Hyper-Parameter Analysis of Deep Auto Encoder for Flow Prediction

Prakrit Tyagi, Pranav Bahl and B B Arora

May 21, 2021

Hyper-Parameter analysis of Deep Auto Encoder for Flow Prediction

Prakrit Tyagi[1], Pranav Bahl[1], and BB Arora[2]

Delhi Technological University, Delhi 110042, India

Abstract. Reducing the order of the system from high order dynamical space to a low dimensional subspace has been a challenging task and the models that achieve this are referred to Reduced Order Models (ROMs). Linear Projection methods have been used extensively in the past for building such ROMs, some common examples are POD,DMD etc. Though they have been successful in modelling several non-linear scenarios, their usage limits their applications pertaining to complex High order dynamical systems efficiently. In this study we aim to make use of advancement made in the field of Deep Learning to build a DL based ROM. We aim to probe the impact of hyper-parameters pertaining to flow prediction using deep autoencoders built with the help of artificial neural network. The parameters for our study that were used here were three different network sizes and two data sizes to compare the performance in flow prediction. Dataset for the Vorticity was generated using in-compressible URANS CFD solver icoFoam in an open-source CFD toolbox OpenFOAM. Von-Karman Vortex Street at Reynolds' number 100 around a bi-dimensional cylinder was simulated for our study.

Keywords: Deep Autoencoder · Von Karman Vortex Street · Hyper-pararmeters.

1 Introduction

Advances in Machine Learning techniques has made its usage reliable and economical and opened plethora of applications in fields of robotics , neuro science, economic and financial prediction, system dynamic modeling and prediction etc.. Autoencoder is one such technique which learns by itself to represent large data in reduced form. This is advantageous as using low dimensional data for analysis and calculation is efficient, computationally economical and reduces cost. Therefore this technique has found extensive usage in flow reconstruction, reduced-order modeling [1], prediction of fluid flow dynamical system [2] and flow prediction [3].

Computational fluid dynamics is the go to technique for simulating fluid flow but it is computationally expensive and if a users has inadequate resources, he/she will find it difficult to simulate and innovate. However, CFD has found its application across various engineering disciplines such as Aerospace industry, Heating and Ventilation applications, Flows corresponding to cardiovascular operation, Fluid machinery etc [4]. The need for such simulation often requires very high temporal and spatial resolution and hence the high order dynamics are simulated using Full-order-Models (FOMs) which are various numerical methods such as Finite Volume Method (FVM), Finite Element Method (FEM) etc. which are based on the governing equations (Partial Differential Equations). These models hence hinders its use in Multi-query [5, 6] scenarios which require rapid simulation result generation. The application of building large scale dynamical systems to simulate highly complex and non-linear flows seems to be failing in its use without taking a lot of computational burden. Hence there is an inherent need to represent higher order dynamics in reduced representations, which could be further used to evolve dynamics and when required can be reconstruct back to it's high-order form. These techniques are often referred to as Reduced-order-Models (ROMs) and many academicians have made their contributions in the past to model such techniques [7–10]. Building ROMs is not an easy task, since the models doesn't perform well in situations where the dynamical systems are complex since the parameters aren't robust enough. There are various projection based methods which have been introduced in the past which makes use of linear basis to form basis functions with help of snapshots generated using FOMs. These methods have found its acceptance among many researchers, such as Proper Orthogonal Decomposition(POD) [11–13], Dynamic mode Decomposition(DMD)[14, 15], Koopman Theory[16, 17] etc.

Here we make use of an Auto-encoder Neural Net to learn the spatial and temporal distribution of flow and give satisfactory results with low error when compared to CFD results. Our efforts are to make use of advancements that have been made in the Neural networks or Deep Learning methods to build an efficient ROM (Reduced order Model) that would be able to capture the High Order Dynamics effectively and represent it in reduced latent space capturing the essential features and thereby reconstructing high order snapshot for the next time step. This paper presents results for comparisons with different hyper-parameters. Three different network sizes are used and following this different data sizes were used for training and comparison was made among them. Below is the generalized mathematical foundation of our model which presents the w^{t+1} predicted vectorized state, the model and its parameters $\varphi(;\theta_\varphi)$ and the input to the model i.e. the current time step w_t. 'e' represents the error term generated by model corresponding to the actual true state, hence predicted state is represented as $\hat{w}^{t+1}$ and the True state as w^{t+1}.

