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

TAU Seminar : LISN—Paris-Saclay University, CNRS, INRIA

Nilo Schwencke

September 24th, 2024

Outline

Physics informed neural networks (PINNs)

Neural networks

Let $\mathcal{H}$ be a Hilbert space coumpound of functions $\Omega \to \mathbb{R}$.

Definition

A neural network is a differentiable non-linear functional

$$u : \begin{cases} \mathbb{R}^P & \to & \mathcal{H} \\ \theta & \mapsto & u_{|\theta} \end{cases}$$

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

PINNs in a nutshell

How to approximate such a solution?

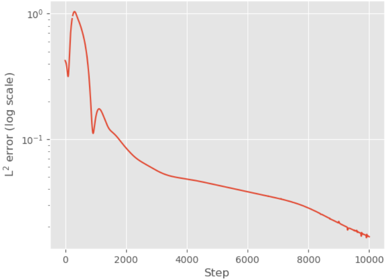
Answer: Just as we would for any neural network, *i.e.* by gradient descent (Lagaris et al., 1998; Raissi et al., 2019). More precisely, given the PDE:

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

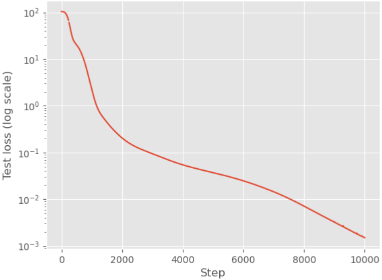
we will optimize the loss:

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

PINNs shortcomings and drawbacks



(a) L^2 error of PINN solution

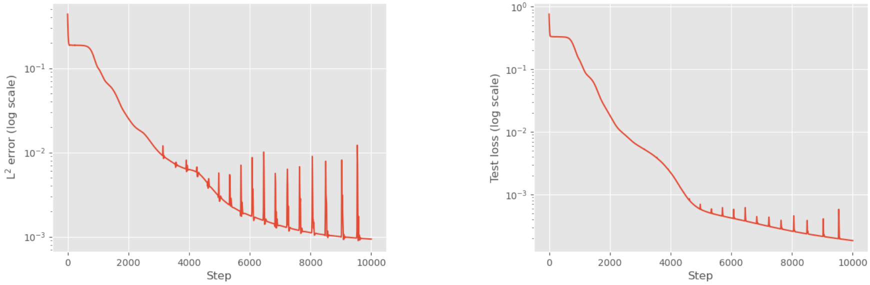


(b) Test loss of PINN solution

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:

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

with G_{θ_t} the Hessian of the Kullback-Leibler divergence. More generally in the context of Riemannian manifolds:

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

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

Shortcomings

- Computation of the Gram matrix G_{θ_t} is quadratic in the number of parameters.
- Inversion of G_{θ_t} is cubic

Common solutions are approximations through Kronecker factorization (Martens and Grosse, 2015).

We will introduce a new kind of natural gradient that scales linearly with the number of parameters.

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:

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

Taking the Fréchet derivative :

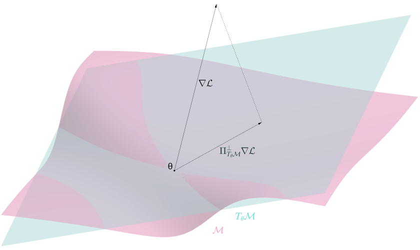
$$\mathrm{d}\mathcal{L}|_u(h) = \langle u - f\,,\, h\rangle_{\mathrm{L}^2(\Omega,\mu)} =: \left\langle \nabla\mathcal{L}_{|u}\,,\, h\right\rangle_{\mathrm{L}^2(\Omega,\mu)}$$

Reinterpreting natural gradient

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

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

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



Neural Tangent Kernel (NTK)

NTK in a nutshell

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

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

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

$$\frac{\mathrm{d}u_{\theta_t}}{\mathrm{d}t}(x) = -\sum_{i=1}^{S} NTK_{\theta_t}(x, x_i)(u_{|\theta_t}(x_i) - y_i),$$

with:

$$NTK_{\theta_t}(x, y) := \sum_{p=1}^{P} \left(\partial_p u_{|\theta_t}(x)\right) \left(\partial_p u_{|\theta_t}(y)\right)^T$$

