Data Driven Approximations Of PDEs

On Robustness of Reduced Order Mappings between Function Spaces Against Noise

Pablo O. Lacombe[1]

Supervisor(s): Dr. David Tax[1]**, Mahdi Naderibeni**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
January 28, 2024

Name of the student: Pablo Olivier Lacombe
Final project course: CSE3000 Research Project
Thesis committee: Dr. David Tax, Mahdi Naderibeni, Nergis Tömen

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

This paper presents a comprehensive exploration of a novel method combining Principal Component Analysis (PCA) and Neural Networks (NN) to efficiently solve Partial Differential Equations (PDEs), a fundamental challenge in modeling a wide range of real-world phenomena. Our research extends the work of Bhattacharya et al. by focusing on PCA for effective dimensionality reduction and utilizing NN for mapping in the reduced dimension. This approach addresses the significant computational challenges and inaccuracies often encountered with classical numerical techniques in solving PDEs.

We specifically investigate the still-water equation, employing our PCA-NN method to learn a reduced order mapping of PDE solutions and evaluate its robustness in diverse noisy environments. Our findings reveal a notable relationship between noise intensity and error, indicating a linear trend for Gaussian and Salt and Pepper noise, and an exponential trend for Uniform noise. Furthermore, this study uncovers a critical weakness of the model in predicting points with a high rate of change.

Overall, our research significantly contributes to understanding the practical applicability and limitations of PCA-NN methods in real-world, noisy settings, offering valuable insights for future applications in this domain.

1 Introduction

Partial Differential Equations (PDEs) play a foundational role in the mathematical modeling of various natural and engineering systems. Equations like the heat equation, Navier-Stokes, and Black-Scholes are famous examples of PDEs, each having extensive use cases in chemistry, physics, and finance, respectively. For example, shallow-water equations (SWE) which describe "the conservation of mass and momentum in a fluid [...] to determine circulation patterns and the maximum and minimum tides at the interior of the region"[1]. These equations are instrumental in our understanding of dynamic changes within different environments. However, this understanding comes with certain caveats. Traditional numerical methods for solving such PDEs exist, but they often face limitations in terms of computational efficiency and struggle with handling high-dimensional data [2]. The advent of data processing, machine learning, and artificial intelligence has led to novel perspectives in addressing this challenge.

This paper particularly focuses on solving PDEs using a synergy between principal component analysis (PCA) and neural networks (NN), as an attempt to provide efficient and accurate solutions in a timely fashion. Foundational work by Bhattacharya et al. 2021 [3] explored a data-driven approach to approximating PDEs using the PCA-NN approach. Our research is anchored in replicating and extending this work. This step is crucial in potentially gaining additional insights into their methodology and findings.

The purpose behind this study is two-fold. First, there is the reproduction of the work from Bhattacharya et al. and second the exploration of the research question: **"How robust are the PCA-neural network based approaches against noise in the data?"**. From weather data to wind tunnels, sensors deal with noisy environments all the time.To this end, we simulate various types of noisy environments to assess the strengths and limitations of this technique depending on the type and intensity of the noise. These findings are significant, as noise received limited attention in Bhattacharya et al's research. By understanding this techniques' performance in noisy conditions, we can better gauge its applicability and limitations in practice.

2 Methodology

To answer the research question it is essential to first discuss the key methods used. There are core data techniques used for processing and predicting data, as well as, noise functions used for 'contaminating' our data in order to understand noise-resilience.

2.1 PCA-NN approach

At the core of this study we find a a synergy and heavy reliance on Principal Component Analysis (PCA) and Neural Networks (NNs). These are the core data processing techniques used in increasing computational efficiency and predictive modelling.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique used to emphasize variation and bring out strong patterns in a dataset. It has tremendous applications from simplifying data exploration and visualisation to compression and dimensionality reduction. PCA works by identifying the directions along which the variation in the data is maximum. These components, in decreasing order, essentially capture the most important parts of the data where more components correlate to a higher resemblance to the original data space.

In our study, we apply PCA to reduce the rank of the dataset while preserving as much information as possible. For example, it could allow us to reduce a 128×128 vector space into say 150 components, while still retaining 99% of the input data. While it is not a PDE, Figure 1 aims to help visualize how PCA works.