$$w^{t+1} = \varphi(w_t; \theta_\varphi) + e \tag{1}$$

$$e = w^{t+1} - \hat{w}^{t+1} \tag{2}$$

2 Deep Learning Architecture : Auto-encoder

Artificial Neural Network is a computational model consisting of various neurons or functional nodes which helps in passing of information and manipulation of the data among themselves. The role of these neurons are often closely compared to the biological neuron present in our brains. A Deep Neural Network represents an ANN with several layers of neurons between the input and the output layer. The input features, which are often represented as "x^i" are fed into each neuron of input layer where different $i \in 0 \rightarrow N$ where "N" represents the number of features. The input data is multiplied to corresponding weights and an bias is added following which an activation function is applied to the whole output of the neuron to add the non-linearity to the function. The activation functions also allows or facilitates in deciding the amount of relevant information and in which quantity to be passed forward. Some common examples of activation functions are ReLU, sigmoid, Leaky-ReLU etc. The output of the first layer of neurons are then used as an input to the next layer of the neurons. A loss function is decided at the end of the network as a means to optimize the parameters of the model. The optimization of the network is facilitated with the help of back-propagation. There are several optimization algorithms used so as to optimize the weights of the models such as ADAM,RMS etc. The Autoencoder which has been used here is a type of Unsupervised ML algorithm that is composed three main subdivisions referred to as Encoder, Bottle-neck Layer and Decoder. It is used as a means to compress the data and then use it back to reconstruct the high order space. The compression of the data leads to representation of only essential features and thereby using it again to reconstruct to the original features. Unlike its counter-parts which rely only on linear projection methods, it's non-linearity gives it an added advantage.

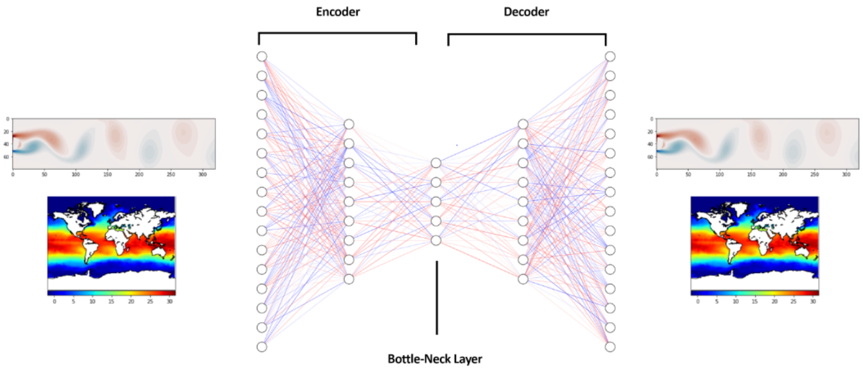


Fig. 1: Autoencoder Architecture

3 Proposed Approach

In this section, we will put forward the details of the framework for Deep Learning based model which can be used as a Reduced Order Model for predicting future time steps of the High Order Model with less computational efforts. The model is based on a Linear Auto-encoder approach which makes use of encoder and decoder architecture for reducing the number of computational nodes. The approach is used to learn the reduced latent subspace through a non-linear approach which helps in reducing the loss of information with comparison to its counterparts such as Proper Orthogonal Decomposition (POD), DMD, Petrov-Gaalerkin approach etc. which on other hand uses linear approach for the aforementioned function. Training of the model uses vectorized form of data $\tau = [\tau_1, \tau_2, \tau_3, ..., \tau_n] \in N_t$ which represents the data over t time steps. The state vector represented as $w \in N_w$ and predicted data as $\tau = [\tau_1+1, \tau_2+1, \tau_3+1, ...]$ The expression of the predicted state variable at time step "t+1" is as presented below.