Proposition

The functionnal dynamic of the natural gradient descent on:

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

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

$$\frac{du_{\theta_t}}{dt}(x) = -\sum_{i=1}^{S} NNTK_{\theta_t}(x, x_i)\tau'(u_{\theta_t}(x_i)),$$

with: $NNTK_{\theta_t}(x, y) :=$

$$\sum_{1\leqslant p,q\leqslant P} \left(\partial_p u_{|\theta_t}(x)\right) G_{\theta_t}{}^{\dagger}_{pq} \left(\partial_q u_{|\theta_t}(y)\right)^T$$

Corollary

The empirical functional dynamics takes place in the empirical tangent space:

$$\hat{T}_\theta \mathcal{M} := \mathrm{Span}\left(NNTK_\theta(x_i, \cdot)\ :\ (x_i)_{1 \leqslant i \leqslant N}\right) \subset T_\theta \mathcal{M}.$$

Corollary

We can define an empirical natural gradient update by:

$$\theta_{t+1} = \theta_t - \eta\, du^{\dagger}_{|\theta_t} \left(\Pi^{\perp}_{\hat{T}_{\theta_t}\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta_t}}\right).$$

empirical Natural Gradient (eNG)

Reproducing Kernel Hilbert Spaces (RKHS) *détour*

Lemma
Any finite dimensional space $\mathcal{H}$ is a RKHS

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: for all $x, y \in \Omega$

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

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

Corollary
The kernel of $\Pi_{T_\theta\mathcal{M}}$ is: for all $x, y \in \Omega$

$$k(x,y) = \sum_{1\leqslant p,q\leqslant P} \partial_p u_{|\theta}(x) G_\theta{}^{\dagger}_{p,q} \partial_q u_{|\theta}(y),$$

where $G_{p,q} := \left\langle \partial_p u_{|\theta} \,,\, \partial_q u_{|\theta} \right\rangle_{\mathcal{H}}$.

Remark
$NNTK_\theta$ is the kernel of $\Pi_{T_\theta\mathcal{M}}$. In particular $NNTK_\theta$ is the reproducing kernel of $T_\theta\mathcal{M}$.

Corollary
The kernel of $\Pi_{\hat{T}_\theta\mathcal{M}}$ is: for all $x, y \in \Omega$

$$\hat{k}(x,y) = \sum_{1\leqslant i,j\leqslant S} NNTK_\theta(x,x_i) \hat{G}_\theta{}^{\dagger}_{i,j} NNTK_\theta(x_j,y),$$

where:

$$G_{i,j} := \langle NNTK_\theta(\cdot,x_i) \,,\, NNTK_\theta(x_j,\cdot) \rangle_{\mathcal{H}} = NNTK_\theta(x_i,x_j)$$

An empirical Natural Gradient computation : ANaGRAM

Theorem (ANaGRAM)

Under mild assumptions, the empirical natural gradient update:

$$\theta_{t+1} = \theta_t - \eta \, du^{\dagger}_{|\theta_t} \left(\Pi^{\perp}_{\hat{T}_{\theta_t}\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta_t}} \right),$$

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

$$du^{\dagger}_{|\theta_t} \left(\Pi^{\perp}_{\hat{T}_{\theta_t}\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta_t}} \right) = \hat{\phi}^{\dagger}_{\theta_t} \widehat{\nabla \mathcal{L}}_{\theta_t},$$

where: for all $1 \leqslant p \leqslant P, 1 \leqslant i \leqslant S$

- $\widehat{\phi}_{\theta_t\, i,p} := \partial_p u_{|\theta_t}(x_i)$
- $\widehat{\nabla \mathcal{L}}_{\theta_t\, i} := \nabla \mathcal{L}_{|u_{|\theta_t}}(x_i)$

Remark

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

- $O(PS)$ for classical gradient update.
- $O(P^3)$ for classical natural gradient update.

Corollary

There exist P points $(\hat{x}_i)$ such that:

$$\Pi^{\perp}_{\hat{T}_{\theta}\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta}} = \Pi^{\perp}_{T_{\theta}\mathcal{M}} \nabla \mathcal{L}_{|u_{|\theta}}.$$

Application to PINNs

PINNs are a quadradic regression problem

The only difference between the losses:

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

and:

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

is the use of the differential operator D and the boundary operator B.

