Deep Learning Approach of Physical Model Regression and Its Application in Digital Image Denoising

Qinfeng Zhu [*1] Tianyu Zhang [*1]

Abstract

We introduce dynamical weights sum of squared errors loss function (DWSSE), discuss and experiment how it will enhanced the performance of physical model regression (PMR). Finally, we apply our regression methods to the area of digital image processing and generated a new powerful method in digital image denoising.

1. Introduction

In recent years, many authors are doing research on deep neural networks (DNNs) methods in solving questions from differential equations (Raissi, 2018; Sirignano & Spiliopoulos, 2018; Weinan et al., 2017; Han et al., 2018; 2017). However, a special kind of questions about how to recover solutions of differential equations from scattered and potentially noisy observation, as we called physical model regression (PMR), still remains to be thoroughly studied.

To clarify the main idea, Let us consider an electrodynamics model, which is given by the 2-dimensional Poisson equation in region $\Omega = [0,1] \times [0,1]$:

$$\Delta\varphi(x,y) = f(x,y,k) \tag{1}$$

where φ is electric potential and

$$f(x,y,k) = k \sin \pi x \sin \pi y$$

is the non-homogeneous term with a unknown parameter k. Now, assume we have data $\{(x_i, y_i, \varphi_i^*)\}_{i=1}^N$ from real-world measurement of electric potential φ, and we want to reconstruct φ according to equation (1) and the data we measured (Figure 1). This is an example of PMR.

There are three direct problems:

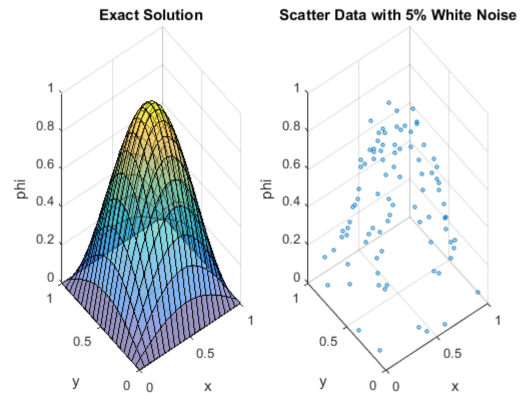

Figure 1. Poisson equation: the left figure is exact solution to this question with parameter $k = -2\pi^2$ and the right figure is the simulation result of measured data with 5% white noise.

- Most of differential equations are notoriously hard to find the analytical solutions or even do not have analytical solutions. Meanwhile, we don't know the boundary conditions or initial conditions. Therefore, even if we know how to solve this kind of equations analytically, we would still fail to make contribution to the question. We have to seek for some numerical methods.
- We have no prior methods to calculate the value of unknown parameters in physical models. This makes many tradition methods, such as finite difference, invalid in solving this question.
- The data from measurement always contain noise. So, noise robustness is important in our methods selection.

From 2017, Raissi et al. published a series of papers (Raissi, 2018; Raissi et al., 2017a;b) on a deep neural network approach to solve this kind of questions. Briefly speaking, for the partial differential equations, which have the form

$$u_t = \mathcal{N}(t, x, u, u_x, u_{xx}, \cdots), \tag{2}$$

[*]Equal contribution [1]Computer Science Department, New York University, New York City, USA. Correspondence to: Qinfeng Zhu <qz981@nyu.edu>, Tianyu Zhang <tz904@nyu.edu>.

with the scatter data $\{(t^i, x^i, u^i)\}_{i=1}^N$, they used two DNNs to approximate the solution u and the function $\mathcal{N}$ separately, applied automatic differentiation (Baydin et al., 2018) to handle the differentiation in equations, then trained both DNNs by the Adam optimization algorithm to minimize the sum of squared errors loss function (SSE):

$$SSE = \sum_{i=1}^{N}(|u(t^i, x^i) - u^i|^2 + |f(t^i, x^i)|^2), \quad (3)$$

where

$$f(t^i, x^i) = u_t^i - \mathcal{N}(t^i, x^i, u^i, u_x^i, \cdots), \quad (4)$$

in which the term $|u(t^i, x^i) - u^i|^2$ corresponds to the error between predictive values and the given scatter data, while the second term $|f(t^i, x^i)|^2$ corresponds to error between the DNNs and the exact solution of equation (2). Minimizing SSE amounts to adjust the DNNs not only to fit the scatter data but also to satisfy equation (2). Equation (2) helps us check the inner structure of our form. It is worth mentioned that their method is free from prior estimate of unknown parameters in $\mathcal{N}$. This makes the method quite powerful in real-world applications.