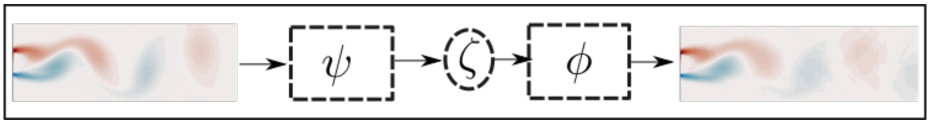


Fig. 2: Auto-encoder Architecture

$$\zeta = \psi(w_t; \theta_\psi) \tag{3}$$

$$\phi(\zeta) = \hat{w}^{t+1} \tag{4}$$

$$\hat{w}^{t+1} = \phi(\psi(w_t; \theta_\psi); \theta_\phi) \tag{5}$$

Where the $\phi(; \theta_\phi)$ represents the decoder network which projects the reduced latent space back to the original high order space with θ_ϕ being its network parameters to optimize. The $\psi(w_t; \theta_\psi)$ network represents the encoder which reduces the previous time step's high order data to a nonlinear reduced subspace with θ_ψ being the network parameters to optimize. We present in the paper following using this architecture that non-linear approach to project is an efficient method in comparison to it's counter-parts. The optimization of the network has been carried out using Mean Squared Error (MSE) approach for the optimization of the network parameters. The mathematical formulation has been presented below, wherein the "e" represents the difference between the predicted and the true state vector.

$$\alpha = min \frac{1}{N-1} \sum_{n=0}^{N-1} \|e\|_2^2 \tag{6}$$

$$\alpha = min \frac{1}{N-1} \sum_{n=0}^{N-1} \left\| w^{t+1} - \phi(\psi(w_t)) \right\|_2^2 \tag{7}$$

4 Methodology

The Auto-encoder approach as stated in this paper involves the use of a deep neural network containing neurons which in the first place are reduced as moving in the forward layers and then increased, so as to reconstruct the next time step. There are two split parts which are referred to as encoder and decoder wherein the encoder compresses the data, thereby performing the function of a Reduced-order-Model and then the decoder reconstructing the reduced state back to the high order output. During the training phase of the architecture, input to the encoder is the vectorized form of the flow-field at the time step ”t” and the label corresponding to the output of the decoder is the vectorized form of the flow field at time step ”t+1”. The Auto-encoder architecture mentioned in the paper has 5 layers of neural network with 3 hidden layers and 3rd layer being the bottle-neck layer. The activation function used here for the purpose of adding non-linearity to the ANN was chosen as ReLU. Linear activation is used in the end layer for predicting the vectorized form of the fluid flow prediction. ReLU has been used here since it is computationally less expensive. The mathematical formulation of ReLU function has been presented below.

$$F(x) = Max(0, x) \tag{8}$$

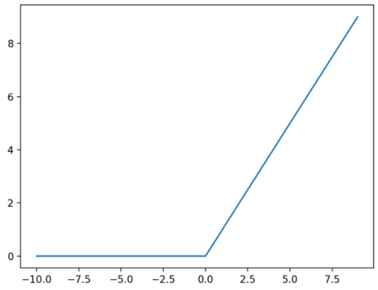
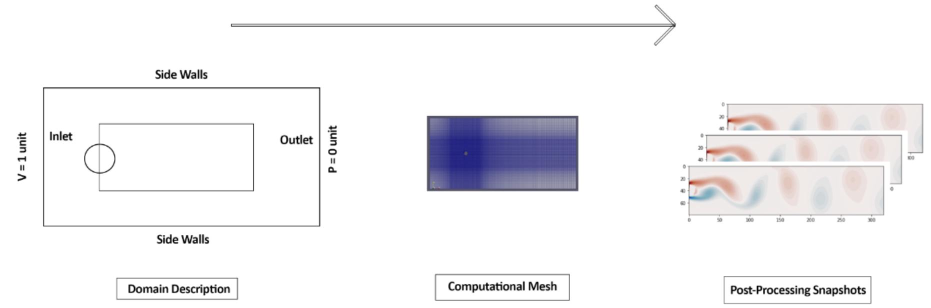


Fig. 3: ReLU Activation Function

The training of the network has been carried out with the help of ADAM Optimizer with a learning rate of 0.01. The parameters corresponding to encoder and decoder are represented as θ_ψ and θ_ϕ which are optimized with the help of ADAM algorithm. For regularization of the network we use L^2 regularization to eradicate the possibility of over-fitting. Weight decay of 0.5 have been used here so as to regularize the network.

