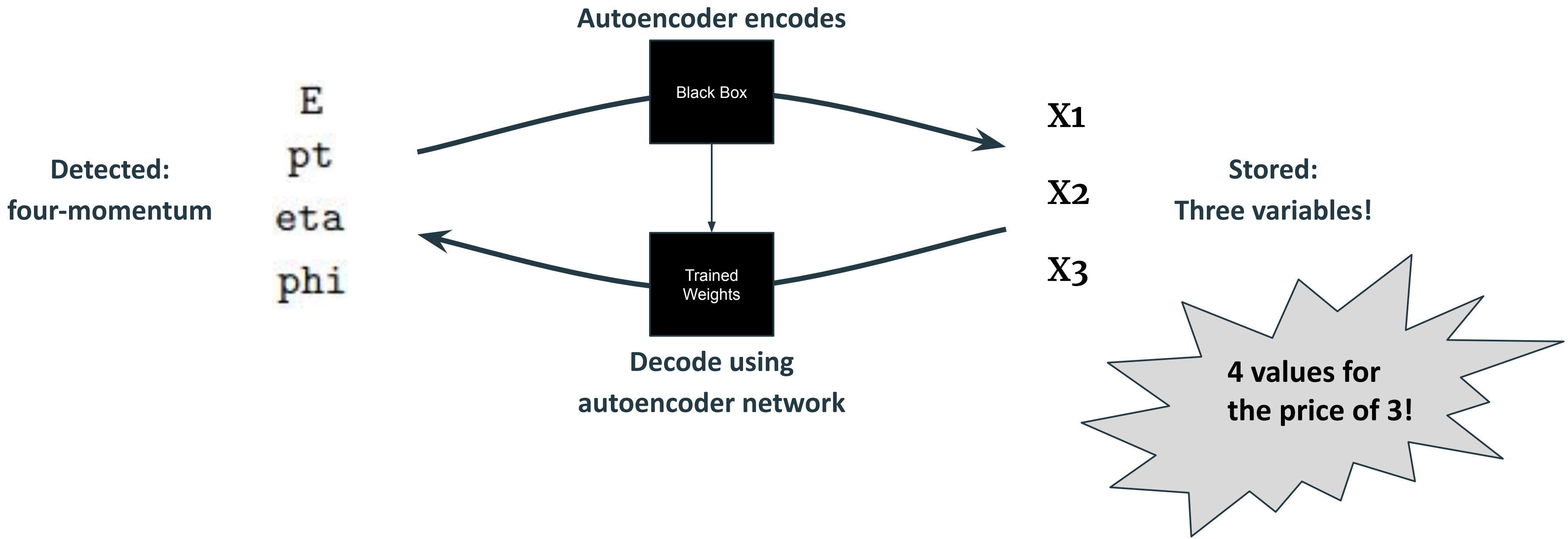
Zachary Martin

Task

- Create a deep autoencoder (AE) using the ATLAS data, monojet_Zp2000.0_DM_50.0_chan3.csv
    - Compress the four momenta of jets into three variables
    - Analyze the performance/accuracy of the autoencoder

Motivation

- Too much data obtained in particle detector -> need to compress
- Solution: train a neural network to link the four momentum to three variables (storage usage reduced by ~25%)!



First: Process the Data

Datafile processed in Python line by line:

1. Split line by ‘;’
2. Only take elements after ‘METphi’
3. Check ‘j’ is inside (to choose <u>only jets</u>)
4. Store the remaining four momentum string
5. Convert stored values into Pandas DataFrame

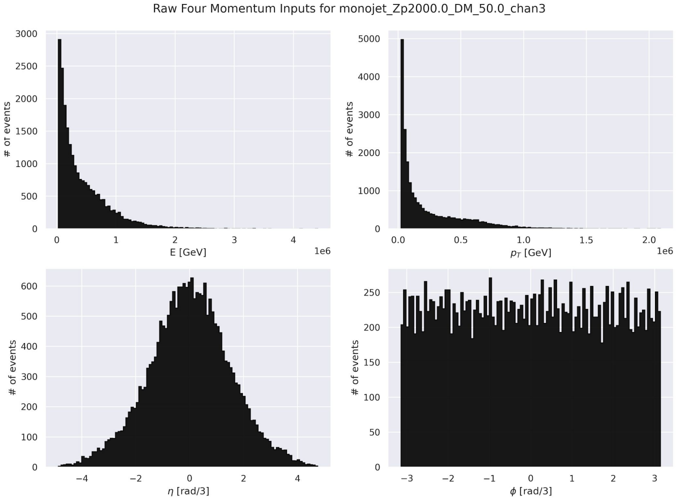
Data Entry Format:

```
event ID; process ID; event weight; MET; METphi;
obj1, E1, pt1, eta1, phi1; obj2, E2, pt2, eta2, phi2; ...
```

