Name - Sandipan Majhi
Email - sandipan.majhi.work@gmail.com

Preprocessing steps and rationale

The dataset is of 448 dimensions. We wish to reduce the dimensionality of the dataset. We observe that features are within 0 and 1 and so we do not try to scale them or try any extra feature engineering. On the other hand the target is very much varied but positive. It has some outliers. This is why we go for MinMax Scaling in this case. The outcome is that we create targets ranging in between 0 and 1. Now this enables us to include Sigmoid function as the output activation and does not let the neural network to explode in gradients and create unlikely results. This stabilizes training hugely.

Insights from dimensionality reduction

From our experiments we find that reducing dimensions to 32 works well. This part significantly reduces computation because we need a smaller model to train on the regression output. We tested two specific use methods for dimensionality reduction, PCA and Variational Auto-Encoder. In VAE, we assume the standard normal distribution as the posterior distribution for the latent dimension. But because we only have 500 data this assumption is only an approximation. We found that this works pretty well in creating reduced dimensional features. This is where PCA takes over the VAE model and we find that the latent features generated by PCA works better here.

Model selection, training, and evaluation details

The VAE model used here is using 3 fully connected layer in encoder, where first hidden layer reduced dimension and next 2 for approximating the mean and log variance. Next, for reconstruction loss, we created a decoder part with two fully connected layers. The loss function used was a reconstruction loss and a KL divergence loss for the latent mean and variance. The best model was saved as a checkpoint.

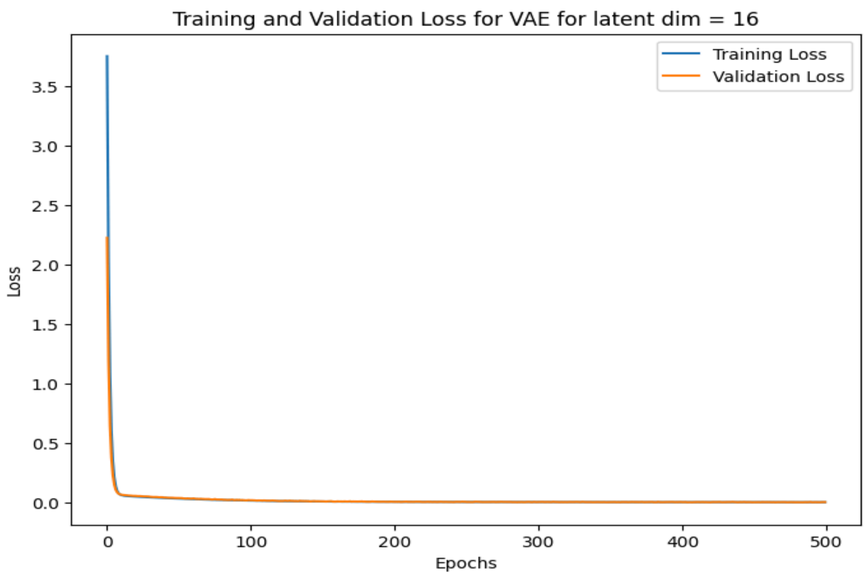
On the other hand, the regression model was a smaller neural network, with 3 hidden layers and 1 output layer. There were intermediate RELU activations but there was Sigmoid activation was used. We used Sigmoid because the output was in between 0 and 1.

