AMATH 590 - Paper Reading and Review, Fluid Flow Reconstruction

Deepak Venkatanarasimhan(1940141)

April 23, 2021

Abstract

The reviewed paper(`https://arxiv.org/abs/1902.07358`) addresses the reconstruction quality of a high-dimensional fluid flow from limited measurements. The paper uses Proper Orthogonal Decomposition(POD) and a shallow decoder(SD) to do the same and compares results. Reconstruction methodology is tested on three datasets - i) Canonical fluid flow around a cylinder, ii) Mean sea-surface temperature, iii) velocity field of a turbulent isotropic flow. The paper studies the impact of the number of sensors and sensor placement on reconstruction quality. It also discusses other important hyperparameters, such as the number of layers, the size of each layer for the shallow decoder, and tuning parameters for regularized POD across datasets. In general, it is found that shallow decoders can reconstruct the flow field better from sparse measurements, as opposed to standard modal approximation techniques. This is because Neural networks(NN) can extract a more accurate basis than POD, given their strengths with nonlinear operator estimation.

1 Paper Overview

The ability to reconstruct a fluid flow from a limited number of sensors is useful across areas of study where fluids are involved, for instance, climate modelling. A fluid flow with a low-dimensional structure can be used to train a model that reconstructs a high-dimensional representation of the fluid from limited measurements. The paper looks at three datasets and two reconstruction methods.

For datasets, we have:

1. Dataset 1: Canonical dataset of Fluid flow past a cylinder at Reynolds number 100, characterized by laminar periodic vortex shedding.
2. Dataset 2: Weekly sea surface temperatures (SST) for the last 26 years. The mean SST flow field has a periodic structure, and is non-stationary.
3. Dataset 3: Velocity field of a turbulent isotropic flow, which is non-periodic in time and highly non-stationary.

As is evident from the above description, the complexity of the reconstruction task increases from datasets (1) to (3). This implies that modelling tasks on datasets (2),(3) require more sensors than the one on dataset (1). Also, the shallow decoder needed to reconstruct the fluid field for Datasets (2) and (3) requires more neurons in the hidden layers because of the higher dimensionality of the input(number of sensors). Reconstruction quality(measured using normalized mean square error) is easier to achieve on dataset (1) because of its high periodicity and the presence of smooth large-scale patterns [8].

The paper compares reconstruction quality for two methods: i)POD-based reconstruction and ii) Shallow decoder. It is observed that POD methods require more sensors as compared to a shallow decoder(SD) for a given dataset and that the reconstruction quality is highly sensitive to sensor location (in the case of POD)[8]. The shallow decoder(SD) reconstructs the original high-dimensional state vector better. Basis learned via an SD exhibits elements resembling physically consistent quantities, in contrast with POD-based modal approximation methods that enforce orthogonality [8]. However, NN-based SD suffers from limitations when one attempts out-of-sample testing.

To train the Shallow Decoder, we start with an input layer that accepts the low-dimensional sensor vector (s), traverses two hidden layers, and outputs the high-dimensional fluid field (x) (Please see Equation 8). Once trained, the SD must reconstruct the high-dimensional fluid field for a given vector of sensor measurements. On the POD-end, Eigen flow fields can be used with the appropriate rank and coefficients to estimate the high-dimensional flow field. Coefficients can be computed with or without regularization. I discuss the math behind both the POD and SD methods in Appendix A. As noted before, POD methods require more sensors to provide accurate estimations for high-dimensional fields. Also, reconstruction quality is highly sensitive to sensor location [8]. In the case of SD, random sensor locations suffice [8].

In terms of reproducing results, I focus on the canonical case. Please see Appendix C for reconstruction of flow around the cylinder from 5 sensors using both POD and SD. Detailed steps followed for reconstruction are specified in Appendix B. The paper also discusses results from the mean sea temperature and the turbulent flow dataset. Driven by differences in the properties of periodicity and stationarity, one requires more sensors to reconstruct the fluid state for datasets (2) and (3). However, SD still outperforms POD for a given number of sensors for all datasets.

