# The Coherent Structure extraction using AI

Interim M.Tech. dissertation report

In

Thermal Engineering


Submitted By

Anirudh Pratap Singh

19METM714

*Under Supervision of*

Dr.Syed Fahad Anwer

Department of Mechanical Engineering

Aligarh Muslim University, Aligarh


2019-2020

# INDEX

**Contents** **Page No.**

# 1. Introduction

### Artificial Intelligence

Simulating Bio-chemical intelligence into another media such as electronic is know as Artificial Intelligence. Artificial Intelligence is generally attributed to machines or inanimate objects. The beginning of modern AI can be traced back to classical Philosophers. But, field of AI wasn't formally introduced till 1956. In 2016 just 60 years ahead Google's DeepMind programmer AlphaGo beat the 18-time World Go champion Lee Sedol, a game that has $10^{700}$ possible games in a 361-move game. This goes to show the level of critical thinking achieved by machine-based intelligence.

Today AI is at the forefront of the technology spreading to nearly all aspects of life from self-driving cars to solving problems to making intricate melodies. And with the advent of fast speed internet AI is getting the resources to accelerate to even higher standards. It is estimated that today humans generate 2.5 Quintillion bytes of data each day. With so much data in hand a well-trained AI can perform almost every task with near and sometimes more accuracy as achieved by humans.

With this vast amount of untapped data great strides can be made in understanding the secrets of the Universe and at the same time help humankind.

### AI and Science

Science is one such area where great advancements have been made due to AI in recent years. Fluid Mechanics being one part of this is where AI is coming up. This field generates intensive amount of data by running simulations which are to be properly interpreted and analyzed. One such area of application if Reduced Order Modelling (ROM) by extracting the important flow features. The reduced model greatly enhances the computation time by allowing computations

to be performed on reduced coordinates. A novel method at achieving this is Neural Networks which unlike linear eigen mode decomposition can learn the non-linear features making the ROM model even more efficient.

**Artificial Neural Network**

An artificial neural network is a system of hardware or software that is patterned after the working of neurons in the human brain and nervous system. Artificial neural networks are a variety of deep learning technology which comes under the broad domain of Artificial Intelligence.

Deep learning is a branch of Machine Learning which uses different types of neural networks. These algorithms are inspired by the way our brain functions and therefore many experts believe they are our best shot to moving towards real AI (Artificial Intelligence).

There are many kinds of Neural Network structure that make use one of the three kinds of learning mode being:

1. *Supervised Learning:* A ML task that maps the input to a known output. E.g., Learning to recognize hand written digits from a data set of labelled data set
2. *Semi-Supervised Learning:* Combination of labelled and un-labelled data which helps in increasing the learning accuracy
3. *Unsupervised Learning:* A ML task that draws inferences from the dataset. In some cases, the input acts as the output. E.g., Image compression makes use same input and output and tries to compress the image into a smaller size containing the maximum details.

One such structure that uses Unsupervised Learning is an Autoencoder. The structure of an Autoencoder is given in figure 1

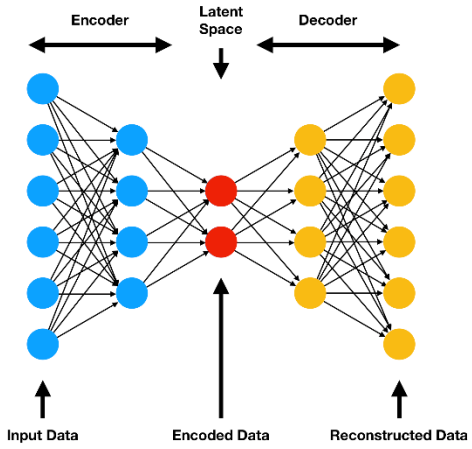


Fig 1: General structure of Autoencoder

## 2.Motivation for work

Fluid Mechanics has traditionally dealt with massive amounts of data from experiments, measurements on fluid, DNS. The analysis this huge amount of data has been a daunting task and has been mostly done with techniques like statistical analysis, heuristic algorithm. With the advent of AI we are provided with another means of analysis methodology. Also, AI is at the fore front of all technology in terms of growth and future scope. This inspired me to take up the work related to Dimensionality Reduction using Autoencoders.

## 3. Methodology

The main part of this thesis is based on mode decomposition of flow fields to extract the relevant coherent structures that govern a flow so that subsequently temporal and spatial analysis can be done as to the future state of flow or understanding the flow characteristics to model the flow in a enhanced manner. For this Autoencoders are used that follow unsupervised learning. The mathematics of Autoencoders is discussed as they are developed and tested to generated data throughout this thesis report. The aim is successfully extract modes of a turbulent flow field and thus.

Mode decomposition has been done for many year using methods such as Fourier mode decomposition, Principal Orthogonal Decompsition(POD) and DMD. While, all these methods are quite robust and easy, but they are limited in capturing the intricate features. E.g., POD gives only orthogonal modes. This drawback can be overcome by using the above Autoencoder method. For comparison POD has been extensively used to strike out points of interest.

## 4. Linear-Autoencoder

A Linear Autoencoder is an Autoencoder that has linear activation function across all its layers. With linear activation function the ANN is very similar to POD [1] when only 1 hidden layer is used. The dimension of the hidden layer directly corresponds to number of modes (as in POD).

