ANaGRAM: A Natural Gradient Relative to Adapted Metrics for efficient PINNs learning

Seminar of the Institute for Applied and Numerical Mathematics, KIT

Nilo Schwencke

July 24th, 2024

Todo

- NNTK plots in abolute value or log that goes into negatives also (avoid white areas)
- Add computing time plots
- Add references to Zeinhofer
- Complete references to Green function (make it one slide) and add the drawing of Green function function update and PINNs as least-square regression

Outline

Physics informed neural networks (PINNs)

Neural networks

Definition
A neural network is a smooth non-linear functional

$$u : \begin{cases} \mathbb{R}^P & \to & \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) \\ \theta & \mapsto & u_\theta \end{cases}$$

Universal approximation property of neural networks (Leshno et al., 1993): u_θ can approximate any function in $\mathrm{L}^2(\Omega \to \mathbb{R}, \mu)$, in particular any solution to a PDE.

How to approximate such a solution?

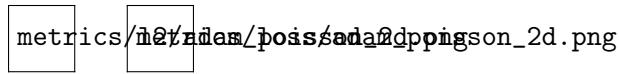
Answer: Just as we would for any neural network, *i.e.* by gradient descent (Lagaris et al., 1998; Raissi et al., 2019). More precisely, given the PDE:

$$\begin{cases} D[u] = f & \text{in } \Omega \\ B[u] = g & \text{on } \partial\Omega \end{cases},$$

we will optimize the loss:

$$\ell(\theta) := \frac{1}{2S_D} \sum_{i=1}^{S_D} \left(D[u_\theta](x_i^D) - f(x_i^D) \right)^2 + \frac{1}{2S_B} \sum_{i=1}^{S_B} \left(B[u_\theta](x_i^B) - g(x_i^B) \right)^2$$

PINNs shortcomings and drawbacks

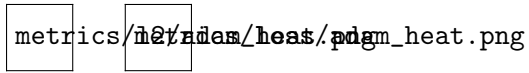


(a) L^2 error of PINN solution (b) Test loss of PINN solution

Figure: PINN solution under standard Adam optimization, to Laplace equation in 2 D:

$$\begin{cases} \Delta u = -2\pi^2 \sin(\pi x_1) \sin(\pi x_2) & \text{in } [0,1]^2 \\ u = 0 & \text{on } \partial[0,1]^2 \end{cases}$$

PINNs shortcomings and drawbacks



(a) L^2 error of PINN solution (b) Test loss of PINN solution

Figure: PINN solution under standard Adam optimization, to Heat equation in 1+1 D:

$$\begin{cases} \partial_t u - \frac{1}{4}\partial_{xx} u = 0 & \text{in } [0,1]^2 \\ u = 0 & \text{on } [0,1] \times \{0\} \bigcup [0,1] \times \{1\} \\ u = sin(\pi x) & \text{on } \{0\} \times [0,1] \end{cases}$$

Natural Gradient

Natural Gradient in a nutshell

Introduced by Amari and Douglas (1998) in the context of Information Geometry. Given a loss: $\ell : \theta \to \mathbb{R}^+$, the gradient descent:

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla \ell,$$

is replaced by the update:

$$\theta_{t+1} \leftarrow \theta_t - \eta G^\dagger \nabla \ell,$$

with G the Hessian of the Kullback-Leibler divergence. More generally in the context of Riemannian manifolds:

$$\theta_{t+1} \leftarrow \theta_t - \eta G^\dagger \nabla \ell,$$

with $G_{p,q} := \mathcal{G}_\theta(\partial_p u_\theta, \partial_q u_\theta)$, the Gram matrix of partial derivatives w.r.t a Riemannian-(pseudo) metric $\mathcal{G}_\theta$.

Shortcomings

- Computation of the Gram matrix G is quadratic in the number of parameters.
- Inversion of G is cubic

Common solutions are approximations through Kronecker factorization (**?**).