Another important learning point is that Neural networks do very well in mapping an input to output within the span of the training data[3]. This is because NNs are fundamentally interpolative [3]. Typically, out-of-sample performance is not as good. The paper being discussed shows that extrapolation has poor performance for the turbulent flow case[8]. A follow-up paper by Erichson et al.[1] outlines how one can use physics-informed prior knowledge to improve extrapolation performance.

In summary, key learnings from the paper are:

1. NNs perform nonlinear basis extraction better than POD and hence can reconstruct fluids better from sparse measurements, in this case. NNs are good at function/ mapping search. However, the optimization to solve weights becomes hard for nonlinear activation functions.

2. Choice of a shallow network is driven by ease of training with a smaller training set, as compared to other architectures such as CNN, which do not suit the problem at hand.

3. Number of sensors required increases with the complexity of the fluid system we attempt to reconstruct. Sensor placement is also an area of study.

2 Research Context

The paper reviewed (`https://arxiv.org/abs/1902.07358`) explores the use of shallow decoders to generate low-rank feature spaces as an alternative to POD. This work is encouraging for the vast body of flow measurement and control work, where sparse sensing is a requirement. This is because once the fluid-structure is captured using low-rank feature spaces, one can reconstruct a fluid from sparse measurements. The inherent compressibility of a system to a low-dimensional representation enables sensor-based flow reconstruction[7]. Sensors are critical to multiple areas of work for enabling measurement and flow control. Also, given sensor cost, reducing the number of sensors through principled design may be critically enabling[7]. An abundance of offline fluid-flow data in laboratory settings is at odds with the availability of sparse, noisy data from sensors in practice[4]. An abundance of offline data has made such work possible, while practical considerations and constraints on sensor data have made it interesting.

For extracting a basis from nonlinear systems, POD/Galerkin models can work well but often require careful tuning to give satisfactory results[5]. As a sidebar, a precursor to such model-free flow field estimation is seen in the stochastic estimation work done by Adrian et al., as described in [4]. The Shallow Decoder work in this paper is a continuation of the POD work that attempts to extract a basis from dynamics that is essentially nonlinear[5]. The SD attempt is informed by the properties of Neural Networks, i.e., i) their ability to handle multiscale phenomenon without significant tuning, and ii) ability to handle invariances due to translation, rotation, or scaling [6]. However, as argued in [6], POD-based methods have the edge in interpretability.

In the paper being reviewed, the authors do a good job of choosing a hierarchical structure that mimics a POD-like basis extraction and guarantees low running times for training. They also discuss remedies for overfitting that are common with NNs. Hence, it advances the state of knowledge in basis-estimation and opens up new areas for inquiry, such as extrapolation and other network structures.

The authors expect this paper to seed more research. One example is using the new feature space in Reduced Order Models for forecasting predictions [8]. Another related area of work is using physics knowledge of the system to design decoders that perform better in extrapolation tasks [1]. Super-resolution involves the inference of a high-resolution image from low-resolution measurements, leveraging the statistical structure of high-resolution training data[3]. Super-resolution work such as in [2] can also be considered a related area of work.

References

[1] Bernjamin Erichson et al. *Physics-informed Autoencoders for Lyapunov-stable Fluid Flow Prediction.* 2019.

[2] Kunihiko Taira et al. *Super-resolution reconstruction of turbulent flows with machine learning.* 2018.

[3] Steve Brunton et al. *Machine Learning for Fluid Mechanics.* 2020.

[4] Steve Brunton et al. *Robust flow field reconstruction from limited measurements via sparse representation.* 2018.

[5] Scott T.M. Dawson Clarence W. Rowley. *Model Reduction for Flow Analysis and Control.* 2017.

[6] Nathan Kutz. *Deep learning in fluid dynamics.* 2017.

[7] Nathan Kutz Steve Brunton et al. *Data-Driven Sparse Sensor Placement for Reconstruction.* 2017.

[8] Nathan Kutz Steve Brunton et al. *Shallow Neural Networks for Fluid Flow Reconstruction with Limited Sensors.* 2019.