To illustrate the similarity between a POD and Linear Autoencoder sample data is taken to verify. The structure of Linear Autoencoder (inherently POD) used is shown in figure (1)

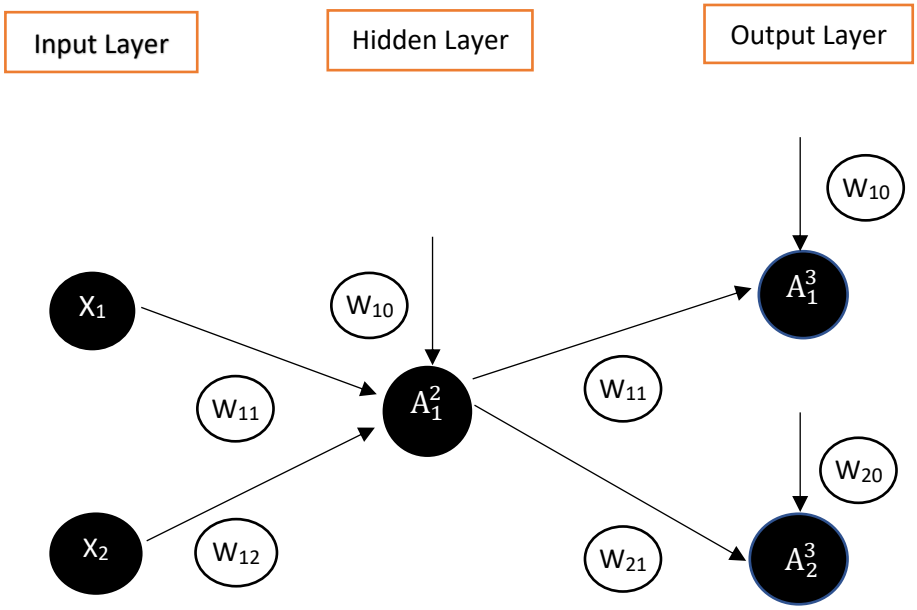


Fig 2: Structure of NN/AE

[1] For more details refer to the Appendix

The above structure can be mathematically surmised as

$$A^2 = f(W^2.X^T) \quad (1)$$

$$A^3 = f(W^3.(A^2)^T) \quad (2)$$

Where,

$f$ is a linear function i.e.the output is second and third layer is simply the weighted sum

$X$ is input vector with an additional bias unit

$A^2$ and $A^3$ are the output vectors, and

$W^2$ and $W^3$ are the weight vectors

The number of free parameters to be trained are 7 being the weight vector. Since, the Autoencoder employs unsupervised mode of learning thus, $A^3$ is same as the input vector $X$ .

The Autoencoder is designed using Keras API (ver 2.3.1) which utilises Tensorflow as the backend library. Method used to train the ANN/AE is Adam Optimizer[2]. The Autoencoder is trained for 5000 epochs[3] for various sets of data and the result is compared with the results obained from POD. Loss function to be optimised is MSE(mean squared error).

[2] ADAM-Adaptive Moment Estimation. It is an optimizer for the Gradient Descent or Stochastic GD

[3] Terminology used for iterations in Machine Learning

The summary for the Autoencoder designed is given below

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | (None, 2) | 0 |
| dense_1 (Dense) | (None, 1) | 3 |
| dense_2 (Dense) | (None, 2) | 4 |

Total params: 7

Trainable params: 7

Non-trainable params: 0

### 4.1 1-D Linear Data

The function is given by y=mx+c with some noise added to it. The independent variable ranges from (0,5) corrresponding to which the y values are obatined. The followind data is generated with m=2, c=0 and random noise added ranging in (-0.5,0.5)

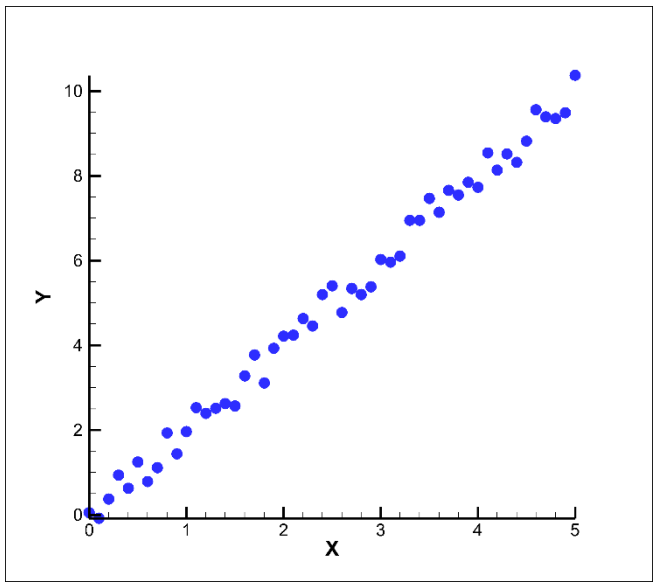


Fig 3: 1D Linear data with noise

The data is compressed and reconstructed with POD and AE using single mode and the results are compared [3].