We will show that, in the context of deep-learning there is a far more efficient way to approximate !

A functionnal analysis perspective

Reinterpreting quadratic loss

Consider the loss of a classical quadratic regression problem, with (x_i) sampled from μ on Ω:

$$\ell(\theta) := \frac{1}{2S} \sum_{i=1}^{S} \left(u_\theta(x_i) - f(x_i)\right)^2$$

In the limit $S \to \infty$ (population limit), this loss can be reinterpreted as the evaluation on u_θ of the functional loss:

$$\mathcal{L}(u) := \frac{1}{2} \|u - f\|^2_{\mathrm{L}^2(\Omega,\mu)}$$

Taking the Fréchet derivative :

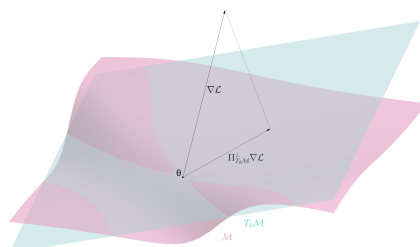
$$\mathrm{d}\mathcal{L}|_u(h) = \langle u - f\,,\, h\rangle_{\mathrm{L}^2(\Omega,\mu)} =: \langle \nabla\mathcal{L}\,,\, h\rangle_{\mathrm{L}^2(\Omega,\mu)}$$

Reinterpreting natural gradient

- $\mathcal{M} := \mathrm{Im}\, u = \{u_\theta \,:\, \theta \in \mathbb{R}^P\}$
- $T_\theta\mathcal{M} := \mathrm{Im}\, \mathrm{d}u_{|\theta} = \mathrm{Span}(\partial_p u_\theta)$

In the population limit, natural gradient can be reinterpreted as the update:

$$\theta_{t+1} \leftarrow \theta_t - \eta\, \mathrm{d}u^\dagger_{|\theta_t} \left(\Pi^\perp_{T_{\theta_t}\mathcal{M}} \nabla\mathcal{L}\right),$$



Reinterpreting PINNs loss

Consider the PINNs loss, with (x_i^D) sampled from μ on Ω and (x_i^B) sampled from σ on $\partial\Omega$:

$$\ell(\theta) := \frac{1}{2S_D}\sum_{i=1}^{S_D}\Big(D[u_\theta](x_i^D) - f(x_i^D)\Big)^2 + \frac{1}{2S_B}\sum_{i=1}^{S_B}\Big(B[u_\theta](x_i^B) - g(x_i^B)\Big)^2$$

In the limit $S \to \infty$ (population limit), this loss can be reinterpreted as the evaluation on u_θ of the functional loss:

$$\mathcal{L}(u) := \frac{1}{2}\|D[u] - f\|^2_{\mathrm{L}^2(\Omega,\mu)} + \frac{1}{2}\|B[u] - g\|^2_{\mathrm{L}^2(\partial\Omega,\sigma)}$$

PINNs natural gradient

Taking the Fréchet derivative :

$$\begin{aligned}\mathrm{d}\mathcal{L}_{|u}(h) &= \big\langle D[u] - f\,,\, \mathrm{d}D_{|u}(h)\big\rangle_{\mathrm{L}^2(\Omega,\mu)} \\ &\quad + \big\langle B[u] - g\,,\, \mathrm{d}B_{|u}(h)\big\rangle_{\mathrm{L}^2(\Omega,\mu)} \\ &= \big\langle \nabla\mathcal{L}\,,\, (\mathrm{d}D_{|u}(h), \mathrm{d}B_{|u}(h))\big\rangle_{\mathrm{L}^2(\Omega,\mu)\times\mathrm{L}^2(\partial\Omega,\sigma)} \\ &= \big\langle \nabla\mathcal{L}\,,\, \mathrm{d}(D,B)_{|u}(h)\big\rangle_{\mathrm{L}^2(\Omega,\mu)\times\mathrm{L}^2(\partial\Omega,\sigma)}\end{aligned}$$