5 2D Von Karman Vortex Street : OpenFOAM

Here we consider a Two-dimensional flow past a cylinder which is shaped of a circular configuration with the Reynolds's number of the flow as 100. The problem is world-wide known phenomenon which is characterized by periodic vortex formation behind the circular body which is laminar in nature. The computational domain of the model has been shown below wherein the structured mesh of the domain was generated using the BlockMesh utility of OpenFOAM. OpenFOAM which has been used here as a means to generate synthetic data for the the training of the model is an Open-source CFD ToolBox which is widely used for performing complex fluid flow simulations across the academic community. The cylinder has a diameter of 1 unit along with its distance corresponding to the centre from the inlet is 8 units and from the outlet is 25 units. Boundary conditions of the domain corresponds to the unit velocity from inlet and pressure of 0 at the outlet. A no-slip BC has been initiated for the circular body. The simulation carried out here is of Unsteady RANS simulation using OpenFOAM. The solver used here is icoFoam which is an incompressible Navier-stokes solver. For the sake of training of only the relevant features the cut out of the computational domain is taken , whose cut-out stretches from [-2 2] in Y-direction to [0 16] in X-direction. The data has been linearly interpolated in this domain for 80 x 320 points. The objective here is to predict and generate the vorticity-field at the following time-step from the current time-step.



Domain Description

Computational Mesh

Post-Processing Snapshots

Fig. 4: Numerical Experiment - SetUp

6 Results

6.1 Effect of Network Size

In this section we present the effect of network size. Three network architecture were chosen with hidden layers as given network:1 $[200, 100.200]$, network:2 $[100, 50.100]$, network:3 $[50, 25.50]$ .The data size contained 400 time steps and each architecture was trained on it. It can be seen in fig:5 that as the network size was increased the training error decreased sharply in case of network one and in case of network three it decrease sharply in initial epochs but the accuracy later decreased due to underfitting.

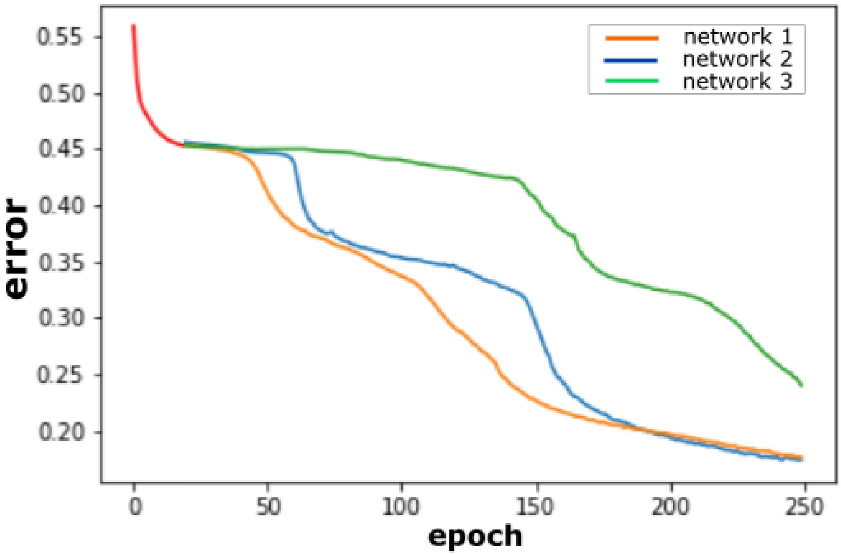
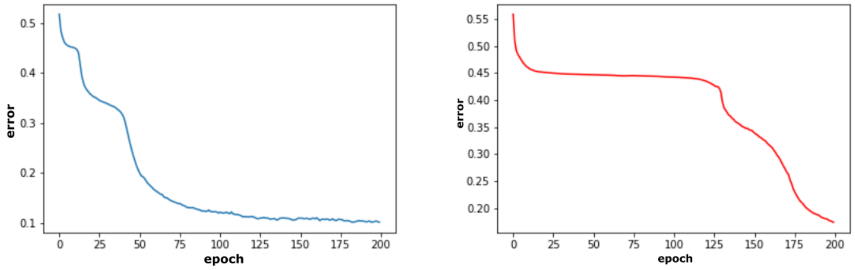