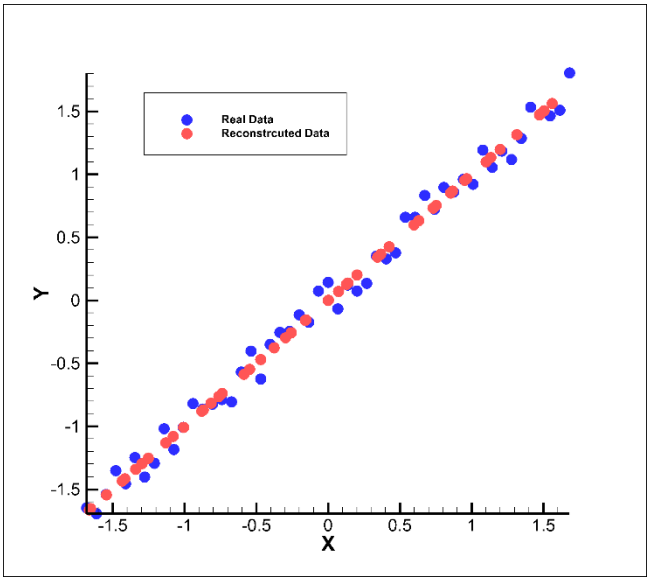


Fig 4: POD

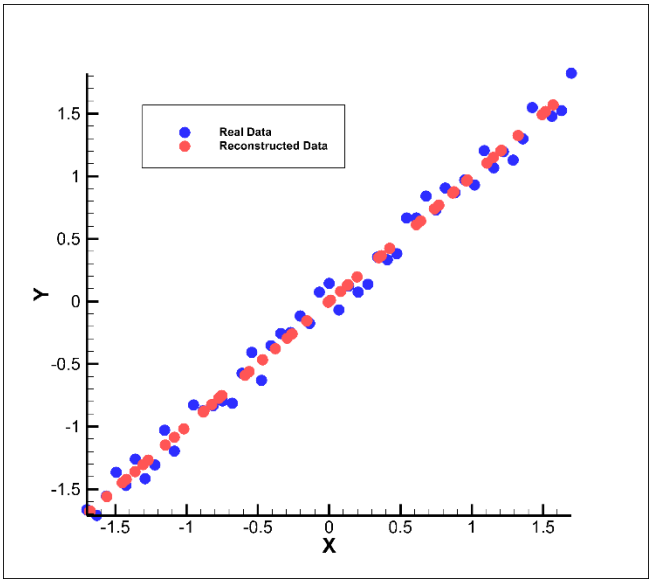


Fig 5: Autoencoder

The MSE for POD reconstruction is 2.2E-3 whereas for the Autoencoder its 2.3E-3. It can be seen that the results are qualitatively very similar and the difference in order of MSE is of the order of 1E-4 which proves that POD is a Linear Autoencoder. The Loss function i.e., MSE v/s number of epochs is given in figure 5. It can be seen that lowest MSE is achieved around 1400 epochs.

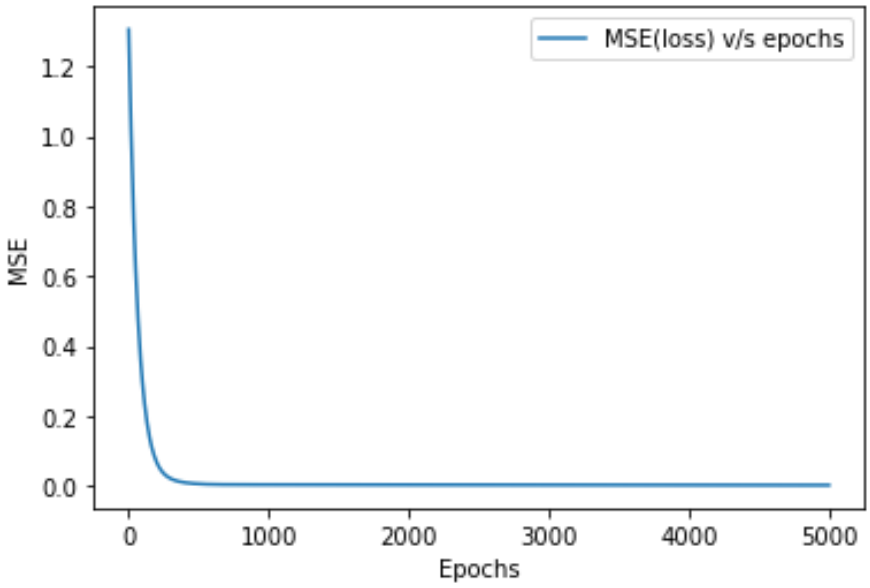


Fig 6: Loss/Cost function V/S Epochs

### 4.2 1-D Non-Linear Data

The function for Non-Linear data is given by $y = mx^4 + c$. Range of *x* is (0,5) and the value of parameters are *m=1, c=0* with noise.

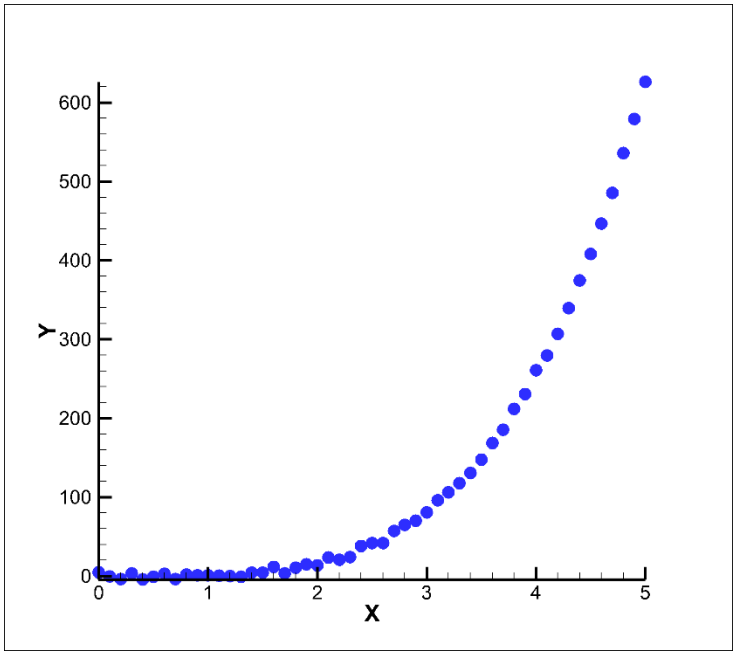


Fig 7: 1D Non-Linear data with noise

For this data same steps are taken as done for 1D Linear Data and the results are present below.