However, their results are not so accurate as to be used in industry. Especially, when the noise is getting large, the error will increase dramatically (Raissi et al., 2017b). Also, the authors left many open problems (Raissi, 2018) that need further investigation.

In our research, in order to enhance the performance of DNNs approach of physical model regression, we considered a generalization of SSE, as we called dynamical weights sum of squared errors loss function (DWSSE), whose formula is given by equation (5) in section 2 of this paper. We experimented the performance of PMR when using DWSSE, and compared the results to the results of PMR when using SSE.

One of the open problems that Raissi announced is how to choose regularization method in PMR. However, the well-known methods such as L^1/L^2 regularization (Goodfellow et al., 2016) generates bad results for PMR questions. In 2018, a new regularization method, called physics-informed (PI) regularization, was introduced to handle this problem (Nabian & Meidani, 2018). We also experimented the results by applying PMR and tested the result when combining DWSSE and PI regularization.

In the rest of our research, we considered a PDE-based digital image denoising model, investigated how our methods can be use in solving this model, and conducted performance tests as well.

2. Dynamical Weights Sum of Squared Errors Loss Function

We define the dynamical weights sum of squared errors loss function as

$$DWSSE = \sum_{i=1}^{N}(w_u^i|u(t^i, x^i) - u^i|^2 + w_f^i|f(t^i, x^i)|^2), \quad (5)$$

where $w_u^i, w_f^i \in [0, 1]$ are weights of terms $|u(t^i, x^i) - u^i|^2$ and $|f(t^i, x^i)|^2$ separately. As can be seen, when $w_u^i = w_f^i = 1$, DWSSE goes back to SSE. In general, we allow w_u^i, w_f^i to change according to the value of terms $|u(t^i, x^i) - u^i|^2$ and $|f(t^i, x^i)|^2$ in order to get best training consequence. The optimal selection of weights w_u^i, w_f^i is a sophisticated problem. Due to the complexity of deep neural networks, the optimal selection of weights can hardly be derived from a single mathematical theory. However, we can still make some choices according to certain principles.

For our purposes, we consider the following weight selection model:

$$w_u^i = 1, \quad w_f^i = \sigma(|u(t^i, x^i) - u^i|^2) \quad (6)$$

where $\sigma(x)$ is a activation function with the properties that $\sigma(0) = 0, \sigma(+\infty) = 1$ and σ is monotonically increasing on $(0, \infty)$.

The intuition behind equations (6) is that when the error term $|u(t^i, x^i) - u^i|^2$ is large, namely the DNN which approximates u is far away from the exact solution u. By differential equations theory, we know the term $|f(t^i, x^i)|^2$ cannot reflect the true error between this DNN and exact solution u (Evans, 1998). So, at the beginning of DNNs' training, this term makes no or even negative effects on reducing the total error, so we should endow it with a small weight at the beginning. In contrast, when the error term $|u(t^i, x^i) - u^i|^2$ is comparatively small, the DNN which approximates u lies in a small enough neighborhood of the exact solution u, then the term $|f(t^i, x^i)|^2$ indeed approximates the true error between this DNN and exact solution u. Also, at this time, reducing noisy data cannot provide us with more useful information, so we should increase the weight of $|f(t^i, x^i)|^2$ in order to make our first DNN more close to the exact solution u. This characteristic also makes our method robust to noise, as we can seen in next section.