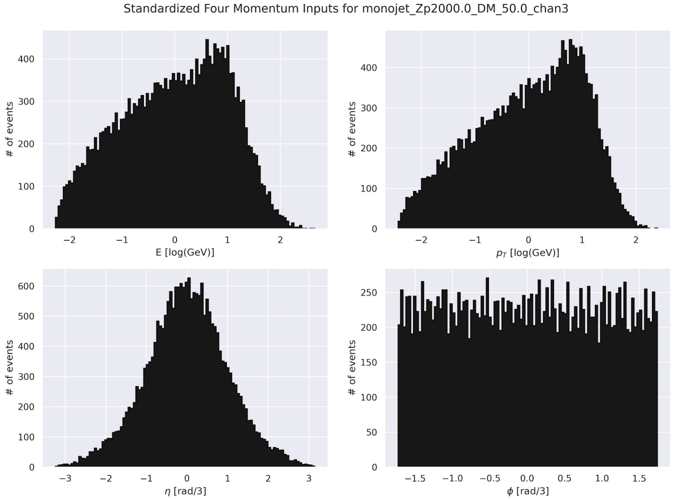
Source: https://zenodo.org/record/3961917/files/dataset.pdf

For deep learning training, we need to scale the data (E, pT >> eta, phi and so carry larger influence on the autoencoder training)

- Here, we used log(E) and log(pT), and Standardization of all input variables (mean=0, std=1)
    - Note that the standardization is reversible when using a specific scaler object in the Python package sklearn
- (Beyond task, not included here: found that log without standardization gives similar results, so log only suffices in this case)



Log E, pT
& Standardized



Prepare Data for Training Autoencoder

- Data was shuffled to prevent any sorting bias (originally sorted by event ID, process, weight, MET, METphi)
- Data was then split into a common training data vs testing data split of Train: 80%, Test: 20%

Define the Autoencoder Model

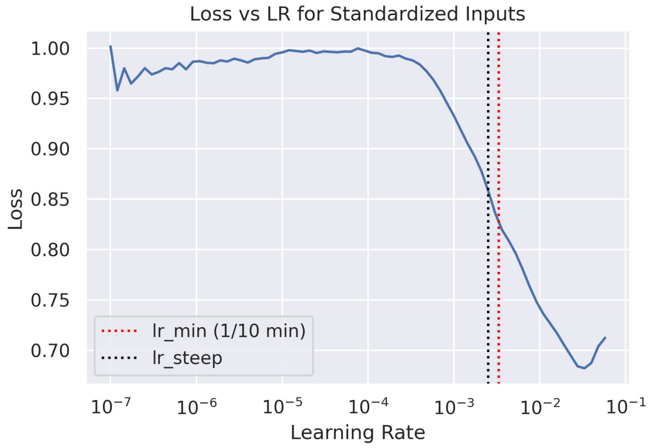
Used model given in the example notebook (Network Layers: 4 -> 200 -> 200 -> 20 -> 3)

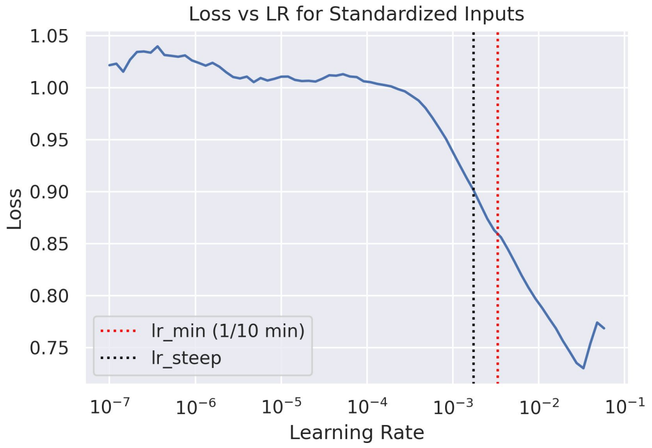
Choose the Hyperparameters - Learning Rate aka How Fast AE will Search for Loss Minimization

Choose the learning rate by slowly increasing and observing the loss [~ $(\text{data - prediction})^2$ / data] in small batch trials

- Loss is expected to decrease until a sudden increase
- Choose either: the rate giving the largest change (lr_steep) or the rate at 1/10th of the rate where loss is minimal (lr_min) (see Leslie N. Smith's approach: https://arxiv.org/abs/1803.09820)