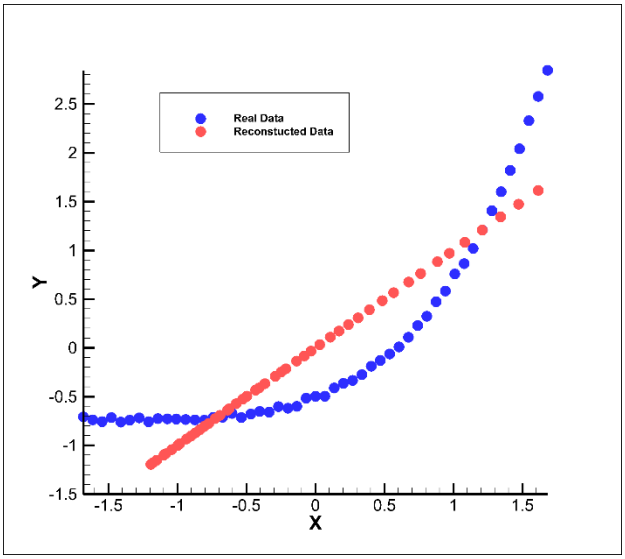


Fig 8: POD

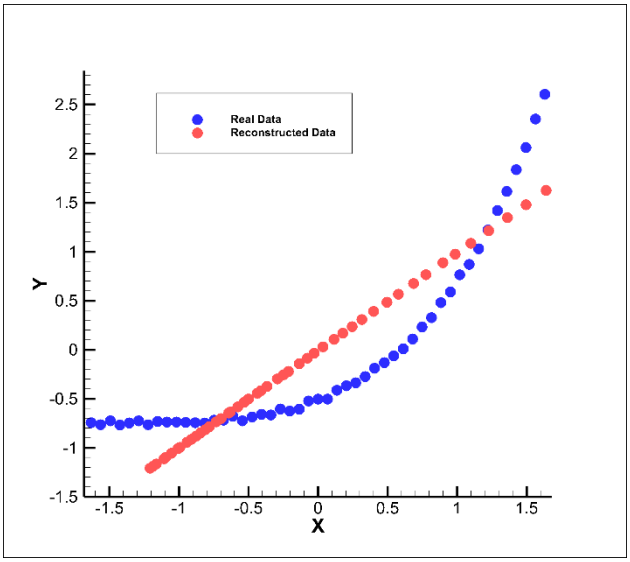


Fig 9: Autoencoder

The Loss Curve is given in figure 9

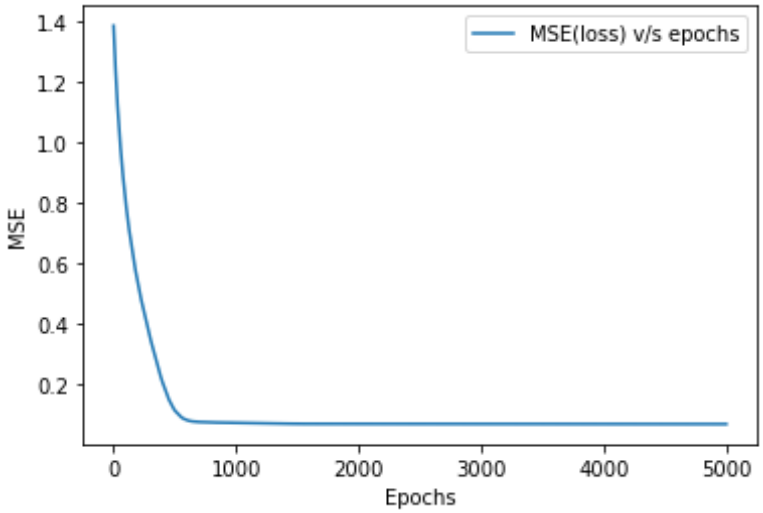


Fig 10: Loss/Cost function V/S Epochs

The MSE for POD reconstruction is 6.6E-2 while for Linear Autoencoder it is 6.8E-2. The lowest MSE for Linear Autoencoder is obtained at around 1800 epochs.

From above two examples it is quite evident that a Linear Autoencoder is POD in iterative method. The Linear Autoencoder method being an iterative method is very slow as compared to POD which is basically a one-step solver. Introducing Non-linearity to the Autoencoder makes it very powerful which is illustrated in the following section.

## 5. Non-Linear Autoencoder

For a Non-Linear Autoencoder we need three hidden layers to capture the non-linearity as shown in figure 11. The activation function taken across the ANN is Leaky ReLu[4].

