Study of FBPINN for shallow networks

Samuel Anderson, Victorita Dolean, Jennifer Pestana

1 Introduction

One of the popular approaches in SciML are physics-informed neural networks (PINNs) [1, 4], which are designed to approximate the solution to the boundary value problem

$$\begin{aligned} \mathcal{N}[u](\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \\ \mathcal{B}_k[u](\mathbf{x}) &= g_k(\mathbf{x}), \quad \mathbf{x} \in \Gamma_k \subset \partial\Omega \end{aligned} \tag{1}$$

where $\mathcal{N}[u](\mathbf{x})$ is a differential operator, u is the solution and $\mathcal{B}_k(\cdot)$ is a set of boundary conditions, such that the solution u is uniquely determined. The approximation to the solution of (1) is given by a neural network $u(\mathbf{x}, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is a vector of all the parameters of the neural network (i.e., its weights and biases). The network is trained via the loss function $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{BC}}$ where

$$\mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}) = \frac{\lambda_I}{N_I} \sum_{i=1}^{N_I} (\mathcal{N}[u](\mathbf{x}_i, \boldsymbol{\theta}) - f(\mathbf{x}_i))^2, \tag{2}$$

$$\mathcal{L}_{\text{BC}}(\boldsymbol{\theta}) = \sum_{k=1}^{N_k} \frac{\lambda_B^k}{N_B^k} \sum_{j=1}^{N_B^k} (\mathcal{B}_k[u](\mathbf{x}_j^k, \boldsymbol{\theta}) - g_k(\mathbf{x}_j^k))^2. \tag{3}$$

Here, $\{\mathbf{x}_i\}_{i=1}^{N_I}$ is a set of collocation points sampled in the interior of the domain, $\{\mathbf{x}_j^k\}_{j=1}^{N_B^k}$ is a set of points sampled along each boundary condition, and λ_I and λ_B^k are well-chosen scalar hyperparameters which ensure that the terms in the loss function are well balanced.

Whilst PINNs have proven to be successful for solving many different types of differential equations, they often struggle to scale to problems with larger domains and more complex, multi-scale solutions [2, 5]. This is in part due to the spectral bias of neural networks [3] (their tendency to learn higher frequencies much slower than lower frequencies), and the increasing size of the underlying PINN optimisation function. One way to alleviate these scaling issues is to combine PINNs with a domain

Victorita Dolean
. e-mail: `v.dolean.maini@tue.nl`
.

decomposition method (DDM); by taking a divide-and-conquer approach, one hopes that the large, global optimisation problem can be turned into a series of smaller and easier localised problems. In particular, [2] proposed finite basis physics-informed neural networks (FBPINNs) where the global PINN is partitioned into many local networks that are trained to approximate the solution on an overlapping domain decomposition.

Here I just copy pasted some text from a previous contribution as an example. But need to include here a proper literature review including the RFM method, ELM approaches and Xu papers on spectral bias

First we briefly present the FBPINN method introduced by [2] from a DDM perspective. The FBPINN method can be seen as a network architecture that allows for a localization of the network training. Therefore, let us consider a set of collocation points $X = \{\mathbf{x}_i\}_{i=1}^N$ in the global domain Ω and a decomposition into overlapping domains $\Omega = \cup_{j=1}^J \Omega_j$ inducing a decomposition into subsets of collocation points $X_j = \{\mathbf{x}_i^j\}_{i=1}^{N_j}$, $j = 1, ..., J$. As usual in overlapping Schwarz methods, $X = \cup_{j=1}^J X_j$ is not disjoint. For each subdomain Ω_j, we denote $\mathcal{N}_j$ the index set of neighbouring subdomains. We now define the global network u as the sum of local networks $u_j(\mathbf{x}, \boldsymbol{\theta}_j)$ weighted by locally supported window functions ω_j:

$$u = \sum_{j, \mathbf{x}_i \in \Omega_j} \omega_j u_j. \tag{4}$$

If we insert eq. (4) into eq. (2), we see that the global loss function can be written as:

$$\mathcal{L}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J) = \frac{1}{N} \sum_{i=1}^N \left(\mathcal{N}[C \sum_{j, \mathbf{x}_i \in X_j} \omega_j u_j](\mathbf{x}_i, \boldsymbol{\theta}_j) - f(\mathbf{x}_i) \right)^2. \tag{5}$$

2 FBPINN for a shallow network: the two domain case

Suppose we want to use an FBPINN like architecture to approximate a function $f : [0, L] \longrightarrow \mathbb{R}$ on a shallow partition of unity network:

$$u(x, \mathbf{a}) = \omega_1 u_1 + \omega_2 u_2 = \sum_{i=1}^n a_i^1 \sigma(w_i x + b_i) \omega_1(x) + \sum_{i=1}^n a_i^2 \sigma(\tilde{w}_i x + \tilde{b}_i) \omega_2(x)$$

where the weights and biases $w_i, \tilde{w}_i, b_i, \tilde{b}_i$ are random but fixed and we train only for $\mathbf{a} = [\mathbf{a}^1, \mathbf{a}^2]$. For simplicity we suppose the two networks u_1 and u_2 have the same capacity.

3 Numerical results

4 Discussion and conclusions

References

1. Lagaris, I.E., Likas, A., Fotiadis, D.I.: Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks **9**(5), 987–1000 (1998)
2. Moseley, B., Markham, A., Nissen-Meyer, T.: Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. arXiv preprint arXiv:2107.07871 (2021)
3. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: K. Chaudhuri, R. Salakhutdinov (eds.) Proceedings of the 36th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 97, pp. 5301–5310. PMLR (2019)
4. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational physics **378**, 686–707 (2019)
5. Wang, S., Wang, H., Perdikaris, P.: On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering **384**, 113938 (2021)