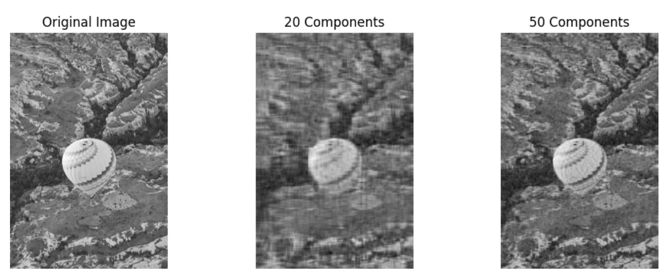

Figure 1: Visualisation of PCA on an image of a hot air balloon

Note that the original image requires 3729×5593 pixels. By using PCA, we can reduce the image to fewer components while still retaining much of the image's variance. In fact, in Figure 1 our reduction to 20 and 50 components allows us to retain around 65.16% and 79.49% of the variance, respectively, while compressing the size by approximately 99.46% and 98.66%, respectively. Figure 1 visualizes the surprising efficiency PCA has in data retention.

It is important to note that the example given in Figure 1 is a simplification of PCA, and that its application on PDE data is a little more complicated, though the underlying idea is very similar. We are applying PCA to reduce the complexity of our data and, hopefully, make solving PDEs simpler and quicker through this data-driven technique.

Neural Networks (NNs)

Neural Networks are a cornerstone of machine learning, due to their incredible ability to learn patterns or classify data. NNs consists of layers of interconnected nodes or 'neurons'. Each neuron in a layer is connected to every neuron in the next layer, serving the purpose of applying a function on passed data and feeding it to the next set of neurons. When data is fed into the network, it undergoes transformations through these layers, enabling the network to learn complex patterns and relationships in the data.

There is extensive theory on NNs and their applications [4], which unfortunately can not covered in full in this paper. What is most important to note is that our study utilizes NNs for predictive modeling. Given that we are trying to 'learn' how a vector space changes from one state to another, we are using NNs to find this mapping.

PCA-NN Method

Central to Bhattacharya et al.'s paper is the combining of PCA and NN techniques. It involves a two-phase approach: dimensionality reduction through PCA and function mapping through a NN. PCA is key given the potentially highly dimensional PDE data, allowing reduction to a manageable and efficient level. We can apply PCA in a way that allows us to retain a percentage of the variance (say 99%) to keep. Applying PCA to the start of the mapping (input) and end of the mapping (output) separately, allows us to reduce the dimensionality of each of these spaces.

Subsequently, we define a function mapping between the PCA reduced spaces through a neural network. The input and output to this NN are the principal components of the input and output of PCA processed data, respectively. By feeding this neural network our PCA processed data we "teach" it this very mapping.

In the context of PDEs, you could of think of Bhattacharya's technique as an aim to map one function space onto another. For example, the heights of waves at initial interaction and their heights after one second.

Using the shallow-water equation as an example, we can look at Figure 2 to try and fully grasp the PCA-NN technique. The task is to find a direct mapping between initial function space, in this example time at t_0 , and the final function space, at t_1. Applying the PCA method brings the high resolution input and output functions into lower representations with dimension of n and m. Following this, there is a mapping between these to low dimension data is learned with a neural network. This means that an input function at t_0 is

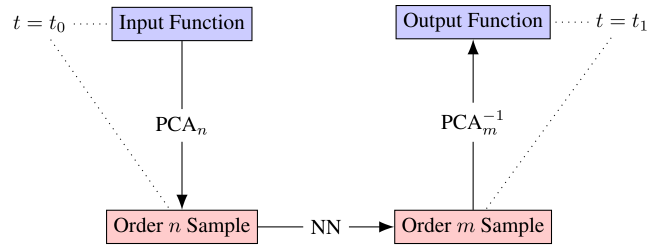

Figure 2: Flow Diagram of the PCA-NN Prediction Process for a Time Dependant PDE

reduced to dimension n through PCA, where it is mapped to a dimension m reduced function at t_1, and finally brought back to its original non-reduced state. It is important to note that Figure 2 is showing the prediction process for time-based PDEs. Certain PDEs, like Darcy Flow, which do not have a time component can still be learned using the PCA-NN technique [3].

2.2 Noise and Intensity

This paper will predominantly focus on the injection of 3 different types of noise; Gaussian noise, uniform noise, and salt and pepper noise. Each noise type brings a unique set of characteristics and challenges to the PCA-NN methodology, thereby providing a comprehensive understanding of how different noise models impact the process. For each of these noise types, we define a γ which generally scales the intensity or frequency of the noise. There are many different ways of implementing noise and noise-intensity and that for that reason we will be explaining our methodology on noise injection. Additionally we will briefly touch upon known strategies employed to mitigate the impact of noise.