- $\mathcal{M}_{D,B} := \mathrm{Im}(D,B) = \{(D[u], B[u]) \,:\, u \in \mathcal{H}\}$
- $T_u\mathcal{M}_{D,B} := \mathrm{Im}\,\mathrm{d}(D,B)_{|u}$

Gradient update:

$$u_{t+1} \leftarrow u_t - \eta\,\mathrm{d}(D,B)^\dagger_{|u_t}\left(\Pi^\perp_{T_{u_t}\mathcal{M}_{D,B}}\nabla\mathcal{L}\right),$$

PINNs natural gradient

Remark

PINNs are essentially a classical quadratic regression using the model:

$$(D, B) \circ u : \mathbb{R}^P \to \mathrm{L}^2(\Omega, \mu) \times \mathrm{L}^2(\partial\Omega, \sigma)$$

- $\Gamma := \mathrm{Im}(D, B) \circ u = \{(D[u_\theta], B[u_\theta]) : \theta \in \mathbb{R}^P\}$
- $T_\theta\Gamma := \mathrm{Im}\, \mathrm{d}\left((D, B) \circ u\right)_{|\theta} = \mathrm{Span}\left((\mathrm{d}|_{u_\theta} D[\partial_p u_\theta], \mathrm{d}|_{u_\theta} B[\partial_p u_\theta])\right)$
- $\nabla\mathcal{L} = (D[u_\theta] - f, B[u_\theta] - g)$

In the population limit, natural gradient of PINNs can be reinterpreted as the update:

$$\theta_{t+1} \leftarrow \theta_t - \eta \left(\mathrm{d}u_{|\theta_t} \circ \mathrm{d}(D, B)_{|u_{\theta_t}}\right)^\dagger \left(\Pi^\perp_{T_{\theta_t}\Gamma} \nabla\mathcal{L}\right)$$

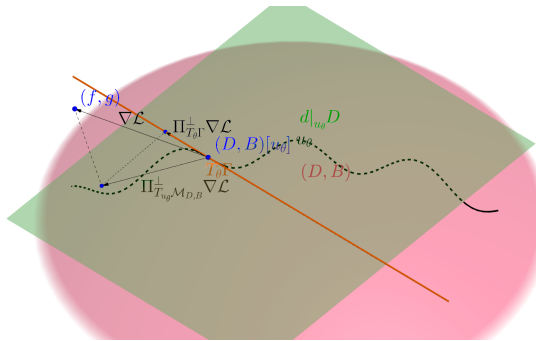


Figure: Natural Gradient of PINNs illustration

Neural Tangent Kernel (NTK)

NTK in a nutshell

Jacot et al. (2018) shows that for an empirical quadratic loss:

$$\ell(\theta) := \frac{1}{2}\sum_{i=1}^{N}(u_\theta(x_i) - y_i)^2$$

the functionnal dynamic of the gradient descent on ℓ can by described by:

$$\frac{\mathrm{d}u_{\theta_t}}{\mathrm{d}t}(x) = -\sum_{i=1}^{N} NTK(x, x_i)(u_\theta(x_i) - y_i),$$

with:

$$NTK(x, y) := \sum_{p=1}^{P} (\partial_p u_\theta(x))\,(\partial_p u_\theta(y))^T$$

Proposition

The functionnal dynamic of the natural gradient descent on:

$$\ell(\theta) := \sum_{i=1}^{N} \tau(u_\theta(x_i))$$

is described by (Rudner et al., 2019):

$$\frac{\mathrm{d}u_{\theta_t}}{\mathrm{d}t}(x) = -\sum_{i=1}^{N} NNTK(x, x_i)\tau'(u_\theta(x_i)),$$

with:

$$NNTK(x, y) := \sum_{1\leqslant p,q\leqslant P} (\partial_p u_\theta(x))\, G^{\dagger}_{pq}\, (\partial_p u_\theta(y))^T$$

