# Deep Autoencoders for Data Compression

Tanay Mehta
LinkedIn | Email | Github

# The Dataset

- The given dataset consists of 2 files, training set and testing set.
- Since the task is to compress the data (thus reconstructing the given data), it differs ever so slightly from a general neural network problem, difference being the absence of **target variables**.
- Training Data contains ~ **1 Million** data points.
- Testing Data contains ~ **27 Thousand** data points.

# Preprocessing

- Any major pre-processing steps are not taken, except **Mean Normalization** of both Training and Testing Datasets

# Data Before and After Preprocessing

```
train_data.sample(5)
```

| | m | pt | phi | eta |
|---|---|---|---|---|
| 103557 | 4604.136719 | 35358.269531 | -0.046221 | 3.153636 |
| 110606 | 5349.682617 | 23026.968750 | -1.602440 | -1.501314 |
| 7376 | 3982.672363 | 21766.623047 | -0.270745 | -1.086375 |
| 28823 | 5434.583008 | 25144.343750 | -0.663255 | 1.483790 |
| 66111 | 19289.832031 | 190417.234375 | -1.605715 | 0.028735 |

Before Normalization

```
train_data.sample(5)
```

| | m | pt | phi | eta |
|---|---|---|---|---|
| 127010 | 0.387414 | 0.055814 | -1.644587 | -1.433460 |
| 6213 | -0.146326 | -0.405315 | -1.563429 | 0.821581 |
| 102498 | -0.567993 | -0.619722 | -0.585495 | 1.109422 |
| 35092 | -0.604736 | -0.464915 | 0.770662 | -2.035339 |
| 56858 | -0.512914 | -0.559525 | 1.498315 | 0.985856 |

After Normalization

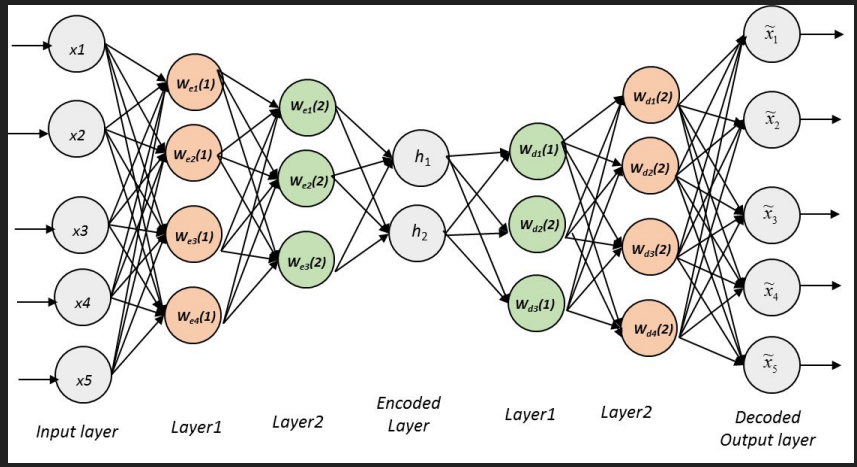
*Both the data samples are drawn Randomly from the dataset

# Deep Autoencoders

- A Deep Autoencoder is a type of Neural Network that can reconstruct the given input data.
- Simply put, It can “**learn**” to generate data similar to what it was provided with.
- This is “learning” is done by reducing dimensions of data by an “**Encoder**” and then increasing the dimensions again by a “**Decoder**”.
- This way, the model learns to express, let’s say 4D data into 3D or less. Although, it doesn’t achieve state-of-the art performance, it’s still usable.

# My Approach

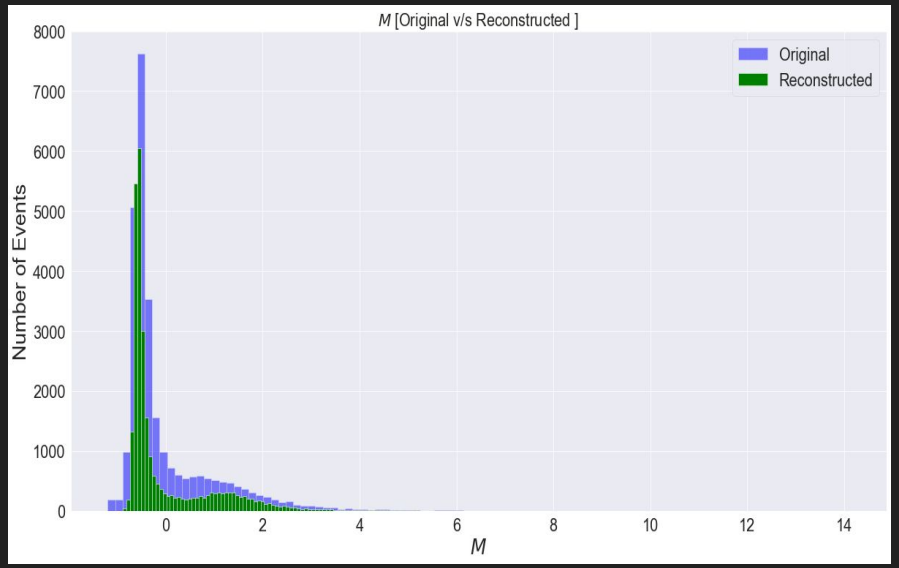
- I have designed a Deep Stacked Autoencoder with 3 Encoding and 3 Decoding Layers, making both encoder and decoder *symmetric*.
- My implementation compresses the data from 4D to 3D (Since it has 3 Latent Dimensions).
- A Sample* Deep Stacked Autoencoder looks like:

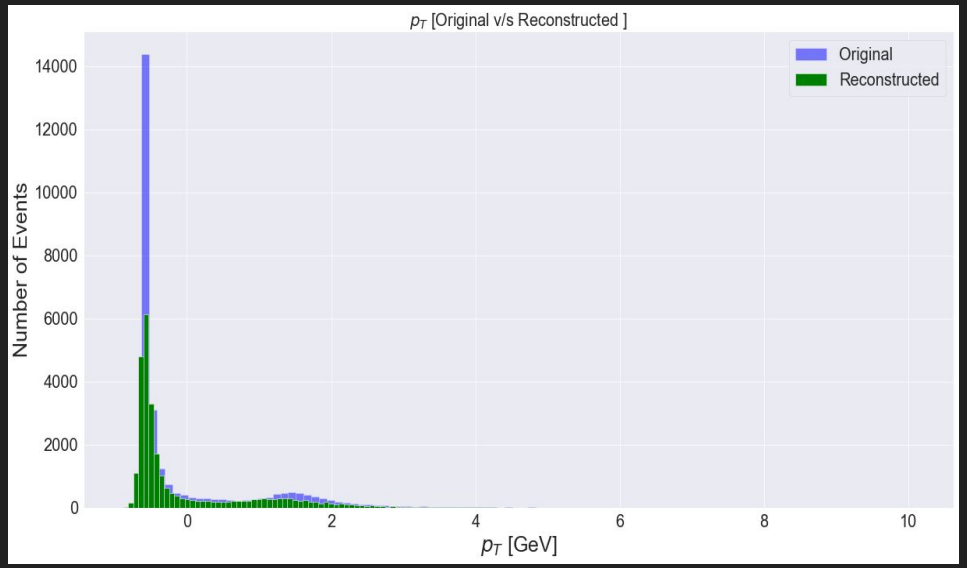



*The above image is a sample image of a DSAE, not the one I made

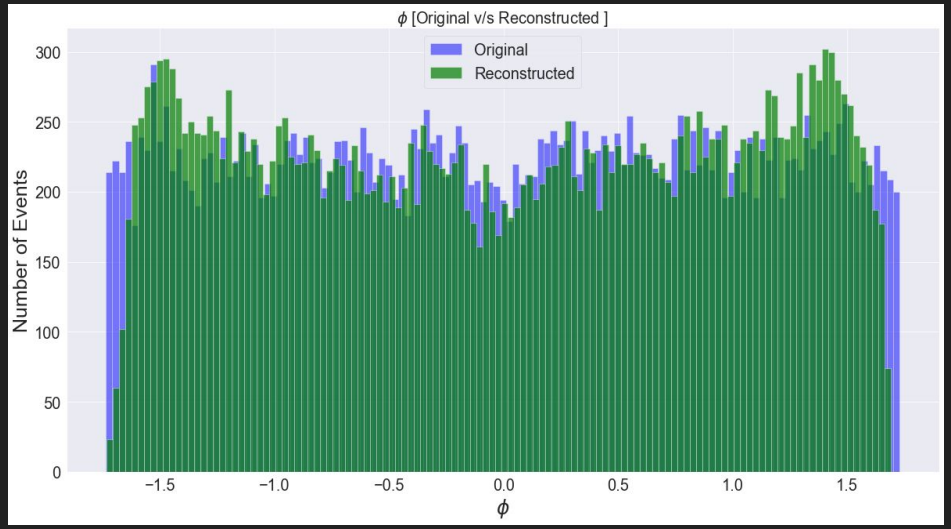
# Results

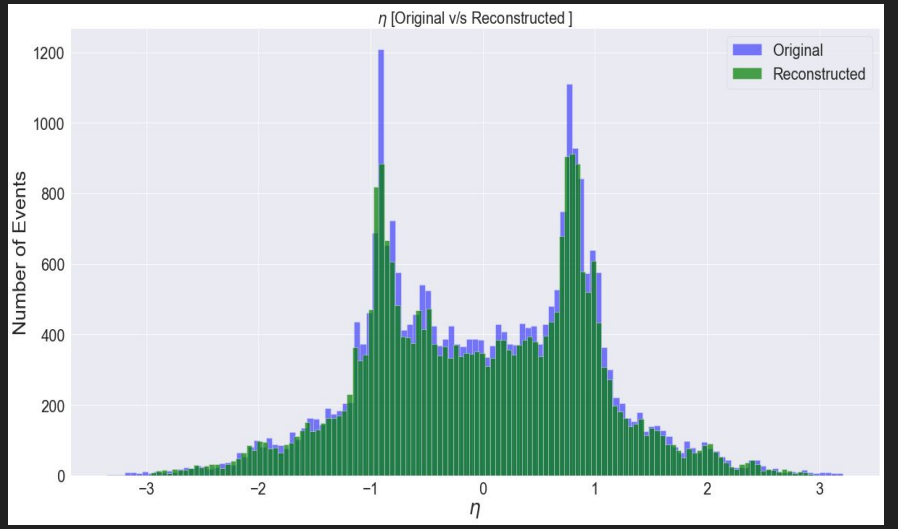
Following are the plots of the 4 Feature vectors from the test data and their reconstruction.



# Results




As seen from former plots, it is clear that the model isn't exactly perfect at reconstructing the data, as a result some of the data may get lost, but better AutoEncoders can be designed to reduce this problem as much as possible.

# Results

Below is the Re-construction Loss (Basically the **element-wise difference between the original data and the reconstructed data, squared!**) , decreasing throughout the training process of 350 Epochs (or cycles).