3. Experiments and Results

In this section, we will discuss the details of our PMR experiments and results on two well known hard-to-compute physical models: Burgers' equation and KdV equation. The PMR experiments are conducted by TensorFlow in Python environment. To evaluate and compare the performance between DWSSE-based PMRs and SSE-based PMRs, we must exclude influence from other factors, so we modified the source code and used the same dataset[1] as in (Raissi, 2018). Physics-informed (PI) regularization (Nabian & Meidani, 2018) is also used and compared in this section. The regularization term is given by

$$\frac{1}{2n}\sum_{i=1}^{n}|\tilde{u}_t - \mathcal{N}(t, x, \tilde{u}, \tilde{u}_x, \tilde{u}_{xx}, \cdots)|^2 \qquad (7)$$

where $\tilde{u}$ is the DNN that approximates exact solution u and $\mathcal{N}$ is the function given in equation (2). We use $\sigma(x) = \tanh(x)$ as the activation function in our weights selection model (6).

3.1. Burgers' Equation

The Burgers' equation we used is

$$u_t + uu_x - 0.1u_{xx} = 0.$$

Define $\mathcal{N} = -uu_x + 0.1u_{xx}$, then we see this equation has the form of equation (2).

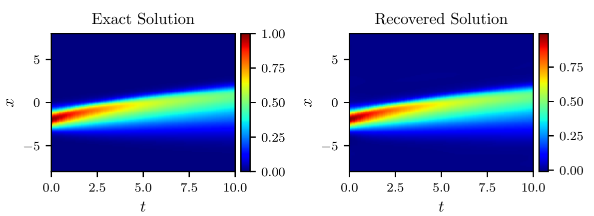

Figure 2. Solutions of Burgers' equation: The left figure is the exact solution, and the right figure is the recovered solution from scatter data with 5% white noise, by the SSE-based PMR method and PI regularization.

The results of PMR for Burgers' equation with and without PI regularization are shown in Table 1, 2.

We see from the Table 1 that DWSSE can partly serve as regularization, so that the L^2 error will not increase exponentially as noise increasing linearly in Table 1. This result justifies our expectation in last section. Consider both tables, we see there is a large improvement of the results when using DWSSE as the training loss function.

[1]The URL is https://github.com/maziarraissi

Table 1. The L^2 error of PMR for Burgers' equation without PI regularization.

	Clean data	1% noise	2% noise	5% noise
SSE	4.78e-03	2.64e-02	1.09e-01	4.56e-01
DWSSE	9.96e-04	2.21e-03	4.20e-03	1.04e-02

Table 2. The L^2 error of PMR for Burgers' equation with PI regularization.

	Clean data	1% noise	2% noise	5% noise
SSE	9.20e-04	2.42e-03	4.71e-03	6.11e-03
DWSSE	5.61e-04	1.82e-03	3.56e-03	4.55e-03

3.2. KdV Equation

The KdV equation we used is

$$u_t + uu_x + u_{xxx} = 0$$

Define $\mathcal{N} = -uu_x - u_{xxx}$, then we see this equation has the form of equation (2).

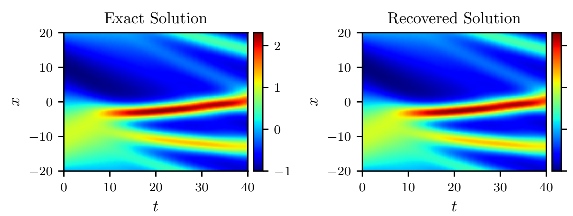

Figure 3. Solutions of KdV equation: The left figure is the exact solution, and the right figure is the recovered solution from scatter data with 5% white noise, by the SSE-based PMR method and PI regularization.

The results of PMR for KdV equation with and without PI regularization are shown in Table 3, 4.

As is shown, due to the higher-order derivative, the DNNs training is expected to take large quantity of time, and the numerical result cannot be as good as lower-order differential equations. However, the results by DWSSE still surpass the results by SSE.

4. Application: Digial Image Denoising

In this section, we will develop an application of PMR methods in digital image denoising.

Table 3. The L^2 error of PMR for KdV equation without PI regularization.