Appendix A Summary of Theoretical Discussion

POD is used to reconstruct the high-dimensional state of a fluid, provided there exists a basis in which the high-dimensional state vector has a sparse representation. Coefficients associated with POD modes to reconstruct a specific flow are obtained by solving an optimization, as shown below in Equation 5. We start with SVD on the mean-centered training set, as shown in Equation 1. Columns of $X = \{x_1 \cdots x_n\}$ store the high-dimensional representation of the flow field.

$$X = U\Sigma V^T \tag{1}$$

We truncate left-singular vectors U at rank k to get the required modes. Please see Equation 2.

$$\Phi := U_k \tag{2}$$

Using Φ, we can reconstruct a high-dimensional state based on the idea that a field x can be expressed in terms of a rank-k approximation. Please see Equation 3.

$$x = \widehat{x} = \Phi\mathbf{v} \tag{3}$$

To approximate coefficients $\mathbf{v}$, we use sensor measurements s. We start with Equation 4, which defines a nonlinear operator $H : \mathbb{R}^m \to \mathbb{R}^p$, as shown below:

$$s = Hx \tag{4}$$

We use s to learn the coefficient $\mathbf{v}$ using the optimization shown below in Equation 5.

$$\mathbf{v} \in \arg\min_{\widetilde{v}} \|s - H(\Phi\widetilde{\mathbf{v}})\|^2 \tag{5}$$

The same equation can also be setup with an L2 regularization term for $\mathbf{v}$, which is presented in the paper as POD-plus. Please see Equation 6 below.

$$\mathbf{v} \in \arg\min_{\widetilde{v}} \left\| s - H(\Phi\widetilde{\mathbf{v}}) \right\|^2 + \alpha \left\| \widetilde{\mathbf{v}} \right\|^2 \tag{6}$$

From Equation 5, we can estimate $\mathbf{v}$ as specified in Equation 7. The coefficients can then be plugged back into Equation 3 to get the high-dimensional field. Here, the superscript + indicates the pseudoinverse.

$$v = (H\Phi)^{+} s \tag{7}$$

The above has been implemented with and without regularization, in the code given at link (`https://github.com/erichson/ShallowDecoder`).

For the shallow decoder(SD) model, we use a standard feed-forward network that is setup using two layers, as shown in Equation 8.

$$\widehat{x} = \sigma(W_3(R(W_2 R(W_1 s)))) \tag{8}$$

Per the paper, the first layer creates an approximation basis, and the second layer constructs the high dimensional fluid state $\widehat{x}$ from this basis [8]. The input layer is of the size of s, and the output layer is of the size of x. The last hidden layer is of the size of the rank of the approximation k, of the dictionary Φ. In contrast, the size of the first hidden layer is driven by the size of the input layer s and the number of nonlinear combinations required to compute the coefficients $\mathbf{v}$. This sort of hierarchical abstraction is said to be effective in feature learning.

The paper uses narrow rather than wide layers. Prescribing the size of the layer restricts the dimension of the space in which the estimation lies, and it effectively regularizes the problem, filtering-out most of the noise which is not living in a low-dimensional space [8]. For the layers, the paper chooses fully-connected as opposed to convolution layers. This is because, convolution layers require more training samples, and in the current example our sensor measurements have no spatial ordering. Finally, the first two layers use a RELU activation function R, and the final layer which combines the modes uses a linear activation function σ. Batch normalization of the activation functions is used as a regularization technique.

In Appendix B, I summarize the detailed steps to conduct the two analyses, i) POD, ii) Shallow Decoder, on the canonical fluid flow around a cylinder dataset.

Appendix B Review of Algorithm/ Steps

In this section, I describe implementation details for the use of POD and SD to estimate the high-dimensional state x from s. I specifically discuss the canonical flow around a cylinder dataset that is made available at `https://github.com/erichson/ShallowDecoder`, as a .npy file. I discuss the below two processes separately:

1. Use POD to reconstruct the high-dimensional state from s (Algorithm 1)
2. Train and use a Shallow Decoder to reconstruct the high-dimensional state from s (Algorithm 2)

The data preparation steps for both methods are pretty similar. To compare performance across methods, we are to setup a reconstruction error metric.