Fig. 5: Effect of Network Size

6.2 Effect of data Size

In this section we present the effect of data size. Two data sizes were used with size one having 921 time steps and size two having 400 timestamps for the training set. It can be seen in fig:6 that with training with larger data size the error decreased rapidly but with smaller data size the error plateaued because it got stuck locally. Below in fig:7 and fig:8 is spatial representation of predicted output compared with true values in the case data size two.



(a)

(b)

Fig. 6: Effect of data size: (a) plot for data size one (b) plot for data size two

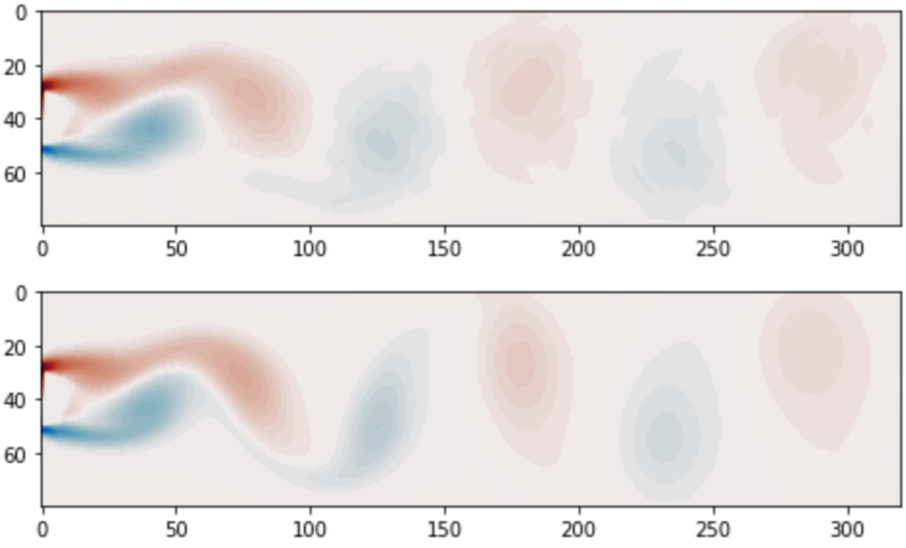


Fig. 7: Comparison of True and predicted at time step 50

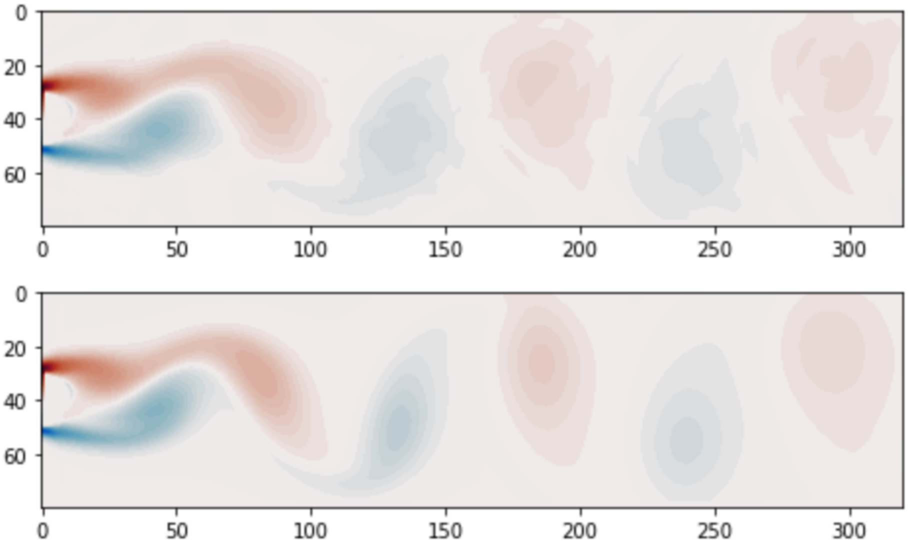


Fig. 8: Comparison of True and predicted at time step 100

7 Conclusion

This paper presents the Hyper-parameters comparison between different Autoencoder architectures and there comparison pertaining to each other. This paper presents a Deep learning based Reduced order Model (ROM) which is expected to reduce the losses in the information and extract only relevant data in the bottleneck layer so as to reconstruct the high order dynamics efficiently from the reduced state. We have made comparison between network architectures wherein the following conclusion was made, i.e. the training error sharply decreased as we increased the network size, which has its underlying reason for added nonlinearity in the network but also leads to a relatively more computationally expensive model to its counter-part architecture. Following this we also make comparison between increased size of dataset and lower amount of dataset which were generated with the help of different time-steps. It was seen and concluded that a larger dataset helped in converging to the solution much rapidly and easily in comparison to the shorter dataset.

References

1. Wang, Xin and Takaki, Shinji and Yamagishi, Junichi and King, Simon and Tokuda, Keiichi: Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. Physics of Fluids **33**(3), 037106 (2021)
2. Agostini, Lionel: Exploration and prediction of fluid dynamical systems using autoencoder technology. Physics of Fluids **32**(6), 067103 (2020)

3. Erichson, N Benjamin and Muehlebach, Michael and Mahoney, Michael W: Physics-informed autoencoders for Lyapunov-stable fluid flow prediction. arXiv preprint arXiv:1905.10866 (2019)