	Clean data	1% noise	2% noise	5% noise
SSE	1.47e-03	5.42e-03	7.71e-03	1.41e-02
DWSSE	1.22e-03	2.35e-03	5.21e-03	9.44e-03

Table 4. The L^2 error of PMR for KdV equation with PI regularization.

	Clean data	1% noise	2% noise	5% noise
SSE	1.35e-03	2.43e-03	4.82e-03	8.52e-03
DWSSE	9.61e-04	1.38e-03	3.66e-03	6.45e-03

4.1. Basic Theories

In RGB system, a (color) image can be considered as a function $f(x, y, c)$ which represents the magnitude at (x, y, c), where $(x, y) \in \Omega$ for some continuous region Ω and $c \in \{Red, Green, Blue\}$. For a $M \times N$ pixel digital image F we means a $N \times M \times 3$ magnitude matrix, where the entries $F[i, j, k]$ are sampling results of $f(x_i, y_j, k)$ (Gonzalez & Woods, 2017).

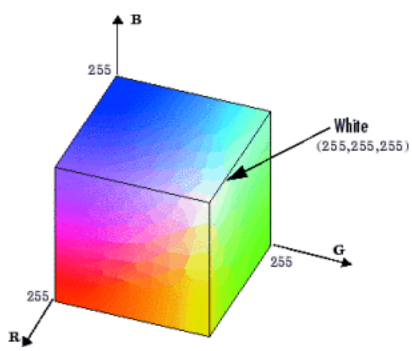

Figure 4. RGB system (Source: msu.edu): the RGB color system can be seen as a cube with 255 discrete points per side.

Image may also contain noise, without loss of generosity, we assume the following noise model:

$$f_0 = f + n$$

where f_0 is the image observation, f is the image without noise, and n is white noise term with average 0. Let F_0 be corresponding sampling result, so we have corresponding digital image noise model:

$$F_0 = F + N$$

where F is the digital image without noise and N is white noise matrix with average 0.

In (Chan & Shen, 2005), a variational denoise model is proposed as

$$\hat{f} = \arg\min_f \left\{ \frac{\alpha}{2} \int_\Omega |\nabla f|^2 \mathrm{d}x + \frac{\lambda}{2} \int_\Omega (f - f_0)^2 \mathrm{d}x \right\} \tag{8}$$

where α and λ are unknown parameters. By calculus of variation, we see the variational equation (8) is equivalent to the elliptic boundary value problem

$$-\alpha \Delta f + \lambda f = \lambda f_0 \tag{9}$$

with boundary condition that the outward derivative $\partial f / \partial v$ on the boundary $\partial\Omega$ always equals to 0. However, this model is hard to use for at least two reasons:

- We do not actually know f_0 because generally we can only get the discrete noisy digital image data F_0.
- There is no way to impose the boundary condition into the model solving.

Actually, F_0 can be seen as the discrete scatter data sampling from f. So this problem is equivalent to recovering f from F_0. This fact inspires us to use our PMR method to solve this problem. upon getting the estimating $\hat{f}$ of f from F_0, we can evaluate $\hat{f}$ at lattice points and get the denoised digital image F.

4.2. Solution Methodology

Because for any noisy color image f_0, it has Red\Green\Blue three levels. We can sequentially denoise each level and integrate them together to form the denoised color image f. So, in the rest of this section, all image and digital image are assumed to be black-and-white image with only one level.

Equation (9) can be rewritten as

$$f_{yy} = -f_{xx} + \frac{\lambda}{\alpha}(f - f_0). \tag{10}$$

By letting $\mathcal{N} = -f_{xx} + \frac{\lambda}{\alpha}(f - f_0)$, we have the train model

$$f_{yy} = \mathcal{N}(f, f_{xx}, f_0)$$

with unknown parameters λ and α.

In consistent with what we did before, we use two DNNs to approximate f and the function $\mathcal{N}$ separately, applied automatic differentiation to handle the differentiation in equations, and trained both DNNs by

L-BFGS methods to minimize the dynamical weights sum of squared loss function (DWSSE).