Corollary

The empirical functional dynamics takes place in the subspace:

$\widetilde{T}_\theta\mathcal{M} := \text{Span}(NNTK(\cdot, x_i) \,:\, (x_i)_{1\leqslant i\leqslant N}) \subset T_\theta\mathcal{M}.$

Corollary

We can define an empirical natural gradient update by :

$$\theta_{t+1} = \theta_t - \eta \, \mathrm{d}u^\dagger_{|\theta_t} \left(\Pi^\perp_{\widetilde{T}_\theta\mathcal{M}} \nabla\mathcal{L} \right).$$

Corollary

There exist P points $(\hat{x}_i)_{1\leqslant i\leqslant P}$ such that the natural empirical dynamics matchs the population dynamics, *i.e.* such that:

$\Pi^\perp_{\widetilde{T}_\theta\mathcal{M}} \nabla\mathcal{L} = \Pi^\perp_{T_\theta\mathcal{M}} \nabla\mathcal{L}.$

Reproducing Kernel Hilbert Spaces (RKHS) *détour*

Proposition

An Hilbert space $\mathcal{H}$ of functions defined on a set $\Omega \to \mathbb{R}$ is a RKHS if and only if the following equivalent conditions are met:

1. The Identity operator I_d of $\mathcal{H}$ is a Hilbert-Schmidt operator.
2. There exist a function $k : \mathrm{L}^2(\Omega \times \Omega \to \mathbb{R})$ such that $\mathcal{H} = \overline{\mathrm{Span}\,(k(x, \cdot) : x \in \Omega)}$
3. for all $x \in \Omega$, the evaluation form $e_x : f \in \mathcal{H} \mapsto f(x)$ is continuous.

Corollary

Any finite dimensional Hilbert space is a RKHS

Proposition

If a RKHS $\mathcal{H}_0$ is isometrically embedded in a Hilbert space $\mathcal{H}_1$, then the orthogonal projection $\Pi_{\mathcal{H}_0}$ into $\mathcal{H}_0$ is a Hilbert-Schmidt operator.

Lemma

If $\mathcal{H}_0 := \overline{\mathrm{Span}(u_i : i \in \mathbb{N})} \subset \mathcal{H}$, then the kernel of $\Pi_{\mathcal{H}_0}$ is:

$$k(x, y) = \sum_{i \in \mathbb{N}} u_i G^{\dagger}_{i,j} u_j \qquad (1)$$

where $G_{ij} := \langle u_i \,,\, u_j \rangle_{\mathcal{H}}$.

Algorithmically efficient natural gradient

Theorem

Under mild assumptions, the empirical natural gradient update:

$$\theta_{t+1} = \theta_t - \eta \, \mathrm{d}u^{\dagger}_{|\theta_t} \left(\Pi^{\perp}_{\widetilde{T}_{\theta}\mathcal{M}} \nabla \mathcal{L} \right),$$

does not require to estimate a Gram matrix. More precisely, we have:

$$\mathrm{d}u^{\dagger}_{|\theta_t} \left(\Pi^{\perp}_{\widetilde{T}_{\theta}\mathcal{M}} \nabla \mathcal{L} \right) = \hat{\Phi}^{\dagger} \widehat{\nabla \mathcal{L}}, \text{where} \quad (2)$$

- $\forall 1 \leqslant p \leqslant P, \forall 1 \leqslant i \leqslant N, : \hat{\Phi}_{i,p} := \partial_p u_{\theta}(x_i)$
- $\forall 1 \leqslant i \leqslant N, : \widehat{\nabla \mathcal{L}}_i := \nabla \mathcal{L}(x_i)$

Remark

The pseudoinverse $\hat{\Phi}^{\dagger}$ can be computed with a SVD. In particular the complexity of the empirical natural gradient update is $O(\min(PN^2, P^2N))$, which has to be compared with:

- $O(PN)$ for classical gradient update.
- $O(P^3 + P^2M)$ with $M > N \log(N)$ (Gram estimation cost) for classical natural gradient update.