Proposition

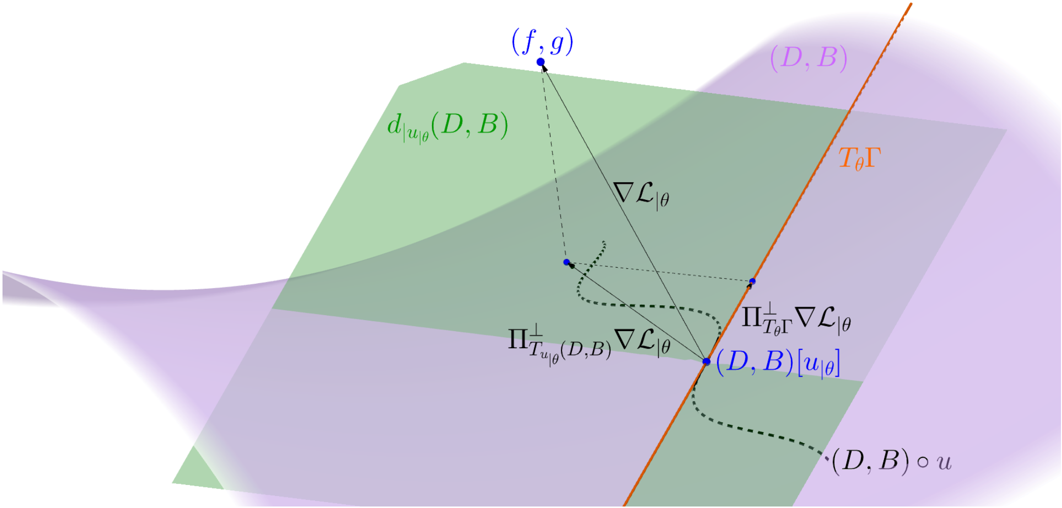
PINNs are a classical quadradic regression problem with model:

$$(D,B)\circ u : \begin{cases} \mathbb{R}^P & \rightarrow & \mathcal{H} & \rightarrow & L^2(\Omega \to \mathbb{R}, \mu) \times L^2(\partial\Omega \to \mathbb{R}, \sigma) \\ \theta & \mapsto & u_{|\theta} & \mapsto & \left(D\left[u_{|\theta}\right], B\left[u_{|\theta}\right]\right) \end{cases}$$

In the same way:

- $\Gamma := \operatorname{Im}\big((D,B)\circ u\big) = \left\{\left(D\left[u_{|\theta}\right], B\left[u_{|\theta}\right]\right) : \theta \in \mathbb{R}^P\right\}$
- $T_\theta\Gamma := \operatorname{Im} \mathrm{d}\big((D,B)\circ u\big)_{|\theta} = \operatorname{Span}\left(\left(\mathrm{d}_{|u_{|\theta}} D\left[\partial_p u_{|\theta}\right], \mathrm{d}_{|u_{|\theta}} B\left[\partial_p u_{|\theta}\right]\right)_{p=1}^{P}\right)$
- $\nabla\mathcal{L}_{|u_{|\theta}} = \left(D\left[u_{|\theta}\right] - f, B\left[u_{|\theta}\right] - g\right)$

Natural Gradient for PINNs



In the population limit, the natural gradient of PINNs is the update:

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

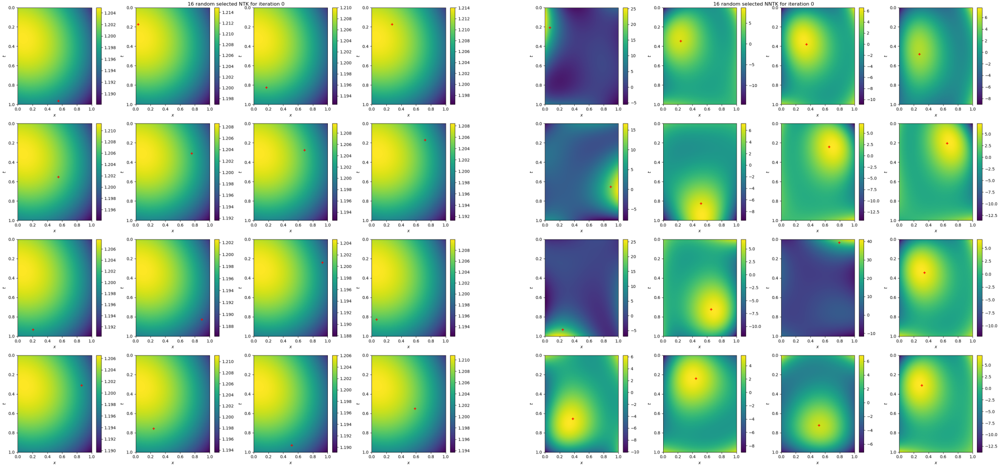
Corollary

The kernel of $\Pi_{T_\theta \Gamma}$ *is: for all* $x, y \in (\Omega \times \partial\Omega)^2$

$$\begin{aligned} NNTK_\theta(x,y) &= \sum_{1\leqslant p,q\leqslant P} \partial_p(D,B)\left[u_{|\theta}\right](x)\, {G_\theta}^{\dagger}_{p,q} \partial_q(D,B)\left[u_{|\theta}\right](y) \\ &= \sum_{1\leqslant p,q\leqslant P} \left(\partial_p D\left[u_{|\theta}\right](x_1), \partial_p B\left[u_{|\theta}\right](x_2)\right) {G_\theta}^{\dagger}_{p,q} \left(\partial_q D\left[u_{|\theta}\right](y_1), \partial_q B\left[u_{|\theta}\right](y_2)\right), \end{aligned}$$

where for all $1 \leqslant p, q \leqslant P$

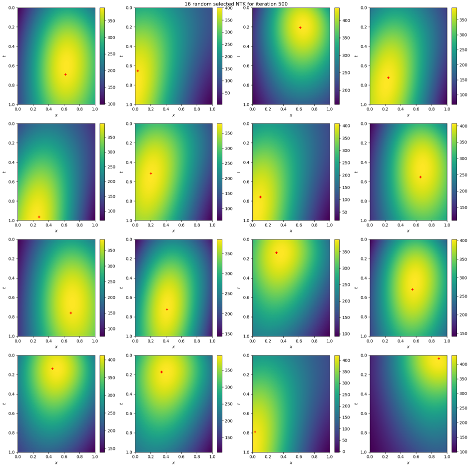
$$\begin{aligned} {G_\theta}_{p,q} &:= \left\langle \partial_p(D,B)\left[u_{|\theta}\right], \partial_q(D,B)\left[u_{|\theta}\right]\right\rangle_{L^2(\Omega\to\mathbb{R},\mu)\times L^2(\partial\Omega\to\mathbb{R},\sigma)} \\ &= \left\langle \partial_p D\left[u_{|\theta}\right], \partial_q D\left[u_{|\theta}\right]\right\rangle_{L^2(\Omega\to\mathbb{R},\mu)} + \left\langle \partial_p B\left[u_{|\theta}\right], \partial_q B\left[u_{|\theta}\right]\right\rangle_{L^2(\partial\Omega\to\mathbb{R},\sigma)}. \end{aligned}$$



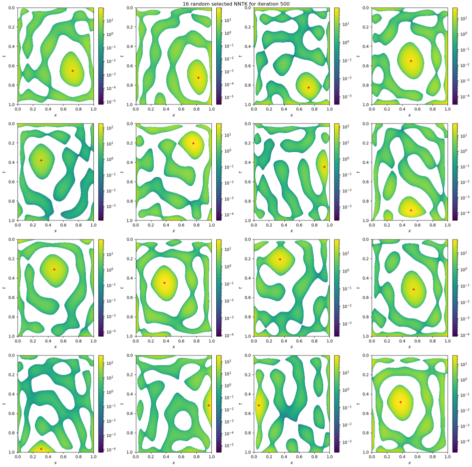
(a) NTK (b) NNTK

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

Natural Gradient for PINNs



(a) NTK



(b) NNTK

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

empirical Natural Gradient and ANaGRAM for PINNs

Corollary

The kernel of $\Pi_{\hat{T}_\theta \Gamma}$ *is: for all* $x, y \in (\Omega \times \partial\Omega)^2$