Mathematically, the expression for such an Autoencoder is

$$A^2 = f(W^2.X^T) \tag{3}$$

$$A^3 = f(W^3.(A^2)^T) \tag{4}$$

$$A^4 = f(W^4.(A^3)^T) \tag{5}$$

$$A^5 = f(W^5.(A^4)^T) \tag{6}$$

Where,

$f$ is the Leaky ReLu function which is defined as

$$f(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases}$$

$A^2,\ A^3,\ A^4, A^5$ are the output vectors for each layer with biases added appropriately

$W^2,\ W^3,\ W^4,\ W^5$ are the weight vectors for each layer

$X$ is the input vector with biases

[4] For details for the activation functions refer to Appendix

The summary for the Autoencoder designed is given below

| Layer (type) | Output Shape | Param # |
|---|---|---|
| AE_input (InputLayer) | [(None, 2)] | 0 |
| encoder_model (Model) | (None, 1) | 9 |
| decoder_model (Model) | (None, 2) | 10 |

Total params: 19

Trainable params: 19

Non-trainable params: 0

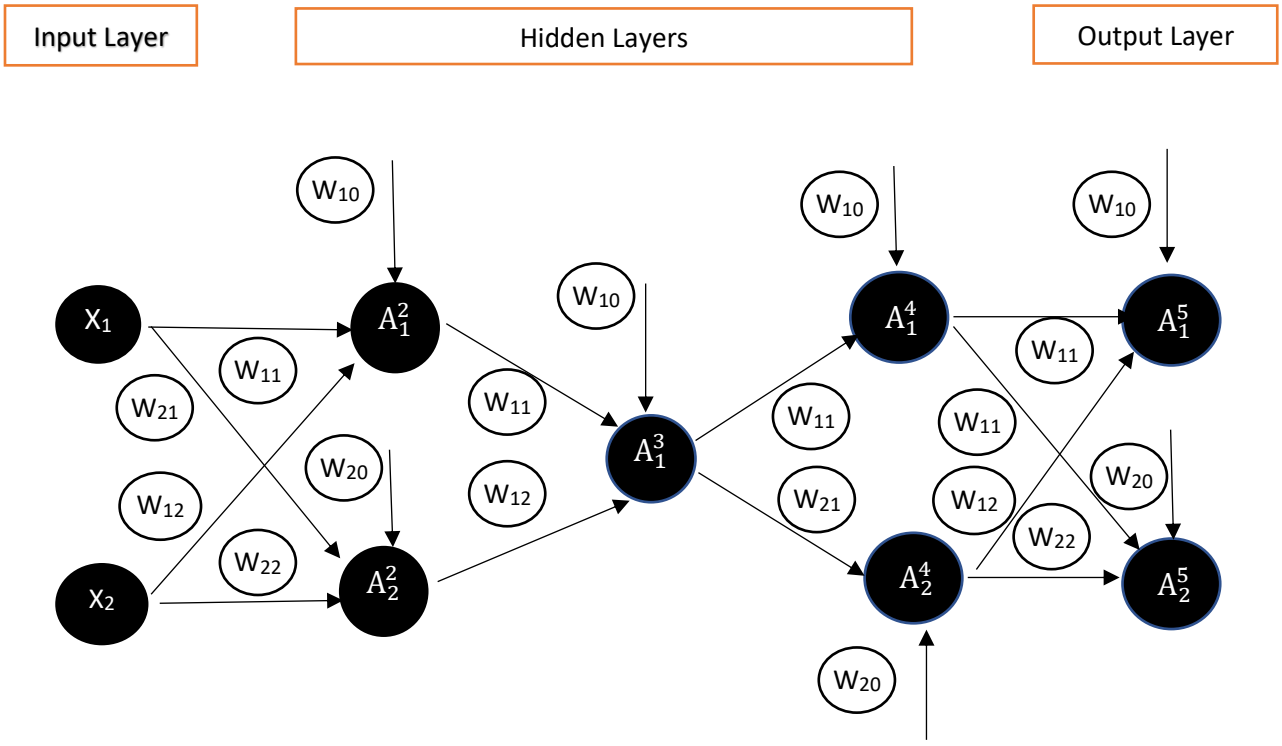


Fig 11: Structure for Non-Linear NN/AE

## 5.1 1-D Linear Data

For first case the same data is used as in figure 2 to observe whether a better is obtained or not.

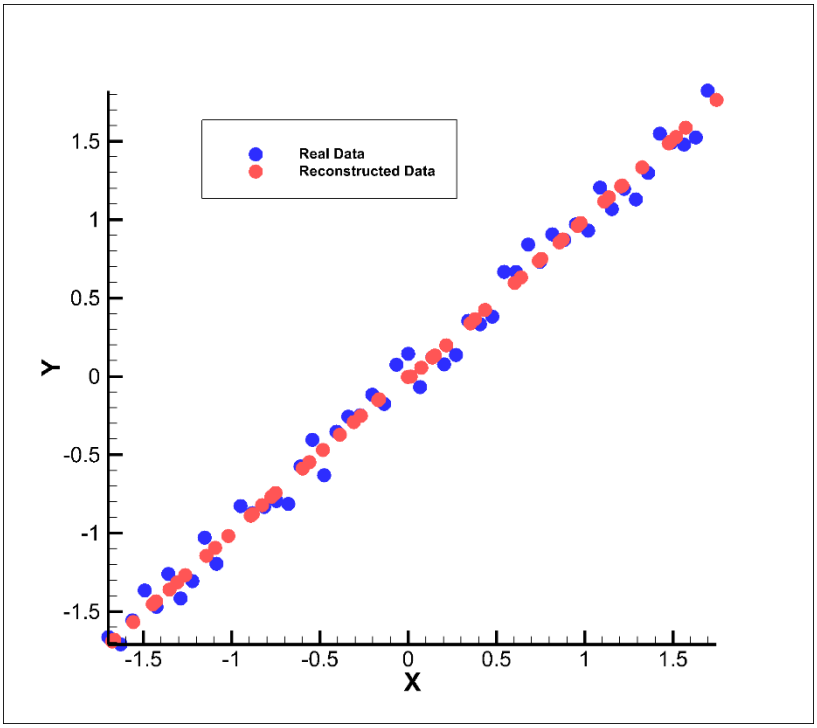


Fig 12: Non-Linear Autoencoder reconstruction efficiency

From figure 12 we see that introducing Non-linearity doesn't give any enhancement in result with minute decrease in MSE of the order of 1E-04. Although, this AE achieves the lowest MSE earlier evident from the loss curve given in figure 13.

