

PINNs_Poisson

Implementation

- Function: $h \in FC(W = 20, L = 5, d_{in} = 2, d_{out} = 1, \sigma = \tanh)$
- Input: x, y
- Output: $u(x, y)$
- Optimizer: **Adam optimizer**
- Loss function: $L_{\theta} = w_1 L_{data} + w_2 L_{PDE}$, we calculate L_{data} with MSE between true and prediction solution, and calculate L_{PDE} with residual equation $F(x) = u_{xx} + u_{yy} - f(x, y)$, w_1 and w_2 are the weights.
- Training Set: 30 points on each boundary and 500 collocation points.(chosen by Latinhypercube)

Here we focus on the Poisson equation of the form:

$$u_{xx} + u_{yy} = f(x, y)$$

having the following 5 manufactured exact solutions taken from Nishikawa (2023):

- (a) $u(x, y) = e^{xy}$
- (b) $u(x, y) = e^{kx} \sin(ky) + \frac{1}{4}(x^2 + y^2)$
- (c) $u(x, y) = \sinh(x)$
- (d) $u(x, y) = e^{x^2+y^2}$
- (e) $u(x, y) = e^{xy} + \sinh(x)$

and the boundary conditions are the true solution $u(x, y)$ on the boundary $x = 0, x = 1, y = 0, y = 1$ respectively.

Here, we focus on the case $u = e^{xy}$ to discuss the approximation accuracy on different aspect.

How to evaluate the accuracy of different aspects

I use L_1, L_2, L_{inf} -norm to evaluate the accuracy.

and the definitions are as below, $u(x, y)$ is the neural network function and $tru(x, y)$ is the true solution of the Poisson equation:

$$\|u(x, y) - tru(x, y)\|_{L_1} = \int_{\Omega} |u(x, y) - tru(x, y)| dx dy$$

L1-norm is used to evaluate the average error between the NN functions and true solutions.

$$\|u(x, y) - tru(x, y)\|_{L_2} = \sqrt{\int_{\Omega} |u(x, y) - tru(x, y)|^2 dx dy}$$

L2-norm is more sensitive on the regions that have larger errors than L1-norm.

$$\|u(x, y) - tru(x, y)\|_{L_{inf}} = \max_{x, y} |u(x, y) - tru(x, y)|$$

L_∞ -norm is used to evaluate the point with the largest error.

Here the integral is calculated by Simpson's Rule.

I partitioned the integral domain $\Omega = [0, 1] \times [0, 1]$ into 201×201 grid, because the number of intervals need to be even.

Since the Simpson's rule is $O(h^4)$ method, and here the h is $1/200$, the error E is about $O(0.005^4)$ in theory.

Different Number of Nuerons in hidden layers

Here the number of hidden layers, data points are the same as hypothesis.

	10	20	40	80
L1-norm	$2.44925 * 10^{-3}$	$1.84068 * 10^{-3}$	$9.44907 * 10^{-4}$	$3.21161 * 10^{-3}$
L2-norm	$2.76515 * 10^{-3}$	$2.07960 * 10^{-3}$	$1.06338 * 10^{-3}$	$3.54485 * 10^{-3}$
L_∞ -norm	$1.10676 * 10^{-2}$	$8.31531 * 10^{-3}$	$3.74746 * 10^{-3}$	$9.45397 * 10^{-3}$

Different Number of hidden layers

	3	6	12	24
L1-norm	$1.83021 * 10^{-3}$	$1.84068 * 10^{-3}$	$7.85961 * 10^{-3}$	$5.99671 * 10^{-3}$
L2-norm	$2.06741 * 10^{-3}$	$2.07960 * 10^{-3}$	$8.49797 * 10^{-3}$	$6.37006 * 10^{-3}$
L_∞ -norm	$8.96120 * 10^{-3}$	$8.31531 * 10^{-3}$	$1.72407 * 10^{-2}$	$1.43781 * 10^{-2}$

Different Number of data points

	15*4 + 250	30*4 + 500	60*4 + 1000	120*4 + 2000
L1-norm	$1.75017 * 10^{-3}$	$1.84068 * 10^{-3}$	$2.01064 * 10^{-3}$	$8.88932 * 10^{-4}$
L2-norm	$2.00143 * 10^{-3}$	$2.07960 * 10^{-3}$	$2.08876 * 10^{-3}$	$1.02585 * 10^{-3}$
L_∞ -norm	$8.17026 * 10^{-3}$	$8.31531 * 10^{-3}$	$3.48488 * 10^{-3}$	$4.70626 * 10^{-3}$

Summary

From the above summary, we can not obtain the result as expected in finite difference, by increasing the number of data points by 2 times, and the error becoomes to 1/4.

And by incresing the number of hidden layers and neurons per layer as well.

With the same order of accuracy, adding more neurons per hidden layer, more hidden layers, and more data points would use more parameters, which means cost more computational power.

So, in PINNs we emphasize on finding the minimum number of neurons in hidden layer, hidden layers, data points to the accuracy we want.