Gaussian Noise

Gaussian noise, a common byproduct of sensor noise, is generally applied when specific information about the noise in data is unknown [5]. It is generated from a normal distribution with a mean of zero and a variance scaled by the square of the intensity factor γ. γ can be any non-negative value, where 0 indicates no noise and values like $\gamma = 1$ indicate a variance directly proportional to the variance of the data. Equation 1 highlights the process of inserting Gaussian noise into the PDE data.

$$\text{Noisy Data} = \text{Original Data} + \mathcal{N}(0, \gamma^2 \cdot \sigma^2) \tag{1}$$

Where $\mathcal{N}(0, \sigma^2)$ represents a normal distribution taking parameters with mean 0 and variance σ^2.

Figure 3 visualizes the effects of Gaussian noise injection. We can see that, for the most part, the general shape of the data is preserved but appears cluttered. This noise seems to vary in intensity but is stable, containing only a few noise spikes.

Uniform Noise

Uniform noise introduces an even distribution of random fluctuations within a specified range. We again use a scaling factor (γ) which dictates the boundaries of the uniform distribution. γ scales proportionally to half the data range meaning

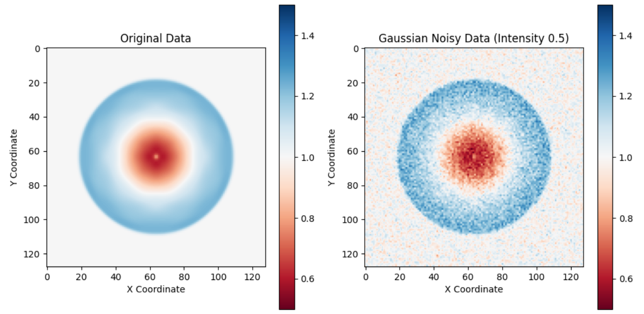

Figure 3: Impact of Injecting Gaussian Noise at 0.5 Intensity on a Shallow-Water Sample

$\gamma = 0$ will add no noise and $\gamma = 1$ will uniformly inject noise between the maximum and minimum points in the data. The addition of uniform noise to PDE data can be represented using Equation 2.

$$\text{Noisy Data} = \text{Original Data} + \mathcal{U}(-\gamma \cdot \frac{r}{2}, \gamma \cdot \frac{r}{2}) \tag{2}$$

In Equation 2, $\mathcal{U}(a, b)$ represents a Uniform distribution taking parameters with lower-bound a and upper bound b. We use r as the range of the data and $\gamma \cdot \frac{r}{2}$ is used so that a value is picked within the range of $[-\gamma, \gamma]$, which is proportional to the range of the data.

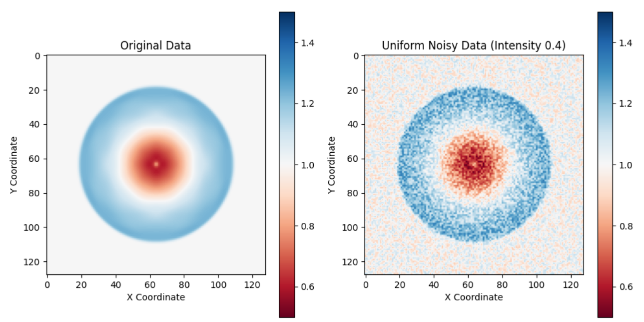

Figure 4: Impact of Injecting Uniform Noise at 0.4 Intensity on a Shallow-Water Sample

While subtle, the uniform noise observed in Figure 4 tells a different story than the Gaussian noise results seen in Figure 3. While having a lower intensity measure, 0.4 rather than 0.5, the noise tends to have more variance, often presenting higher noise values and rendering the original sample less clear.

Salt and Pepper Noise

Salt and Pepper noise introduces sharp, sparse disturbances in the data by randomly setting certain points to either the maximum or minimum value in the data. This noise type is characterized by its abrupt and unpredictable nature, where points may read as the maximum (salt) or minimum (pepper) value of the data. To apply Salt and Pepper noise to data we assign a maximum or minimum noise with probability γ to each datapoint, meaning at $\gamma = 0$ the original data is not altered and that at probability $\gamma = 1$ all data points become noisy rendering the sample unrecognizable. Equation 3 demonstrates this process, the effects of which can be seen in Figure 5

$$\text{Noisy Data} = \begin{cases} \max(\text{Data}) & \text{with probability } \frac{\gamma}{2} \text{ (salt)} \\ \min(\text{Data}) & \text{with probability } \frac{\gamma}{2} \text{ (pepper)} \\ \text{Original Data} & \text{with probability } 1 - \gamma \end{cases} \tag{3}$$

Where $\max(X_s)$ and $\min(X_s)$ represent the maximum and minimum value of a data-set X_s, respectively.