We can conclude from this that even for a Non-Linear AE there is nothing much to learn!!

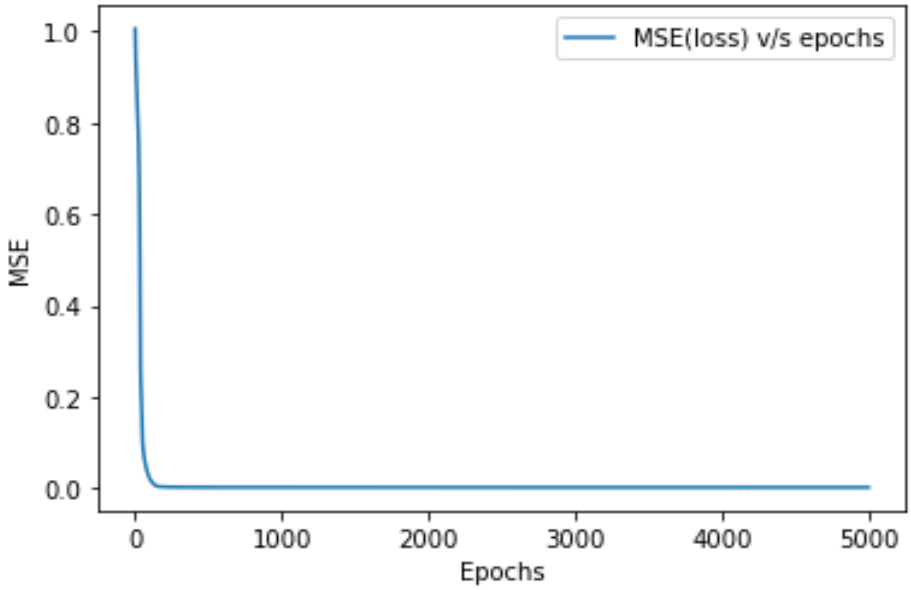


Fig 13: Loss Curve

## 5.2 1-D Non-Linear Data-1

Here also we will use the same data as given in figure 6. The results of reconstruction are given in subsequent figures.

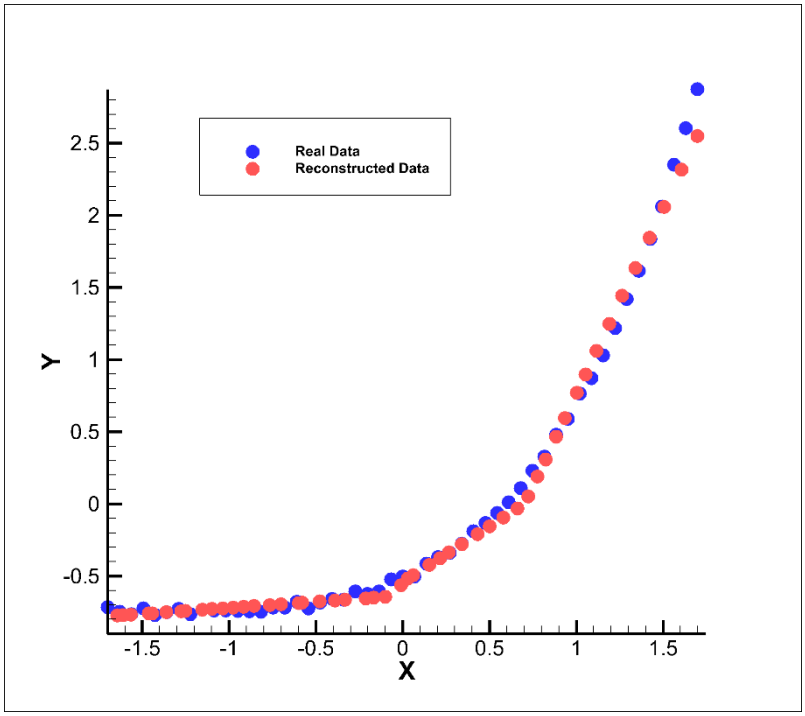


Fig 14: Reconstructed Data by Non-Linear AE

The loss curve for the Non-Linear Autoencoder training is given in figure 15. We can clearly see that introduction of Non-Linearity to the AE helps it to capture the non-linear characteristics. The minimum MSE we get is 8.9E-4 which is a great improvement over the linear methods using only on mode.