This method can be briefly shown as the algorithm 1.

Algorithm 1 PMR Denoising

Input: noisy image F_0
1. Use PMR to recover f from F_0
2. Evaluate f on lattice points to recover F
3. Output F

4.3. Results

The results is shown as Figure 5.

Figure 5. Recover results: The result of PMR filtering is better than well-used Median filtering. The image it recovered is more smooth.

4.4. Comments

Although we get a decent result from this method, the running time of our algorithm is too large to be frequently used in practice.

There are many other models that can be used instead. For example, we can use unitized gradient in equation (9):

$$-\alpha \frac{\Delta f}{|\Delta f|} + \lambda f = \lambda f_0$$

so the training process will not be influenced by the magnitude of gradient and the training efficiency is expected to be enhanced. Also, convolution neural networks (CNNs) are expected to be good selection of neural network for this denoising task, due to the grid-like topology of the image data (Goodfellow et al., 2016).

Because of the time limitation, we do not make complete numerical experiments for them. For the reader who are interested in this topic, he or she can refer to (Lysaker et al., 2003; Gilboa et al., 2006; Liu et al., 2011) and references therein.

5. Conclusion

In this research, we have presented a DWSSE version of DNNs-based PMR that can be used to recover solutions of differential equations from noisy scatter points data. By numerical experiments, our method reveals its power in accuracy and robustness when handling noisy data. Finally, we made a new step to applying this physical model regression in digital image denoising.

However, there are many other applicable methods that can be used in PMR or digital image denoising, that we cannot fully experiment on, due to the time limitation. Especially, convolution neural networks (CNNs) are supposed to have a fancy performance on PMR problems because physical models are generally easy to be grided. In addition, CNNs can mitigate the complexity associated with partial differential equations with very high-dimensional inputs (Raissi, 2018). We expect further investigation on those wonderful topics.

Acknowledgements

We sincerely thank graders' careful reading and grading. All the data and codes used in this research are available on GitHub at https://github.com/MichaelZhangty/ML_Finalproject.git

References

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey. Journal of Marchine Learning Research, 18:1–43, 2018.

Chan, T. F. and Shen, J. J. Image processing and analysis: variational, PDE, wavelet, and stochastic methods, volume 94. Siam, 2005.

Evans, L. C. Partial Differential Equations. American Mathematical Society, 2nd edition, 1998.

Gilboa, G., Sochen, N., and Zeevi, Y. Y. Variational denoising of partly textured images by spatially varying constraints. IEEE Transactions on Image Processing, 15(8):2281–2289, 2006.

Gonzalez, R. C. and Woods, R. E. Digital Image Processing. Pearson Education, 4nd edition, 2017.

Goodfellow, I., Bengio, Y., and Courville, A. Deep learning. MIT press, 2016.

Han, J., Jentzen, A., and Weinan, E. Overcoming the curse of dimensionality: Solving high-dimensional partial differential equations using deep learning. arXiv preprint arXiv:1707.02568, pp. 1–13, 2017.

Han, J., Jentzen, A., and Weinan, E. Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences, 115(34):8505–8510, 2018.

Liu, X., Huang, L., and Guo, Z. Adaptive fourth-order partial differential equation filter for image denoising. Applied Mathematics Letters, 24(8):1282–1288, 2011.

Lysaker, M., Lundervold, A., and Tai, X.-C. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. IEEE Transactions on image processing, 12(12):1579–1590, 2003.

Nabian, M. A. and Meidani, H. Physics-informed regularization of deep neural networks. arXiv preprint arXiv:1810.05547, 2018.

Raissi, M. Deep hidden physics models: Deep learning of nonlinear partial differential equations. Journal of Machine Learning Research, 19(25):1–24, 2018.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561, 2017a.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:1711.10566, 2017b.

Sirignano, J. and Spiliopoulos, K. Dgm: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics, 375:1339–1364, 2018.

Weinan, E., Han, J., and Jentzen, A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Communications in Mathematics and Statistics, 5(4):349–380, 2017.