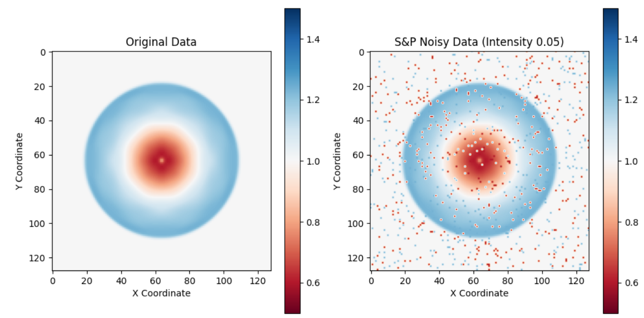

Figure 5: Impact of injecting Salt and Pepper Noise at 0.05 intensity on a Shallow-Water sample

The salt and pepper noise observed in Figure 5 shows a stark contrast from the behaviors seen in Gaussian and Uniform noise. The unique high salt points and low pepper points will present a different type of challenge for the PDE solving PCA-NNs, as opposed to the Gaussian and uniform noise.

2.3 Gaussian Noise PCA Optimal Hard Threshold

Prior research has been done on mitigating the effects of Gaussian noise [6], with claims that there is a "optimal hard threshold τ for singular value truncation under the assumption that a matrix has a low-rank structure contaminated with Gaussian white noise" [6].

This implies that, for Gaussian noise, we can know exactly how many PCA components are needed for the lowest Mean Squared Error (MSE). This approach argues that there is an optimal trade-off between meaningful data and noise where higher values would represent noise rather than meaningful data and lower values would fail to capture meaningful data. The calculation for the optimal threshold τ can be seen in Equation 4 and Equation 5 where we assume our data to be a 2D matrix $X \in \mathbb{R}^{n \times m}$ and define $\beta = \frac{n}{m}$ when $m \ll n$ and γ is our Gaussian noise variance intensity factor.

$$\tau = \lambda(\beta)\sqrt{n} \cdot \gamma \tag{4}$$

where

$$\lambda(\beta) = \left(2(\beta + 1) + \frac{8\beta}{(\beta + 1) + (\beta^2 + 14\beta + 1)^{\frac{1}{2}}}\right)^{\frac{1}{2}} \tag{5}$$

However, it is important to note that this optimal hard threshold only reduces the MSE when comparing a dataset to a PCA-reduced dataset. This method does not make claims on the ideal number of components for the function mapping, this is something that needs to be explored.

3 Experimental Setup

In attempting to reproduce part of Bhattacharya et al.'s work and evaluating their method's resilience against noise, certain implementation decisions were made. This was either because certain decisions were not explicitly expressed in the paper, or because a more promising alternative was found. This section will highlight our own process in creating PDE approximating PCA-NN, as well as our unique approach to tackling noisy data.

3.1 Data Collection

The data used in training and evaluating the PCA-NN comes from PDE Bench [7], which was created for the very purpose of machine learning techniques. While data generation techniques do exist, these were not deemed necessary for the aims of the study. This paper almost exclusively uses data from the 2 Dimensional shallow-water Equation, though the PCA-NN methodology is not exclusive to this PDEs.

PDE Bench claims the array dimensions are organized

> according to the convention $[b, t, x1, ..., xd, v]$, where b is the batch size (i.e. number of samples), t is the time dimension, $x1, ..., xd$ are the spatial dimensions, and v is the number of channels (i.e. number of variables of interest)[7]

Due to this data shape, and the intention to find a direct mapping over time, it is recommended to reduce the data down to just two time points, t_0 and t_1, where mapping needs to be learned.

3.2 Shallow Water Equation (SWE)

The shallow-water equation (SWE) is the focal point of this research. SWE are a set of hyperbolic partial differential equations (PDEs) that describe the flow below a pressure surface in a fluid (often, but not exclusively, water). The SWE are typically used to model geophysical flows such as ocean currents, weather fronts, and tsunamis, as they efficiently describe the flow dynamics of fluids with a free surface [1]. In the context of this research, SWE data offers a rich field to apply PCA-NN, due to the complex dynamics and non-linear characteristics of fluid motion. These dynamics make the SWE a challenging and relevant test case for evaluating the resilience of the PCA-NN method against noise.

To aid in understanding SWE, please refer to Figure 6 which demonstrates an example of SWE. We notice that it is mapped in a 2 dimensional space through the X and Y coordinates and has a values that ranges from 0 to 2. The color bar the height of of the liquid in the SWE equation, with blue colors indicating above average liquid heights and red colors indicating below average liquid heights. In this example, the SWE input function space essentially models placing elevated water on perfectly still water, while the output looks at the physical consequences of the added water. You can think of this as a raindrop in a pond, and SWE attempting to model the tides following the drop.