$$\hat{k}(x, y) = \sum_{1 \leqslant i,j \leqslant S} NNTK_\theta(x, x_i)\, \hat{G}_{\theta\, i,j}^{\dagger} NNTK_\theta(x_j, y), \text{ where}$$

$$G_{\theta\, i,j} := \langle NNTK_\theta(\cdot, x_i)\,,\, NNTK_\theta(x_j, \cdot)\rangle_{L^2(\Omega \to \mathbb{R}, \mu) \times L^2(\partial\Omega \to \mathbb{R}, \sigma)} = NNTK_\theta(x_i, x_j)$$

Theorem (ANaGRAM for PINNs)

Under mild assumptions, the empirical natural gradient update:

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

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

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

where: for all $1 \leqslant p \leqslant P, 1 \leqslant i \leqslant S$

- $\hat{\phi}_{\theta_t\, i,p} := \left(\partial_p D\left[u_{|\theta_t}\right](x_{i1}), \partial_p B\left[u_{|\theta_t}\right](x_{i2})\right)$
- $\widehat{\nabla \mathcal{L}}_{\theta_t\, i} := \nabla \mathcal{L}_{|u_{|\theta_t}}(x_i)$

Definition (Green's function of D)

A Green's function is any kernel function $g : \Omega \times \Omega \to \mathbb{R}$ such that the operator:

$$R : f \in D\left[\mathcal{H}\right] \mapsto \left(x \in \Omega \mapsto \int_\Omega g(x,s) f(s) \mu(\mathrm{d}s)\right) \in \mathcal{H}$$

verifies the equation: $D \circ R = I_{D[\mathcal{H}]}$

Definition (generalized Green's function of D on $\mathcal{H}_0 \subset \mathcal{H}$)

A generalized Green's function is any kernel function $g : \Omega \times \Omega \to \mathbb{R}$ such that the operator:

$$R : f \in \mathrm{L}^2(\Omega \to \mathbb{R}, \mu) \mapsto \left(x \in \Omega \mapsto \int_\Omega g(x,s) f(s) \mu(\mathrm{d}s)\right) \in \mathcal{H}$$

verifies the equation: $D \circ R = \Pi^{\perp}_{D[\mathcal{H}_0]}$

Proposition

The Green's function of the operator D on the space $u_\theta + T_\theta\Gamma$ is given by:

$$g(x,y) := u_{|\theta}(x) + \sum_{1\leqslant p,q\leqslant P} \partial_p u_{|\theta}(x) G^{\dagger}_{\theta\,p,q} \left(\partial_q D\left[u_{|\theta}\right](y) - \frac{1}{2}\int_{\Omega} \partial_q D\left[u_{|\theta}\right]^2(s)\mu(ds) \right)$$

Proposition

The Green's function of the operator D on the space $u_\theta + \widehat{T}_\theta\Gamma$ is given by:

$$g(x,y) := u_{|\theta}(x) + \sum_{p=1}^{P}\sum_{i=1}^{S} \partial_p u_{|\theta}(x) \widehat{\phi}^{\dagger}_{\theta\,p,i} \left(NNTK_\theta(x_i,y) - D\left[u_{|\theta}\right](x_i) \right)$$