While the algorithm does not always produce the same result, note the consistency below… 1e-3 < lr_steep, lr_min < 1e-2

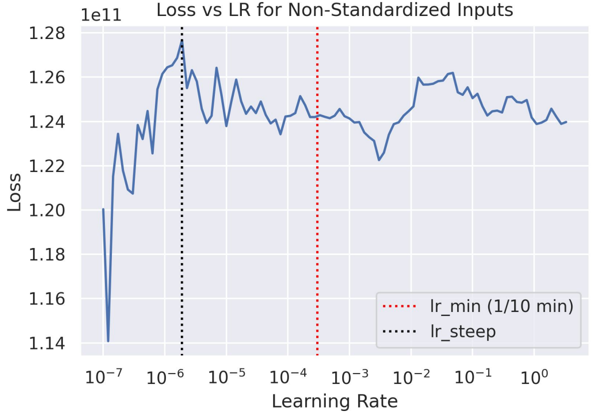


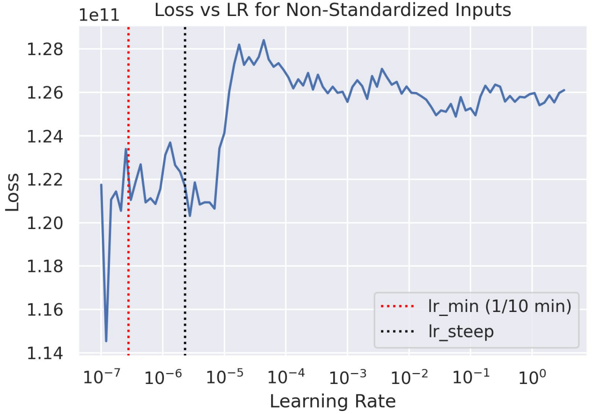


Loss vs LR for independent runs to check for consistency

A Look at the Learning Rate for Non-Standardized Inputs

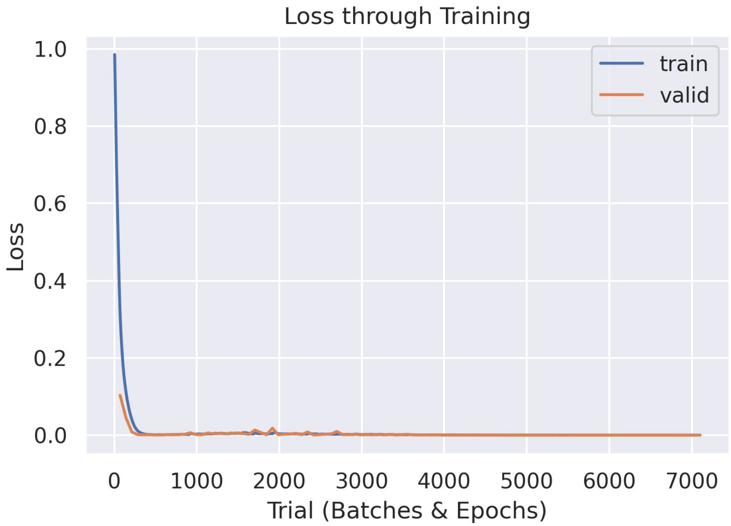
- Without standardization, we lose the expected features (decrease until a sudden increase)
- lr_min and lr_steep are not consistent
- These ugly results emphasize the need to standardize





With a Learning Rate Chosen, Begin Training Autoencoder

- 100 epochs (trials or generations) for the length of training
  - Finishes under 70s (fairly fast for training standards)
- The loss after each epoch shown to the right
  - Final loss value < 1e-5
  - A good loss value is under 1e-3, so the AE has been trained successfully!