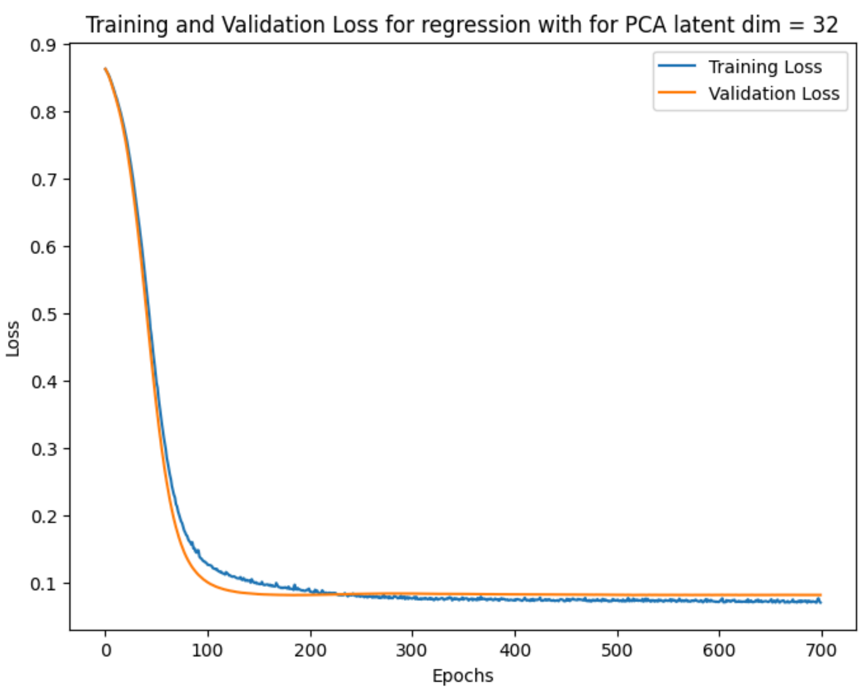
The data used for training the regression model was the low dimension projected data from the original data.

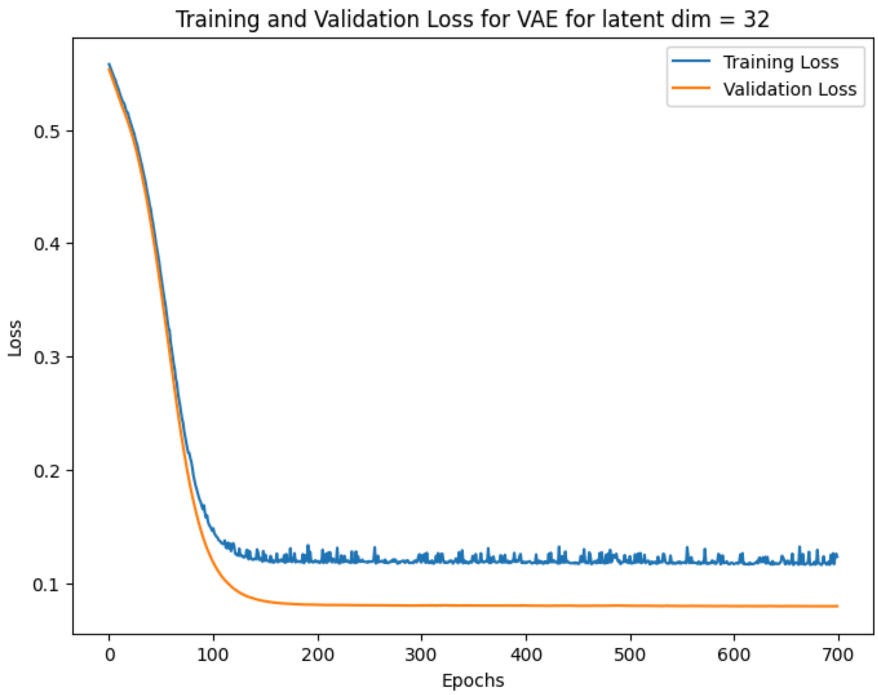
Training was done on projected low dimension data of 32 dimensions. We used a batch size of 32 and for 700 epochs. The results were as follows -

Model	Test Loss
Regression NN + PCA Latent 32 dimension	0.18186
Regression NN + VAE Latent 32 dimension	0.24823

We also show the following training curves,







Key findings and suggestions for improvement.

One of the most important findings was that, although there was only 500 data, VAE worked pretty well in such a small dataset. It is possible that VAE could be a better model, where there is a large amount of data collected. In addition to that, VAE provides flexibility in learning uncertainties in data. When much more data is fed into the model the distribution of features might change very much. This could only be learned using VAE. Unlike, PCA, VAE would be able to capture highly complex and non-linear relationship in the data.