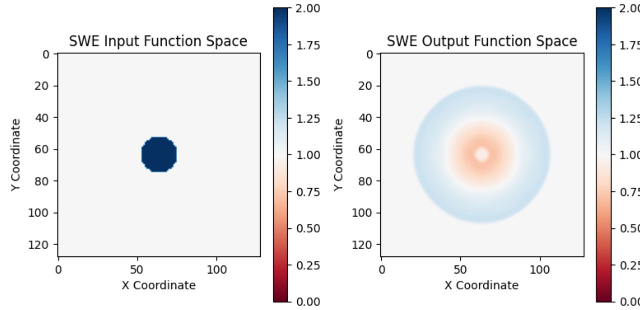

Figure 6: Example of Input and Output Function Spaces for SWE

It is important to note however, that the aims of this research is not based on legitimizing its specific use for SWE, but rather aims to explore the legitimacy the PCA-NN technique as a means of solving PDEs, particularly in noisy environments.

3.3 PCA-NN Implementation

In creating our own PCA-NN we largely followed Bhattacharya et al.'s implementation [3] alongside Kovachki et al.'s implementation [8], which are both papers that focused on reduced order function mappings. The following experimental setup defines the decisions made in the process.

Generally it is highly recommended to apply PCA independently to the input and output of the function mapping. This is because, depending on the PDE, one of these reduced-order function spaces may have a much higher variance than the other. This depends on the specific use-case but in order to capture a higher data variance more components are needed.

We are training our network on the rank-reduced function spaces. While the network is training on this reduced data, it should still be evaluated against the actual output. Doing inverse PCA on the prediction is an essential step when testing and comparing different reductions.

We defined our network according to convention and prior implementations [3] [8]. We used a 7 layer fully connected neural network initialized with following neurons per layer: n, 500, 1000, 2000, 1000, 500, m. Where n is the number of input components and m is the number of output components. The network used a SELU activation function on all layers and an Adam optimizer. After some hyper-parameter tuning through a grid search on SWE, we use 8 input and output components, a learning rate of 0.0006, batch size of 32, step size of 5, and gamma of 0.1. Figure 7 shows the model's rapid ability to learn the data given the above configuration, converging on a test MSE of approximately 0.00228.

While this was our implementation, ideal network architecture and decisions may vary across PDEs and this is by no means a "Gold Standard". Optimisations and Different networks exist depending on chosen PDE, number of components, network architecture, and hyper-parameters. The focus of this research is not on finding the best possible way of doing PCA-NN, but rather more exploratory in nature.

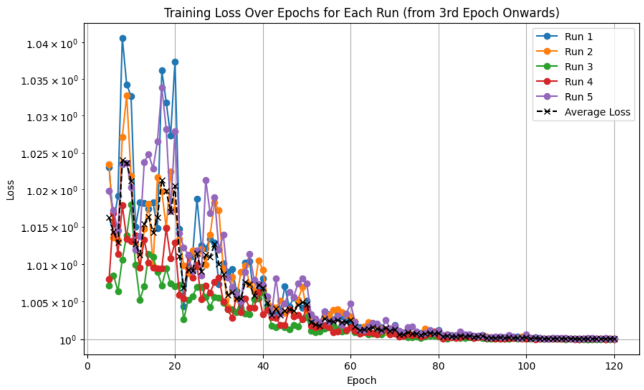

Figure 7: Loss Over 120 Epochs of 5 Runs using found Hyper-Parameters

4 Responsible Research

Conducting research responsibly necessitates a rigorous reflection on its ethical implications, integrity, and reproducibility. This section elucidates the potential ethical issues identified in our research and the measures adopted to mitigate their impact.

4.1 Data and Machine Learning

The use of predictive modeling and artificial intelligence raises concerns about data bias and the potential harm caused by model predictions [9]. Our research exclusively utilized physics-based data, specifically the SWD and Burger's equation, ensuring no involvement of personal or identifying data. The nature of our data limits its application to physics simulations, thereby reducing ethical risks typically associated with more sensitive datasets.

4.2 Ethical Concerns

Recognizing the broad applications of technologies developed in science and engineering is important. Partial Differential Equations (PDEs), such as the diffusion equations or Navier-Stokes, find uses in diverse fields, including sensitive areas like nuclear science and weapons manufacturing.