Since $\widetilde{T}_\theta\mathcal{M} = \mathrm{Span}(NNTK(\cdot, x_i) : (x_i)_{1\leqslant i\leqslant N}) \subset T_\theta\mathcal{M}$, the associated projection kernel is:

$$k(x, y) = NNTK(x, x_i)\hat{G}^\dagger_{i,j}NNTK(x_j, y)$$

where $\hat{G}$ is defined through: $\text{for all } 1 \leqslant i, j \leqslant N$

$$\hat{G}_{i,j} := \langle NNTK(x_i, \cdot)\,,\, NNTK(\cdot, x_j)\rangle = NNTK(x_i, x_j) = e_i^t\hat{\Phi}G^\dagger\hat{\Phi}^t e_j$$

$$\begin{aligned}\Pi^{\perp}_{\widetilde{T}_\theta \mathcal{M}} \nabla \mathcal{L} &= \sum_{i,j=1}^{N} NNTK(\cdot, x_i)\, \hat{G}^{\dagger}_{i,j} \langle NNTK(x_j, \cdot)\,,\, f \rangle \\ &= \sum_{p,q=1}^{P} \sum_{i,j=1}^{N} \partial_p u_{|\theta}\, G^{\dagger}_{p,q} \hat{\Phi}^{t}_{q,i} \hat{G}^{\dagger}_{i,j} \langle NNTK(x_j, \cdot)\,,\, f \rangle \\ &= \sum_{p,q=1}^{P} \sum_{i,j=1}^{N} \partial_p u_{|\theta}\, G^{\dagger}_{p,q} \hat{\Phi}^{t}_{q,i} \left(\hat{\Phi} G^{\dagger} \hat{\Phi}^{t} \right)^{\dagger}_{i,j} \langle NNTK(x_j, \cdot)\,,\, f \rangle\end{aligned}$$

Sketch of proof III

Let us write the SVD of $\hat{\Phi} = V\Delta U^t$. Then $\left(\hat{\Phi} G^\dagger \hat{\Phi}^t\right)^\dagger = V\Delta^\dagger U^t G U \Delta^\dagger V^t$ and thus:

$$\begin{aligned}\Pi^\perp_{\widetilde{T}_\theta \mathcal{M}} \nabla \mathcal{L} &= \sum_{p=1}^{P} \partial_p u_\theta e_p^t G^\dagger U \Delta V^t V \Delta^\dagger U^t G U \Delta^\dagger V^t \sum_{j=1}^{N} e_j \left\langle NNTK(x_j, \cdot)\,,\, f \right\rangle \\ &= \sum_{p=1}^{P} \partial_p u_\theta e_p^t U \left(U^t G U\right)^\dagger \begin{pmatrix} I_n & 0 \\ 0 & 0 \end{pmatrix} U^t G U \Delta^\dagger V^t \sum_{j=1}^{N} e_j \left\langle NNTK(x_j, \cdot)\,,\, f \right\rangle\end{aligned}$$

Let us write:

$$\tilde{G} := U^t G U = \begin{pmatrix} A & B \\ B^t & C \end{pmatrix},$$

where $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,P-n}$ and $C \in \mathbb{R}^{P-n,P-n}$. If we make the ansatz $B = 0$ and $f \in T_\theta \mathcal{M}$, the result holds.