4. Pragati, Kaushal and Sharma, HK and others:Concept of computational fluid dynamics (CFD) and its applications in food processing equipment design. Journal of Food Processing and Technology **3**(1), (2012)
5. Kookjin Lee and Kevin T. Carlberg: Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. Journal of Computational Physics **404**, 108973 (2020)
6. Stefania Fresca and Luca Dede and Andrea Manzoni: A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. arXiv, 3 (2020)
7. Zhaojun Bai: Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. Applied Numerical Mathematics **43**(1), 9-44 (2002)
8. Akhtar, Imran and Borggaard, Jeff and Burns, John A. and Imtiaz, Haroon and Zietsman, Lizette: Using functional gains for effective sensor location in flow control. Journal of Fluid Mechanics **781**, 622–656 (2015)
9. David J. Lucia and Philip S. Beran and Walter A. Silva : Reduced-order modeling: new approaches for computational physics. Progress in Aerospace Sciences **40**(1), 51-117 (2004)
10. F. Fang and C.C. Pain and I.M. Navon and G.J. Gorman and M.D. Piggott and P.A. Allison and P.E. Farrell and A.J.H. Goddard: A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows. Ocean Modelling **28**(1), 127-136 (2009)
11. Berkooz, G and Holmes, P and Lumley, J L: The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows. Annual Review of Fluid Mechanics **25**(1), 539-575 (1993)
12. Holmes, Philip and Lumley, John L. and Berkooz, Gahl and Rowley, Clarence W: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge Monographs on Mechanics (2012)
13. Aubry, Nadine and Holmes, Philip and LUMLEY, JOHNL and Stone, Emily and others: The dynamics of coherent structures in the wall region of a turbulent boundary layer. Journal of Fluid Mechanics **192**(1), 115–173 (1988)
14. Proctor, Joshua L and Brunton, Steven L and Kutz, J Nathan: Dynamic mode decomposition with control. SIAM Journal on Applied Dynamical Systems **15**(1), 142–161 (2016)
15. Gilles Tissot and Laurent Cordier and Nicolas Benard and Bernd R. Noack: Model reduction using Dynamic Mode Decomposition. Comptes Rendus Mécanique **342**(6), 410-416 (2014)
16. Mezić, Igor: Analysis of fluid flows via spectral properties of the Koopman operator. Annual Review of Fluid Mechanics **45**, 9357–378 (2013)
17. Brunton, Steven L and Brunton, Bingni W and Proctor, Joshua L and Kutz, J Nathan: Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. PloS one **11**(2), e0150171 (2016)