Natural Gradient and Green's function

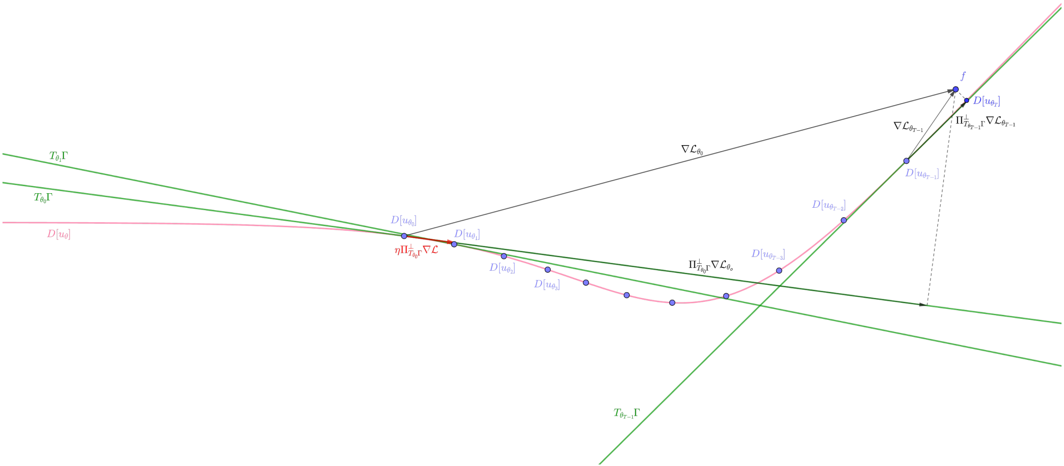
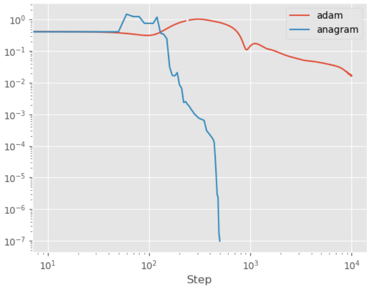
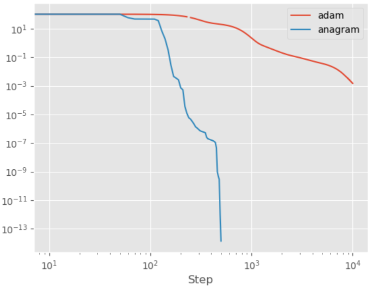


Figure: Illustration of PINNs learning process under natural gradient, as successive applications of Green's function

Experiments



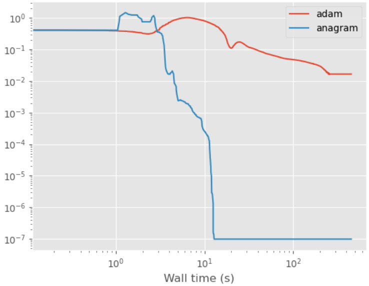
(a) L^2 error of PINN solution



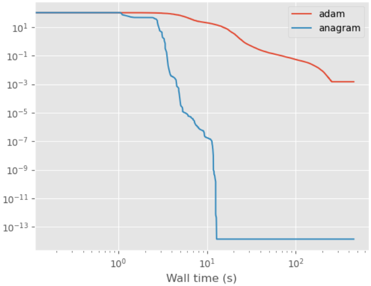
(b) Test loss of PINN solution

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

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



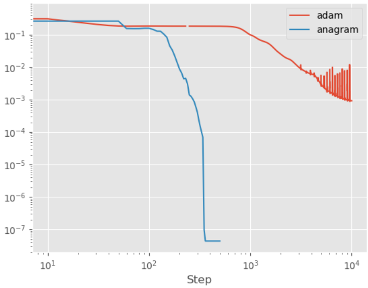
(a) L^2 error of PINN solution



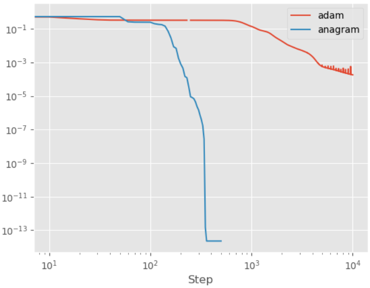
(b) Test loss of PINN solution

Figure: Time-performance comparison of Anagram and Adam optimization for Laplace equation in 2 D:

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



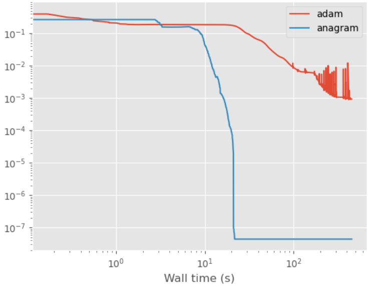
(a) L^2 error of PINN solution



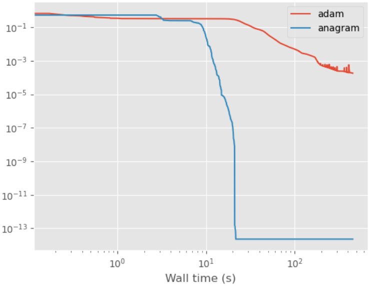
(b) Test loss of PINN solution

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

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



(a) L^2 error of PINN solution

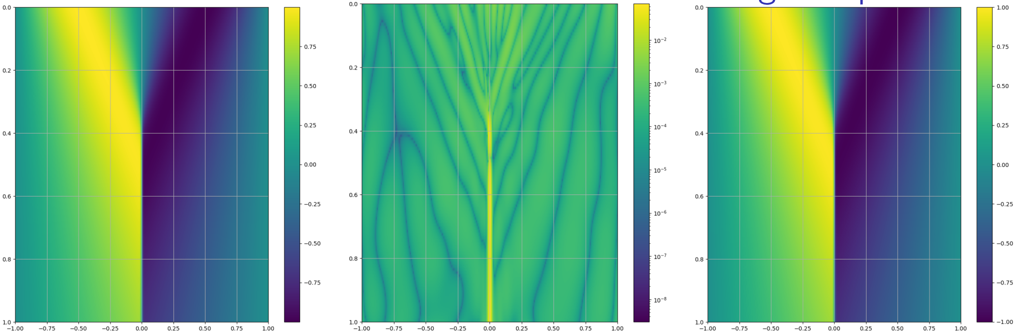


(b) Test loss of PINN solution

Figure: Time-performance comparison of Anagram and Adam optimization for Heat equation in 1+1 D:

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

Burgers equation

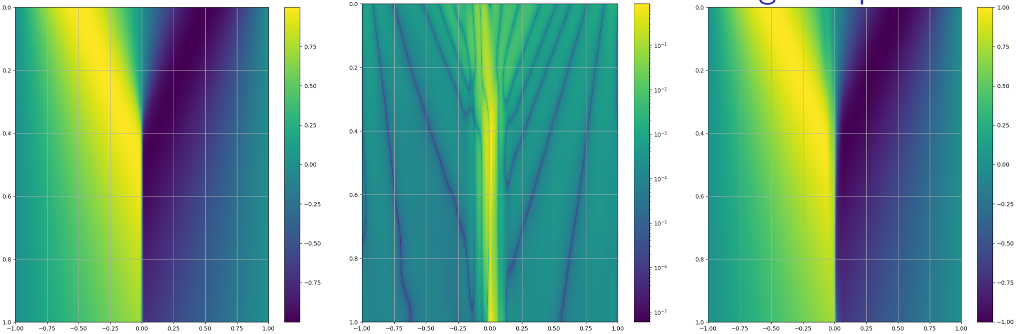


(a) True solution profile (b) L^2 error of PINN solution (c) PINN's solution profile

Figure: PINN's solution under Adam (15k steps) + L-BFGS (15k) steps for Burgers equation in 1+1 D:

$$\begin{cases} \partial_t u + u\partial_x u = \nu \partial_{xx} u & \text{in } [0,1] \times [-1,1] \\ u = 0 & \text{on } [0,1] \times \{-1,1\} \\ u = -sin(\pi x) & \text{on } \{0\} \times [-1,1] \end{cases}$$

Burgers equation



(a) True solution profile (b) L^2 error of PINN solution (c) PINN's solution profile

Figure: PINN's solution under Anagram (500 steps) for Burgers equation in 1+1 D:

$$\begin{cases} \partial_t u + u\partial_x u = \nu \partial_{xx} u & \text{in } [0,1] \times [-1,1] \\ u = 0 & \text{on } [0,1] \times \{-1,1\} \\ u = -sin(\pi x) & \text{on } \{0\} \times [-1,1] \end{cases}$$

Conclusion and Perspectives

Conclusions

- Anagram gives a theoretically founded simplication to any natural-gradient algorithm lowering the complexity from $O(P^3)$ 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's 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, FDMs, *etc.*
- Include data assmilation in this theoretical setting, and understand its regularizing effect.
- Include common optimization techniques (*e.g.* Momentum)
- Extend to order 2 methods

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 (2015): "Optimizing Neural Networks with Kronecker-factored Approximate Curvature," in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, 2408–2417.

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)*.