In Summary, a couple of observations to make before we head to the next section:

1. In the POD case, we are required to choose one hyperparameter α (ridge tuning parameter). The paper discusses searching for this to minimize the reconstruction error, in Appendix A and B.
2. In the SD case, we are required to choose hyperparameters for the optimizer - number of epochs and batch size. I accept the given hyperparameters.
3. In the SD case, one also chooses width of each layer. The philosophy for choosing the same is explained in the theory section around Equation 8.

Algorithm 1: POD-based reconstruction

Read the fluid flow matrix from the given input file
Measure the size of the matrix - dimensions of each sample, and number of samples
Split the samples into training and test. The paper keeps 2/3rd of the samples in training.
Demean the training and test sets, and store the mean
Randomly choose sensor locations on the cylinder. Paper chooses 5 locations
Setup the sensor measurements data from the training data
Compute the linear coefficients $\mathbf{v}$ using Equation 6
Compute the reconstructed state using $\mathbf{v}$ and sensor measurements s, using Equation 3
One could also work with regularized POD, where the norm of $\mathbf{v}$ is minimized, as in Equation 6
Plot the reconstructed flows along with the original flows

Algorithm 2: Shallow Decoder reconstruction

Read the fluid flow matrix from the given input file
Measure the size of the matrix - dimensions of each sample, and number of samples
Split the samples into training and test. The paper keeps 2/3rd of the samples in training.
Demean the training and test sets, and store the mean
Randomly choose sensor locations on the cylinder. Paper chooses 5 locations
Setup the sensor measurements data from the training data
Setup the training data in Tensorflow using samples of the sensor vector s and full-state vector x
Setup the structure of the shallow decoder as explained in the theory section around Equation 8
Setup the required hyperparameters and choose an optimizer
Train the shallow decoder
Plot the in-sample and out-of-sample reconstructions

4. For the mean sea temperature and isotropic flow use-cases, the paper uses more sensors, and SD structures with greater width. This is discussed in Appendix C of the paper.

I choose minimum required sections of the code from `https://github.com/erichson/ShallowDecoder`, and setup my own notebook at Colab Link. With this I am able to plot the original dataset, and reconstructions using POD and SD from the sensor state.

Appendix C Reproducible Results

For the canonical flow around the cylinder case, I was able to follow the steps described in Appendices A, B, to plot the original as well as reconstructed data. Please see Figure 1 for the original flow field, and Figure 2, 3 for reconstructions from Shallow decoder(SD) and POD. In the results show below, we train the SD to be able to reconstruct from a sensor state of length 5. The sensor locations are randomly chosen on the cylinder, as shown in Figure 1. The high-dimensional output state length is $384 * 199 = 76416$. The ability to be able to do such a thing implies that the field has an inherent structure in sparse dimensions that the SD is able to capture. Similarly, the POD-based method is using an orthogonal-basis Φ, to reconstruct the high-dimensional state x from s. It does this by learning $\mathbf{v}$ using Equation 5. The quality of the reconstruction is limited by the ability of the new basis to represent the high-dimensional state, as well as the coefficients $\mathbf{v}$.