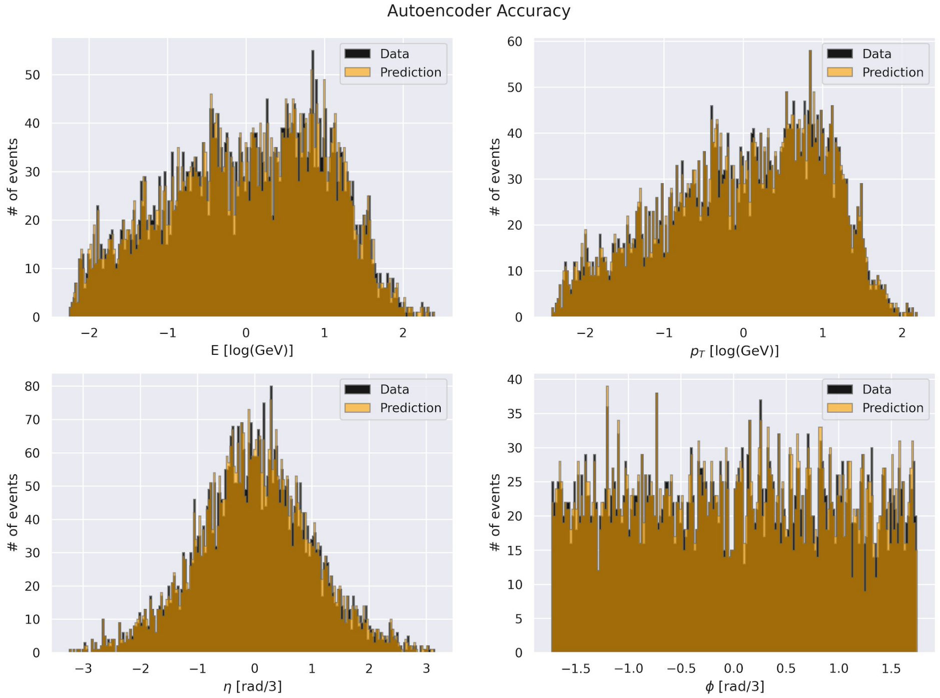


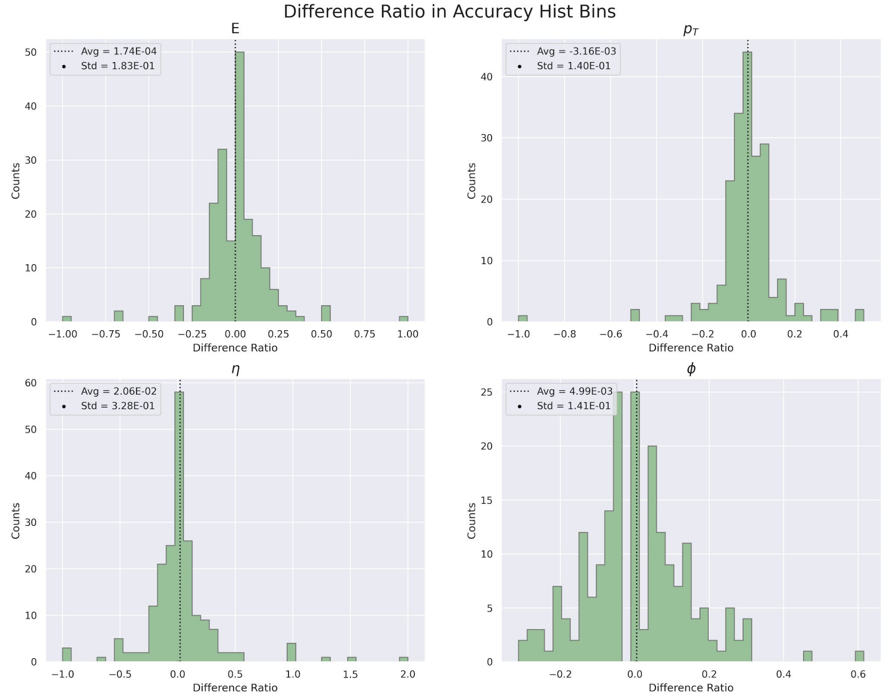
Performance Results

- Can view how the trained AE predicts vs the actual data (left: Predictions fairly match with Data)
- To quantify the accuracy, the difference ratio [(data - prediction) / data] is used (right)
    - Average ratio < 1e-2 for all inputs

A few outliers may need to be investigated to improve the AE (note the relatively large standard deviations due to outliers)

Nonetheless, we can conclude that the AE is working successfully!





Links

This Presentation:
https://docs.google.com/presentation/d/15rzxFdT6WenvKE89sHjRluVEb-sQDs_nGxpfVQNzu7I/edit?usp=sharing

Exercise Notebook: https://github.com/ZackAshM/GSoC2021-ATLAS-Autoencoder

Github: https://github.com/ZackAshM

Notable Projects/Repos:

- Daytime Star Tracker: https://github.com/ZackAshM/stereo
- Geant4 Sims for ACE: https://github.com/ZackAshM/ACE4
- Data Science Exercises (for me to learn): https://github.com/ZackAshM/DataScience4Fun
- Python Bootcamp (for me to teach): https://github.com/ZackAshM/PythonBootcamp

CV: https://github.com/ZackAshM/GSoC2021-ATLAS-Autoencoder/blob/master/.etc/Resume_ZacharyMartin_210309.pdf

Motivation: https://raw.githubusercontent.com/ZackAshM/GSoC2021-ATLAS-Autoencoder/master/.etc/motivation