While acknowledging these concerns, we believes the direct implications of our work on such areas are minimal. Our research is exploratory in nature, focusing on the efficacy and resilience of PCA-NN methodologies in handling noisy data. We contend that applications in contentious fields are more likely to rely on traditional PDE solving methods rather than our proposed PCA-NN approach due to its approximate nature.

Nevertheless, it is important to note that the optimization of the PCA-NN algorithm for PDEs has broader implications beyond engineering. This methodology could be effectively applied in various domains where prediction and approximation are valuable. However, it is imperative to recognize that the application of this method in different contexts, particularly with varying datasets, could lead to unethical outcomes or biased predictions. Future practitioners and researchers utilizing this PCA-NN approach must remain vigilant regarding its ethical implications and potential biases inherent in their specific use-case scenarios.

4.3 Reproducibility and Research Integrity

Ensuring the reproducibility and integrity of our research has been a cornerstone of our methodology. To this end, we have included the data used, algorithms developed, and the steps involved in our experiments. This transparency allows other researchers to replicate our study, verify our findings, and build upon our work, thereby contributing to the scientific community's collective knowledge. We aim to provide a comprehensive and honest account of our findings, thereby upholding the principles of research integrity.

In conclusion, our commitment to responsible research extends beyond mere compliance with ethical standards. It involves a proactive approach to anticipate potential ethical dilemmas, mitigate risks, and ensure that our research contributes positively to the scientific community and society at large.

5 Findings

The aim of our research was to reproduce Bhattacharya et al's research and to answer the question how robust PCA-NN based approaches are against noise in the data. Here we will highlight significant results as part of our research process.

5.1 PCA-NN Methodology

This section will highlight some of the key takeaways and insights found when applying the PCA-NN technique to the 2D SWE.

Selecting Number of Components

We found that it is hard to make a legitimate claim as to the ideal number of components in lowering MSE. One would think that, in more components, and therefore better capturing the data, would lead to a more accurate predictions.

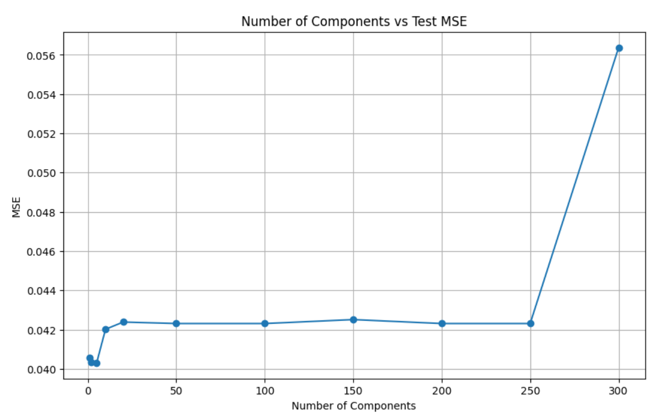

Figure 8: Plot of number of components against test MSE

Figure 8 plots the number of components against the Test MSE and shows an unlikely relationship between the two, with higher number of components having limited impact on the accuracy of the model. While this may seem surprising, this actually aligns with what Bhattacharya et al. found in their paper on the linear elliptic PDE:

> we observe a slight amount of overfitting when more training samples are used and the reduced dimension is sufficiently large [...] While this suggests that simpler neural networks might perform better on this problem, we do not carry out such experiments as our goal is simply to show that building in *a priori* information about the problem (here linearity) can be beneficial [3]

Their goal aligns with that of this study; exploring and developing an understanding of this method. While we intuitively believe that this trade-off could exist, There are many variables that could effect this efficiency such as network architecture, hyper-parameters, or limited data size.

Predictive Efficiency

The efficacy of our PCA-NN implementation can be understood almost immediately by looking at Figure 9 which highlights the key strengths and limitations of this technique.

