Present Improvements for HEP Data using Adversarial & Variational AutoEncoders

Deep Autoencoders for scientific compression
GSoC 2023

Mentor - Alexander Ekman
Student: Devesh Marwah

[Github Repository]
[Detailed Report]

Idea behind the project

Going through some of the GSOC projects from 2021, I came across an implementation of KL divergence in loss function for variational autoencoder here.

Instead of implementing the same thing, I have experimented with three completely novel approaches, with two of them achieving significantly better results.

These results can further be improved with layer finetuning, and adequate guidance from esteemed mentors of your organization.

“It is apparent that VAE shows quite poor performance compared to the other two AEs. A reason for that could be the KL divergence term in its loss function which also showed to degrade the performance in the case of the SAE.”

~ GSOC 2021 Project Report

Observations and Improvements

- Wasserstein distance
  Different from the regularizer employed by the Variational Auto-Encoder, WAE minimizes a penalized variant of the Wasserstein distance between the model distribution and the target distribution (VAE).
  After compression: Mass : 171.71 +/- 0.33 Width : -7.24 +/- 0.35
  This is a significant improvement from previous implemented KL divergence loss and the normal encoder.

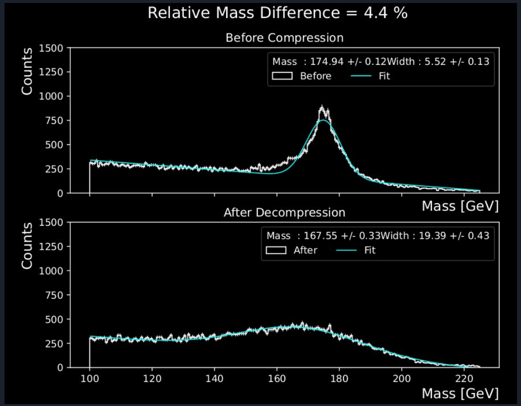
- Sliced-Wasserstein Autoencoders
  Instead of using the KL divergence, I experimented with a different loss functions metric which can be also used in place to improve model's performance. The idea behind is to regularize the autoencoder loss with the sliced-Wasserstein distance between the distribution of the encoded training samples and a predefined samplable distribution (Generally Gaussian).

  Mass : 160.92 +/- 0.58 Width : 17.68 +/- 0.7

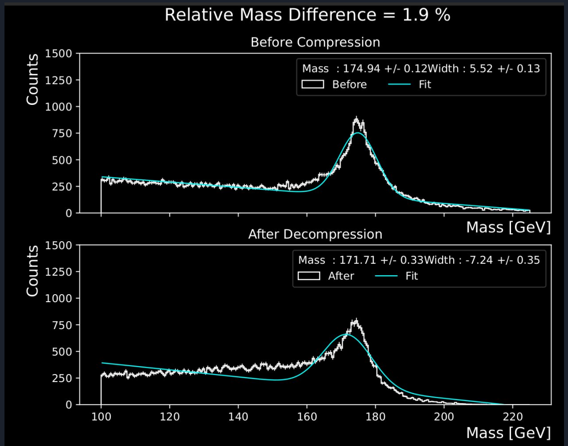
Adversarial Autoencoders

- While getting my hands dirty with the variational AutoEncoder loss function, I realized a seemingly basic problem which had far-reaching consequences.
- The integral of the KL divergence term does not have a closed form analytical solution except for a handful of distributions hence It is not straightforward to use discrete distributions for the latent code z_dim. This is because backpropagation through discrete variables is generally not possible, making the model difficult to train efficiently.
- Results after compression: Mass : 167.77 +/- 0.29 Width : -19.35 +/- 0.37

Adversarial Autoencoder



Variational Autoencoder Wasserstein Auto-Encoders



Implementation details

- The adversarial is implemented separately in “adversial” as config.model_name employed.
- Traditional autoencoders frequently provide a low-dimensional representation that is unhelpful or meaningless for subsequent tasks. Adversarial training can be useful in this situation.
- To distinguish between actual data and false data produced by the autoencoder, a second neural network called the **discriminator** must be trained using adversarial training. After that, the discriminator instructs the autoencoder to produce fake data that is identical to the genuine data.
- **Why it Works**: Adversarial autoencoders avoid using the KL divergence altogether by using adversarial learning. In this architecture, a new network is trained to discriminatively predict whether a sample comes from the hidden code of the autoencoder or from the prior distribution. Here the distribution is gaussian. The loss of the encoder is now composed by the reconstruction loss plus the loss given by the discriminator network.
- **A detailed report on working and result discussion can be found here**

Discussion & Future Scope

- A combination of Adversarial Networks along with Convolutions can be tested in order to capture the spatial relations in the data which are necessary to look over for physics data especially as it is possible for data to follow some hidden latent distributions.
- New loss functions can be experimented with instead of KL divergence loss to test the VAE.
- The current implementation of adversarial is bare bones and can be modified further with more depth in layers and stronger GPU can be needed for them.
- Evaluation can be a better metric than analysis as it shows the nitty gritty of the performance and as in the VAE analysis we notice how adversarial model fits the curve better than the Variational Autoencoder. However analysis results can be used for crude approximation and get an eyeball view on the dataset.