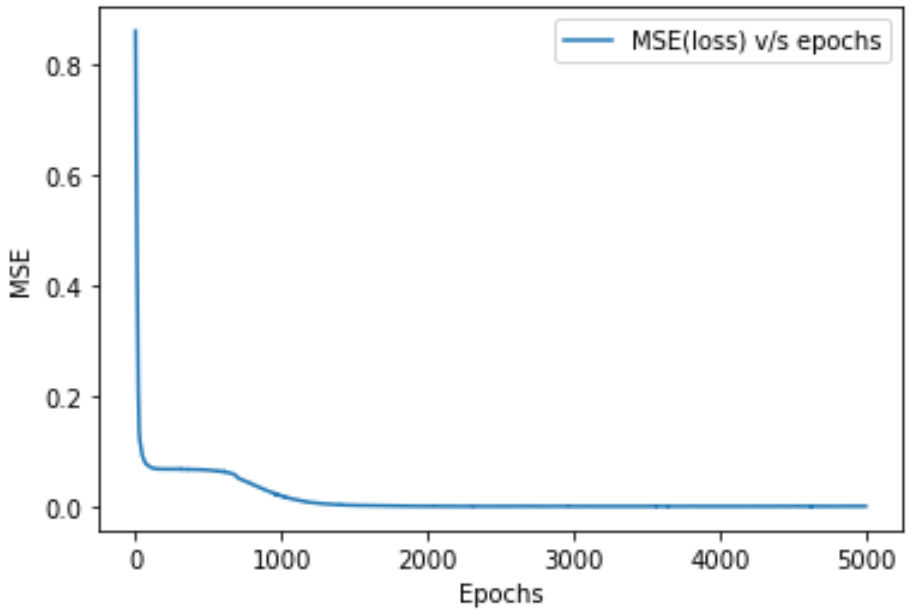


Fig 15: Loss Curve

## 5.3 1-D Non-Linear Data-2

For next case we consider the function to be $y = mx^8 + c$ where the parameter m=1, c=0 with noise. The result of reconstruction is given in figure 16. The MSE obtained for this is 2.2E-3. The difference between the MSE values for Linear Autoencoder and Non-Linear Autoencoder is tabulated (Table 1)

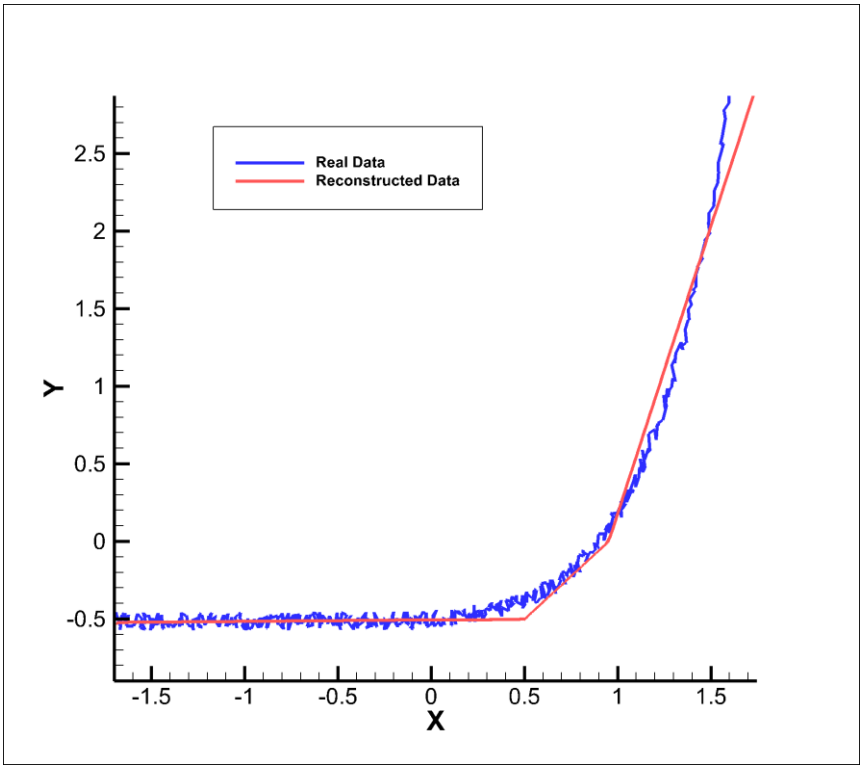


Fig 16: Reconstructed Data by Non-Linear AE

| Function | MSE(Linear Autoencoder) | MSE (Non-Linear Autoencoder) |
|---|---|---|
| $y = mx + c$ | 2.3E-3 | 2.2E-3 |
| $y = mx^4 + c$ | 6.8E-2 | 8.9E-4 |
| $y = mx^8 + c$ | 0.2056 | 2.2E-3 |

## 5.4 1-D Non-Linear Data-2 with more hidden layers

To increase accuracy of reconstruction increasing the hidden layers is one option. With this the number of trainable parameters increase to 31. Even with increase in hidden layers the MSE value remains same suggesting that a single mode cannot capture the intricate data characterstics.

The details for the new AE are given below

| Layer (type) | Output Shape | Param # |
|---|---|---|
| AE_input (InputLayer) | [(None, 2)] | 0 |
| encoder_model (Model) | (None, 1) | 15 |
| decoder_model (Model) | (None, 2) | 16 |

Total params: 31

Trainable params: 31

Non-trainable params: 0

## 5.5 1-D Non-Linear Data-3 with 7-Layer AE

For the next case we use normal distribution curve with noise.The range of independent varible is (0,5) in steps of 0.01. The Normal distribution has Avg of 2.5 and Standard Deviation of 1. The Autoencoder with 31 parameters is trained with a changed learning rate[5] of 0.001. The reconstructed data along with real data is shown in figure 17.

[5] Learning rate is analogous to step size in conventional computing. It was taken 0.01 for all cases before this.

