Step 1

$$u_{NN}(x) = \sum_{i=1}^{n} a_i \sigma(w_i x + b_i) \qquad \text{shallow}$$

a_i ↳ learnable coeff. ; w_i, b_i ↳ random

$$\underline{a} = (a_i)_{i=1,\dots,n}$$

Sol PDE $\begin{cases} -u''(x) = f(x) \\ u(0) = u(1) = 0 \end{cases}$

1. $\mathcal{L}(\underline{a}) = \int_0^1 (u_{NN}(x) - f)^2 \, dx$ → function approx
2. $\mathcal{L}(\underline{a}) = \frac{1}{2}\int_0^1 (u_{NN}')^2 \, dx - \int_0^1 u_{NN}(x) f(x) \, dx$, FNM
3. $\mathcal{L}(\underline{a}) = \int_0^1 (u_{NN}'' + f)^2 \, dx$ PINNs

Apply gradient descent to $\mathcal{L}$

$$\underline{a}^{k+1} = \underline{a}^k - \mu \nabla_{\underline{a}} \mathcal{L}(\underline{a}^k) \quad , \quad w_i, b_i \text{ randomized}$$

σ : ReLu, Hat, tanh (for PINN)

Step 2 $\quad u_{NN}(x) = \sum_{i=1}^{J} \omega_i \underbrace{\sum_{j=1}^{N_i} a_j^i \sigma(w_j^i x + b_j^i)}$ → "local" networks → start with $J = 2$

ω_i ↳ window functions

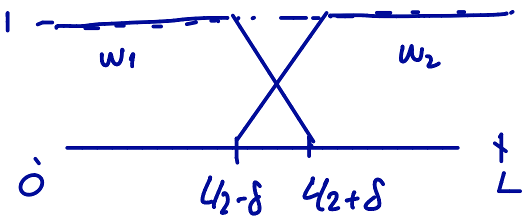
Repeat the above. Randomize w_j^i, b_j^i

→ one-level FPBINN ⟶ RFM without a coarse network

$$M_{ij} = \int_0^L \sin(w_i x + b_i) \sin(w_j x + b_j) \, dx$$

Study of RFM

w_j are randomly sampled in an interval / for b_i it can be a different one



partition of unity

$$u_{NN} = \sum_{i=1}^{n_1} a_i \sigma(w_i x + b_i) \, \omega_1(x) + \sum_{i=1}^{n_2} \tilde{a}_i \sigma(\tilde{w}_i + \tilde{b}_i) \, \omega_2(x)$$

$$L(a) = \frac{1}{2}\int_0^L (f(x) - u_{NN}(x))^2 \, dx = \frac{1}{2}\int_0^{L/2-\delta} (f(x) - u_{NN}^1(x))^2 dx + \frac{1}{2}\int_{L/2-\delta}^{L/2+\delta} (f(x) - u_{NN}^1 \omega_1 - u_{NN}^2 \omega_2)^2$$

$$+ \frac{1}{2}\int_{L/2+\delta}^{L} (f(x) - u_{NN}^2(x))^2 \, dx$$

$$= -b_1^T a_1 + \frac{1}{2} a_1^T M_1 a_1 - [\tilde{b}_1 \ \tilde{b}_2]^T [a_1 \ a_2]$$

$$- b_2^T a_2 + \frac{1}{2} a_2^T M_2 a_2 + \frac{1}{2} [a_1 \ a_2]^T \tilde{M} [a_1 \ a_2]$$

$$(M_1)_{ij} = \int_0^{L/2-\delta} \sigma(w_i x + b_i) \sigma(w_j x + b_j) \, dx.$$

$$\tilde{M} = \begin{bmatrix} \tilde{M}_1 & M_{12} \\ M_{21} & \tilde{M}_2 \end{bmatrix} \qquad (\tilde{M}_1)_{ij} = \int_{L/2-\delta}^{L/2+\delta} \sigma(w_i x + b_i) \cdot \sigma(w_j x + b_j) \cdot \omega_1^2(x)$$

$$(M_2)_{ij} = \int_{L/2+\delta}^{L} \sigma(\tilde{w}_i x + \tilde{b}_i) \sigma(\tilde{w}_j x + \tilde{b}_j) \, dx$$

$(\tilde{M}_2)_{ij}$ same with $\tilde{w}, \tilde{b}$ and $\omega_2^2(x)$

$$= -b_1^T a_1 + \frac{1}{2} a_1^T M_1 a_1 \quad - [\tilde{b}_1 \; \tilde{b}_2]^T [a_1 \; a_2]$$

$$-b_2^T a_2 + \frac{1}{2} a_2^T M_2 a_2 + \frac{1}{2} [a_1 \; a_2]^T \tilde{M} [a_1 \; a_2]$$

$$\tilde{M} = \begin{bmatrix} \tilde{M}_1 & M_{12} \\ M_{21} & \tilde{M}_2 \end{bmatrix} \qquad M_{12} = M_{21}^T$$

$$\mathcal{L}(\underline{a}) = \text{---} + \frac{1}{2} a_1^T M_1 a_1 + \frac{1}{2} a_2^T M_2 a_2 + \frac{1}{2} a_1^T \tilde{M}_1 a_1 + \frac{1}{2} a_2^T \tilde{M}_2 a_2 + \frac{1}{2} a_1^T M_{12} a_2 + \frac{1}{2} a_2^T M_{21} a_1$$

$$= \text{- - -} + \frac{1}{2} [a_1 \; a_2]^T \begin{bmatrix} M_1 + \tilde{M}_1 & M_{12} \\ M_{21} & M_2 + \tilde{M}_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

$$\nabla_a \mathcal{L}(\underline{a}) = \text{---} + \begin{bmatrix} M_1 + \tilde{M}_1 & M_{12} \\ M_{21} & M_2 + \tilde{M}_2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

$$M = \begin{bmatrix} M_1 + \tilde{M}_1 & M_{12} \\ M_{21} & M_2 + \tilde{M}_2 \end{bmatrix}$$ is this matrix better conditioned?

$$w_1(x) = \frac{L/2 + \delta - x}{\delta} \quad \text{on } (L/2 - \delta, L/2 + \delta)$$

$$w_2(x) = \frac{x - (L/2 - \delta)}{\delta} \quad \text{on } (L/2 - \delta, L/2 + \delta)$$

Entries of the matrix M are all computable

A few questions:

- assess the condition number of this matrix
- assess approximation properties
- can we have a rough estimate of the condition number?

Rayleigh coeff
$$\frac{x^T(M_1 + \tilde{M}_1)x + y^T(M_2 + \tilde{M}_2)y + 2x^T M_{12} y}{(x \; y)^T (x, y)} \geq \frac{\tilde{\lambda}_1 \|x\|^2 + \tilde{\lambda}_2 \|y\|^2 + 2x^T M_{12} y}{\|x\|^2 + \|y\|^2 + 2x^T y}$$

$\tilde{\lambda} = \min(\tilde{\lambda}_1, \tilde{\lambda}_2)$, what do we need about M_{12} to estimate

$x^T M_{12} y \geq \underline{\lambda}\, x^T y$? If we suppose x and y are of the same size (same number of weights for each

$\Rightarrow$ M_{12} is square (also symmetric if we use the same activation function)

Plan:

- maybe compute everything numerically then try to get some theory
- we would expect the M to be better conditioned than just working on the whole interval with larger networks