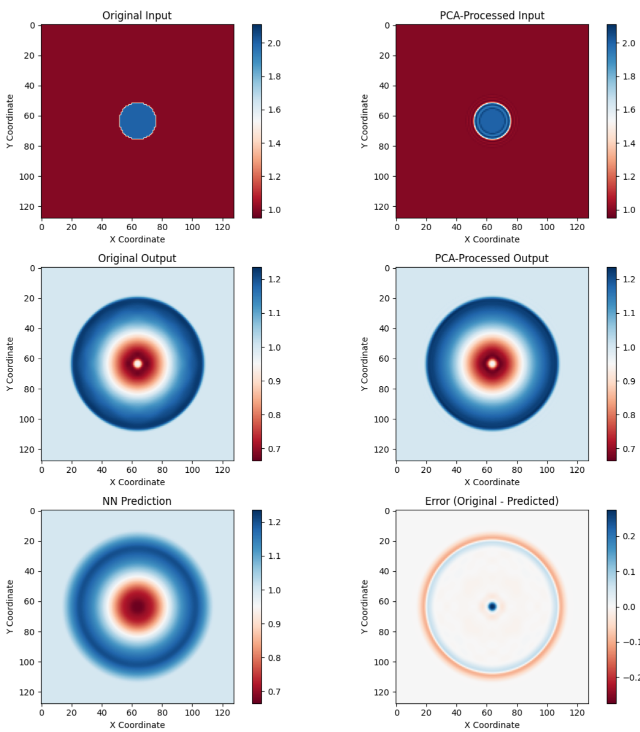

Figure 9: Diffusion-Reaction PDE example following PCA-NN

Figure 9 showcases the entire PCA-NN mapping process on a Reaction-Diffusion input function. Here we see PCA reduction down to 8 components, for both input and output, and its strong ability to capture the like-hood of the data. Please note that a PCA inverse has been applied to the PCA-processed input and output for a clearer visualisation in the original dimensionality.

The NN is trained to learn the mapping from the PCA reduced input to the PCA reduced output. Figure 9 shows us the models prediction on this test case, proving a strong understanding of the output shape. This is highlighted in the error which shows very little error with exception of points with a sharp rate of change.

This is one of the first things we can notice about the PCA-NN technique. While it does have a very strong ability to predict general mapping, it really seems to struggle at points with a high first-order derivative. We believe that this is partly due to PCA and partly due to the complex nature of PDEs.

PCA might struggle with this because it inherently approximates data by focusing on maximizing variance along its principal components. This approach can sometimes oversimplify the data, especially in areas with rapid changes or non-linearities, as it tends to smooth out these sharp features. In the context of PDEs, which often exhibit complex, non-linear behaviors and sharp gradients, this characteristic of PCA might lead to a loss of critical information during the dimensionality reduction process.

The model's difficulty at these sharp points might also be due to the difficulty of predicting the values themselves. Take for example the center point which in SWE is the one that undergoes the largest fluctuations in height. Due to the almost instability at this point, predicting this frequency might be difficult for the NN to learn and require more rigorous network tuning or more data.

5.2 Robustness against Noise

We elaborate on the outcomes of experiments aimed at evaluating the robustness of the Principal Component Analysis-Neural Network (PCA-NN) method for solving two-dimensional shallow water PDEs amidst various noisy conditions. Through rigorous testing, we have investigated the influence of three distinct types of noise—Gaussian, Uniform, and Salt and Pepper—on the PCA-NN model's accuracy and reliability. Our experimental protocol adhered to a fixed network architecture and hyperparameter set, as detailed in Section 3.3, utilizing 78 input and 22 output components to account for 99.99% of data variance.

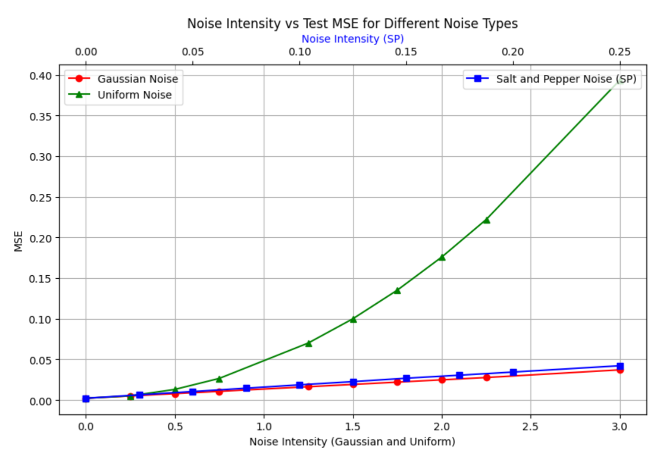

Figure 10: Test MSE for Different Noise Intensities

Analysis of Noise Impacts

An examination of the Mean Squared Error (MSE) across noise intensities, as presented in Figure 10, reveals distinct trends in error amplification attributable to each noise type. Gaussian and Salt and Pepper noises exhibit a linear increase in MSE with rising noise levels, indicating a proportional decline in model performance. Conversely, uniform noise demonstrates a non-linear growth in error, potentially exponential, which suggests a more severe degradation in model accuracy as noise intensity escalates.

The linear deterioration in model performance due to Gaussian and salt and pepper noises underscores the PCA-NN method's predictable loss in precision with increased noise levels. In stark contrast, uniform noise imparts an apparently exponential increment in MSE, indicating a drastic and rapid decline in the PCA-NN model's accuracy. This steep increase in error, particularly at higher noise intensities, necessitates a more robust noise-handling strategy.