Since PINNs are essentially a quadratic regression for the model $(D, B) \circ u$, we have :

Corollary

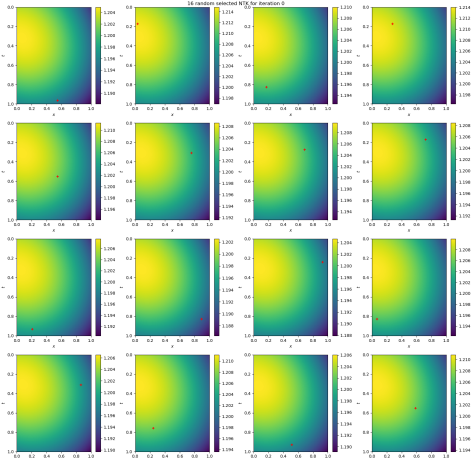
$$\left(\mathrm{d}u_{|\theta_t} \circ \mathrm{d}(D,B)_{|u_{\theta_t}}\right)^\dagger \left(\Pi^{\perp}_{\widetilde{T}_{\theta_t}\Gamma} \nabla \mathcal{L}\right) = \hat{\Phi}^\dagger \widehat{\nabla \mathcal{L}}, \text{where}$$

- $\widetilde{T}_{\theta_t}\Gamma := \mathrm{Span}(NNTK(\cdot, x_i) \,:\, (x_i)_{1 \leqslant i \leqslant N}) \subset T_\theta \mathcal{M}$
- $\forall 1 \leqslant p \leqslant P, \forall 1 \leqslant i \leqslant N, : \hat{\Phi}_{i,p} := \partial_p \left((D, B) \circ u\right)_{|\theta}(x_i)$
- $\forall 1 \leqslant i \leqslant N, : \widehat{\nabla \mathcal{L}}_i := \nabla \mathcal{L}(x_i)$

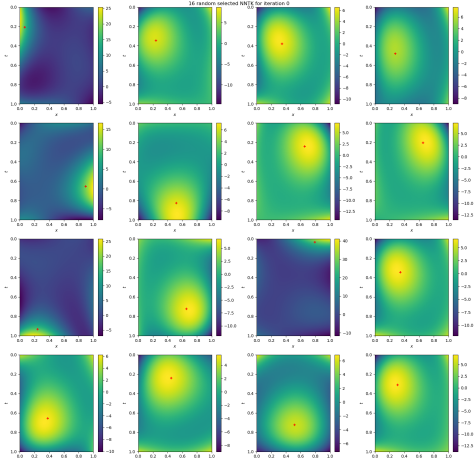
Corollary

The Green function of the operator on the space $u_\theta + \widetilde{T}_\theta \Gamma$ is given by:

$$g(x, y) := u_\theta(x) + \sum_{p=1}^{P} \sum_{i=1}^{N} \partial_p u_\theta(x) \hat{\Phi}^\dagger_{p,i} (NNTK(x_i, y) - D[u_\theta](x_i))$$

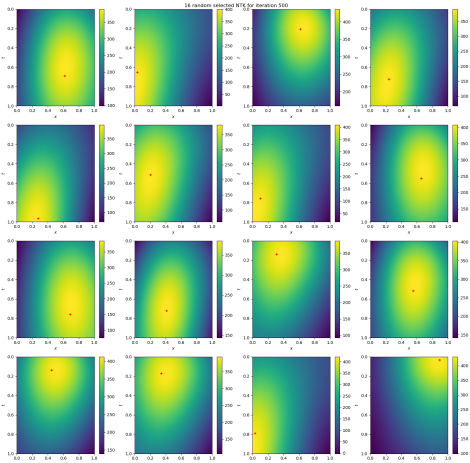


(a) NTK

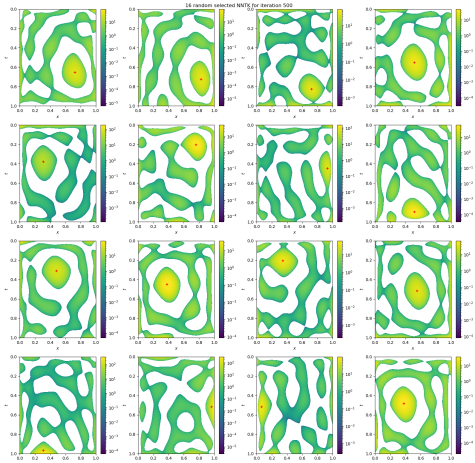


(b) NNTK

Figure: Comparison of NTK and NNTK at initialization for Heat equation in 1+1 D



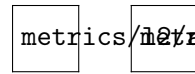
(a) NTK



(b) NNTK

Figure: Comparison of NTK and NNTK at the end of optimization for Heat equation in 1+1 D