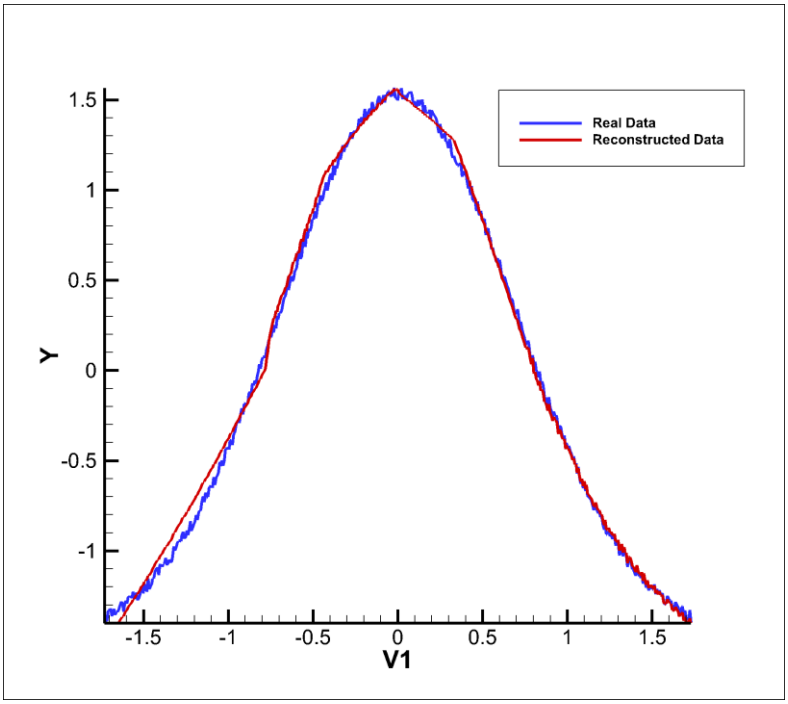


Fig 17: Reconstructed Data by Non-Linear AE with 5 hidden layers

The MSE value obtained for the above reconstruction is 1.9E-3. For same data trained using 3 hidden layer AE yields a MSE value of 0.2074

## 5.5 1-D Non-Linear Data-4 with 7-Layer AE

Final 1D data generated is a complex curve without noise. The result is shown in figure 18 where the MSE obtained is 2E-3.

The equation of curve is

$$0.1x(0.75x + \cos(2x)) \qquad (7)$$

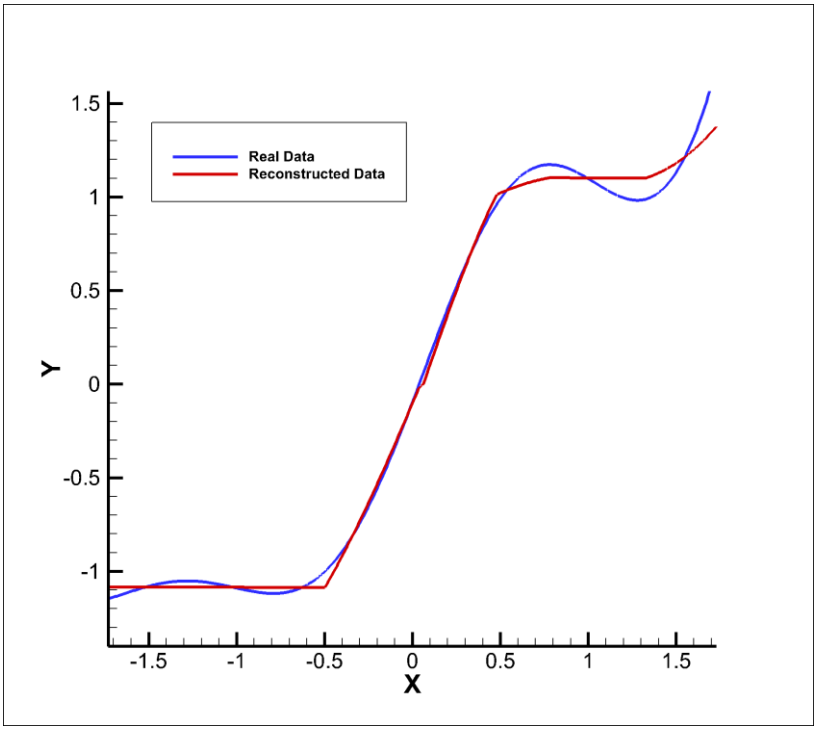


Fig 18: Reconstructed Data by Non-Linear AE with 5 hidden layers

From the figure 18 we can see that altough the MSE value is low the reconstructed data is not able to capture the more intricate feature vis-à-vis the humps and valleys.

It can be concluded that an ANN with 2 dimensional feature space is limited in its ability to reconstruct complex data.

## Bibiliography

[1] Brunton S.L. et al. 'Machine Learning for Fluid Mechanics', Annual reviews of fluid mechanics-2020, edition 52, pg 477-508

[2] Milano M., Koumoutsakos P. et al, Neural Network Modelling for Near Wall Turbulent Flow, Journal of Computational Physics 182, 1-26 (2002), doi: :10.1006/jcph.2002.7146

[3] Fukami K., Nakamura T., Fukagata K., Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fuid field data, Department of Mechanical Engineering, Keio University, Yokohama 223-8522, Japan, 2020

[4] Murata T., Fukami K., Fukagata K., Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, Journal of Fluid Mechanics (2020), vol. 882, A13, doi:10.1017/jfm.2019.822

[5] Baldi P., Hornik K., Neural Network and Principal Component Analysis: Learning from Examples without Local Minima, Neural Networks, Vol. 2, pp. 53-58, 1989

## APPENDIX