Our comprehensive analysis across noise types has revealed that while the PCA-NN model maintains a degree of robustness against Gaussian and salt and pepper noise, it is significantly more vulnerable to uniform noise. This variation in noise impact necessitates tailored adjustments to the PCA components and neural network settings for optimal noise mitigation. It also highlights the importance of developing adaptive noise-handling mechanisms for PCA-NN frameworks to enhance the resilience of PDE solutions in noisy environments.

6 Discussion

The exploration of PCA-NN approaches in solving PDEs, particularly in noisy environments, opens up a multitude of avenues for further research and improvement. The potential advancements in this area are not only limited to refining PCA-NN models but also extend to their broader application in noisy data contexts.

The current study was able to simulate noisy environments and evaluate the PCA-NN technique in these environments. However, this study took a very general and broad look at noise and highlights the need for more specific noise types. This study would greatly value further research into noise types with specific real-world use-cases. Extending the application of PCA-NN models to real-world noisy datasets, like financial data or biological signals, could provide practical insights and improvements in model robustness and generalizability.

The study demonstrates the impact of PCA components on model performance. Further research could focus on dynamic selection of PCA components. This would allow for perhaps a targeting and segmented approach at applying PCA. Additionally, PCA could be adaptive to noise intensity and type, possibly through the use of machine learning algorithms.

The importance of these further research areas is underscored by the findings of this study, which reveal both the strengths and limitations of PCA-NN models in handling noise. This research thus acts as a springboard for more in-depth and applied studies in the field.

7 Conclusion

The objective of this paper was two-fold: to build upon and understand Bhat et al.'s research and to develop an understanding of "how robust PCA-neural network-based approaches are against noise in the data." This understanding has been nuanced and developed throughout the paper. Our replication of Bhat et al.'s research not only confirmed their findings but also provided additional insights into the behavior of PCA-NN models. This replication served as both a validation of their work and a foundation for our exploration into noise resilience. Our investigations revealed that while PCA-NN models show a degree of robustness to Gaussian noise, their performance is significantly challenged by Uniform and Salt and Pepper noise. This finding is crucial for understanding and improving the application of PCA-NN in real-world noisy environments. The research undertaken provides a clearer picture of the strengths and vulnerabilities of PCA-NN models in noisy conditions, emphasizing the need for continued innovation in this area, particularly in enhancing noise resilience. The findings from this study contribute to the academic understanding of PCA-NN models and have practical implications for their application in various fields where noise is an inherent part of the data. In conclusion, this study successfully extends the work of Bhat et al. by providing a deeper understanding of the robustness of PCA-NN approaches against noise, opening up new possibilities for improvement and application, and marking a significant step forward in the field of neural network research and its practical applications.

References

[1] I. Kinnmark, *The Shallow Water Wave Equations: Formulation, Analysis and Application*, ser. Lecture Notes in Engineering. Springer Berlin Heidelberg, 2012. [Online]. Available: https://books.google.nl/books?id=iWDmCAAAQBAJ

[2] J. Han, A. Jentzen, and E. Weinan, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, pp. 8505–8510, 2018.

[3] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, "Model reduction and neural networks for parametric pdes," 2021.

[4] E. E. Clothiaux and C. M. Bachmann, *Neural Networks and Their Applications*. Dordrecht: Springer Netherlands, 1994, pp. 11–52. [Online]. Available: https://doi.org/10.1007/978-94-011-1122-5_2

[5] W. Cao and W. Zhang, "Machine learning of partial differential equations from noise data," 2021.

[6] S. Brunton and J. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. [Online]. Available: https://books.google.nl/books?id=CYaEDwAAQBAJ

[7] M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert, "PDEBench

Datasets," 2022. [Online]. Available: https://doi.org/10.18419/darus-2986

[8] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, "Neural operator: Learning maps between function spaces," *CoRR*, vol. abs/2108.08481, 2021. [Online]. Available: https://arxiv.org/abs/2108.08481

[9] T. P. Pagano, R. B. Loureiro, F. V. N. Lisboa, R. M. Peixoto, G. A. S. Guimarães, G. O. R. Cruz, M. M. Araujo, L. L. Santos, M. A. S. Cruz, E. L. S. Oliveira, I. Winkler, and E. G. S. Nascimento, "Bias and unfairness in machine learning models: A systematic review on datasets, tools, fairness metrics, and identification and mitigation methods," *Big Data and Cognitive Computing*, vol. 7, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2504-2289/7/1/15