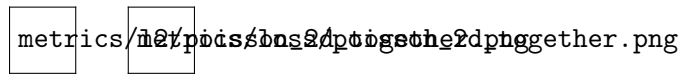
Experiments



(a) L^2 error of PINN solution (b) Test loss of PINN solution

Figure: PINN solution under Anagram optimization, to Laplace equation in 2 D:

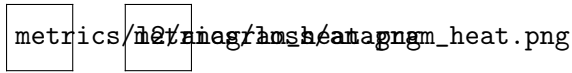
$$\begin{cases} \Delta u = -2\pi^2 \sin(\pi x_1) \sin(\pi x_2) & \text{in } [0,1]^2 \\ u = 0 & \text{on } \partial[0,1]^2 \end{cases}$$



(a) L^2 error of PINN solution

(b) Test loss of PINN solution

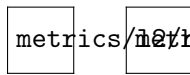
Figure: Performance comparison of Anagram and Adam optimization for Laplace equation in 2 D



(a) L^2 error of PINN solution (b) Test loss of PINN solution

Figure: PINN solution under Anagram optimization, to Heat equation in 1+1 D:

$$\begin{cases} \partial_t u - \frac{1}{4}\partial_{xx} u = 0 & \text{in } [0,1]^2 \\ u = 0 & \text{on } [0,1] \times \{0\} \bigcup [0,1] \times \{1\} \\ u = sin(\pi x) & \text{on } \{0\} \times [0,1] \end{cases}$$



(a) L^2 error of PINN solution

(b) Test loss of PINN solution

Figure: Performance comparison of Anagram and Adam optimization for Heat equation in 1+1 D

Conclusion and Perspectives

Conclusions

- Anagram gives a theoretically founded simplication to any natural-gradient algorithm lowering the complexity from $O(P^3 + P^2M)$, $M > N\log(N)$ to $O(\min(PN^2, P^2N))$, which is above stochastic gradient descent only by a factor $\min(P, N)$.
- In the case of PINNs, we prove that natural gradient correspond to an optimal linear update following the Green function.
- Empirical results are improved by several orders of magnitude.
- The SVD cut-off factor appears to be a pivotal hyper-parameter of the algorithm.

Perspectives

- Design of an optimal collocation points procedure, coupled with SVD cut-off factor adaptation strategy
- Establish theoretical connections with classical algorithms, such as FEMs
- Include in this theoretical setting the data assmilation, and understand its regularizing effect

Next pivotal challenge is the design of an efficient algorithm for non-linear PDEs

Thank you for your attention !

References I

Amari, S.-I. and S. C. Douglas (1998): "Why Natural Gradient?" in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, IEEE, vol. 2, 1213–1216.

Jacot, A., F. Gabriel, and C. Hongler (2018): "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," *Advances in neural information processing systems*, 31.

Lagaris, I. E., A. Likas, and D. I. Fotiadis (1998): "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations," *IEEE transactions on neural networks*, 9, 987–1000.

Leshno, M., V. Y. Lin, A. Pinkus, and S. Schocken (1993): "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural networks*, 6, 861–867.

References II

Martens, J. and R. Grosse (2020): "Optimizing Neural Networks with Kronecker-factored Approximate Curvature," .

Raissi, M., P. Perdikaris, and G. Karniadakis (2019): "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.

Rudner, T. G., F. Wenzel, Y. W. Teh, and Y. Gal (2019): "The Natural Neural Tangent Kernel: Neural Network Training Dynamics under Natural Gradient Descent," in *4th Workshop on Bayesian Deep Learning (NeurIPS 2019)*.