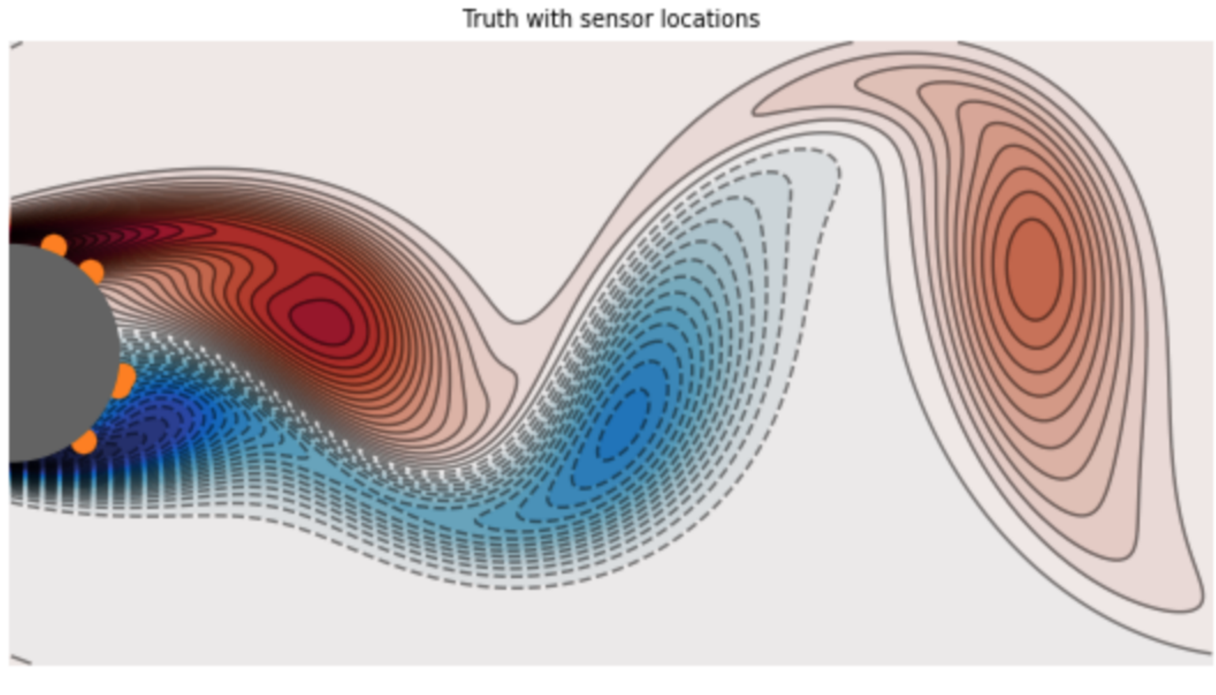

Figure 1: True High-Dimensional Image of Fluid Flow past a Stationary Cylinder, with sensors

In the paper, it is observed that the reconstruction error of the SD case is lower than POD. POD performance can be improved by regularizing the coefficient parameter $\mathbf{v}$. However, SD still does a better job. This is because it is more efficient in learning and approximating a basis Φ in the first hidden layer of the network, through the use of RELU. RELU does not constrain the modes to be linear and orthogonal, as is enforced by POD. So one can say that Neural networks using nonlinear activation functions better approximate the nonlinear mapping H from $s \rightarrow x$ (Please see Equation 4).

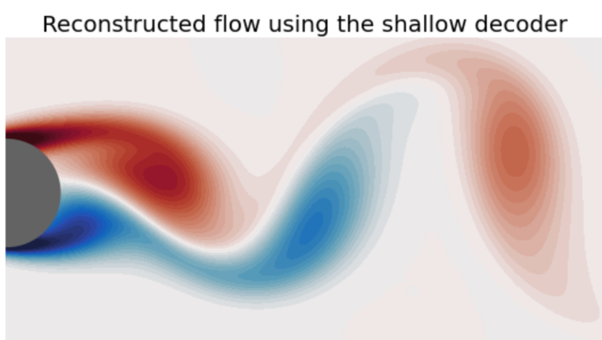

Figure 2: Shallow Decoder Reconstruction from Sensor State

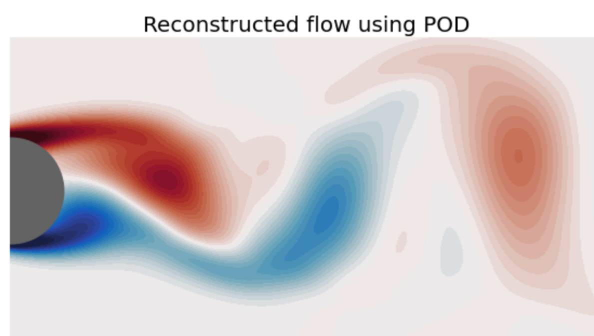

Figure 3: POD-based Reconstruction from Sensor State