PINNs and GaLS: A Priori Error Estimates for Shallow Physics Informed Neural Networks

The Third Conference of Young Applied Mathematicians

G. Turkiyyah, S. Zampini, U. Zerbinati

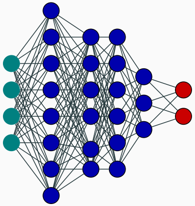
September 21, 2023

Neural Networks

Let us fix an **activation function** $\sigma : \mathbb{R} \to \mathbb{R}$, we extend it component wise to,

$$\sigma : \mathbb{R}^n \to \mathbb{R}^n$$
$$\mathbf{x} \mapsto (\sigma(x_1), \dots, \sigma(x_n))$$

Now we fix $\mathbf{n} \in \mathbb{N}^{(d+2)}$ and we say that a function $f : \mathbb{R}^{n_1} \to \mathbb{R}^{n_{d+2}}$ is a neural network if,



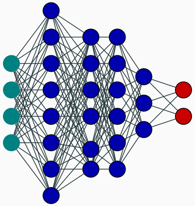
$$f(\mathbf{x}) = W_d \left(\sigma \left(W_{d-1} \dots \sigma \left(W_0 \mathbf{x} + \mathbf{b}_0\right) + \mathbf{b}_{d-1}\right) + \mathbf{b}_d\right).$$

Neural Networks

Let us fix an **activation function** $\sigma : \mathbb{R} \to \mathbb{R}$, we extend it component wise to,

$$\sigma : \mathbb{R}^n \to \mathbb{R}^n$$
$$\mathbf{x} \mapsto (\sigma(x_1), \ldots, \sigma(x_n))$$

Now we fix $\mathbf{n} \in \mathbb{N}^{(d+2)}$ and we say that a function $f : \mathbb{R}^{n_1} \to \mathbb{R}^{n_{d+2}}$ is a neural network if,



$$f(\mathbf{x}) = W_d \left(\sigma \left(W_{d-1} \ldots \sigma \left(W_0 \mathbf{x} + \mathbf{b}_0 \right) + \mathbf{b}_{d-1} \right) + \mathbf{b}_d \right).$$

We say a neural network is **shallow** if $d = 1$ while we say that a neural network is deep if $d > 1$. We will write DNN_N to denote a shallow neural network with N neurons in its hidden layer.

Abstract PDE

- Let X, Y be two Hilbert space, equipped respectively with the scalar products $(\cdot, \cdot)_X$ and $(\cdot, \cdot)_Y$.

Abstract PDE

- Let X, Y be two Hilbert space, equipped respectively with the scalar products $(\cdot, \cdot)_X$ and $(\cdot, \cdot)_Y$.
- Let $Q \in \mathcal{L}(X, Y)$ be a Fredholm differential operator, in particular $\dim \Big(N(Q) \Big) = 0$ where $N(Q)$ is the null space of Q.

Abstract PDE

- Let X, Y be two Hilbert space, equipped respectively with the scalar products $(\cdot,\cdot)_X$ and $(\cdot,\cdot)_Y$.
- Let $Q \in \mathcal{L}(X, Y)$ be a Fredholm differential operator, in particular $\dim\Big(N(Q)\Big) = 0$ where $N(Q)$ is the null space of Q.

$$\boxed{\textit{Given } F \in Y \textit{ find } u \in X \textit{ such that } Qu = F \in Y.}$$

Abstract PDE

- Let X, Y be two Hilbert space, equipped respectively with the scalar products $(\cdot,\cdot)_X$ and $(\cdot,\cdot)_Y$.
- Let $Q \in \mathcal{L}(X, Y)$ be a Fredholm differential operator, in particular $\dim\Big(N(Q)\Big) = 0$ where $N(Q)$ is the null space of Q.

$$\boxed{\textit{Given } F \in Y \textit{ find } u \in X \textit{ such that } Qu = F \in Y.}$$

The Poisson Problem

$$\boxed{\textit{Given } F \in Y \textit{ find } u \in X \textit{ such that } \Delta u = F \in Y.}$$

Abstract PDE

- Let X, Y be two Hilbert space, equipped respectively with the scalar products $(\cdot,\cdot)_X$ and $(\cdot,\cdot)_Y$.
- Let $Q \in \mathcal{L}(X,Y)$ be a Fredholm differential operator, in particular $\dim\Big(N(Q)\Big) = 0$ where $N(Q)$ is the null space of Q.

$$\boxed{\textit{Given } F \in Y \textit{ find } u \in X \textit{ such that } Qu = F \in Y.}$$

The Poisson Problem

$$\boxed{\textit{Given } F \in Y \textit{ find } u \in X \textit{ such that } \Delta u = F \in Y.}$$

Variational Formulation – Poisson Problem

$$\boxed{\textit{Given } f \in L^2(\Omega) \textit{ find } u \in H_0^1(\Omega) \textit{ such that } \int_\Omega \nabla u \nabla v = \int_\Omega fv,\ \forall v \in H_0^1(\Omega).}$$

Abstract PDE

- Let X, Y be two Hilbert space, equipped respectively with the scalar products $(\cdot,\cdot)_X$ and $(\cdot,\cdot)_Y$.
- Let $Q \in \mathcal{L}(X, Y)$ be a Fredholm differential operator, in particular $\dim\Big(N(Q)\Big) = 0$ where $N(Q)$ is the null space of Q.

$$\boxed{\textit{Given } F \in Y \textit{ find } u \in X \textit{ such that } Qu = F \in Y.}$$

The Poisson Problem

$$\boxed{\textit{Given } F \in Y \textit{ find } u \in X \textit{ such that } \Delta u = F \in Y.}$$

Variational Formulation – Poisson Problem

$$\boxed{\textit{Given } f \in L^2(\Omega) \textit{ find } u \in H^1_0(\Omega) \textit{ such that } \int_\Omega \nabla u \nabla v = \int_\Omega fv,\ \forall v \in H^1_0(\Omega).}$$

$$\boxed{\textit{Given } f \in L^2(\Omega) \textit{ find } u \in H^2_0(\Omega) \textit{ such that } \int_\Omega \Delta u \Delta v = \int_\Omega f \Delta v,\ \forall v \in H^2_0(\Omega).}$$

Energy Minimisation Aspects – Critical Action Principle

There exists a fundamental connection between PDE and energy critical points, in particular **minimisers** if the Lagrangian is convex.

Energy Minimisation Aspects – Critical Action Principle

There exists a fundamental connection between PDE and energy critical points, in particular **minimisers** if the Lagrangian is convex.

$$u = \underset{v \in X}{argmin}\, J(v, F) := \frac{1}{2} \int_{\Omega} |\nabla v|^2 - \int_{\Omega} fv.$$

Energy Minimisation Aspects – Critical Action Principle

There exists a fundamental connection between PDE and energy critical points, in particular **minimisers** if the Lagrangian is convex.

$$u = \underset{v \in X}{argmin}\, J(v, F) := \frac{1}{2}\int_\Omega |\nabla v|^2 - \int_\Omega fv.$$

$$\begin{aligned} u &= \underset{v \in X}{argmin}\, \|Qv - F\|_Y^2 = \int_\Omega |\Delta v - f|^2, \\ &= \underset{v \in X}{argmin}\, J(v, F) := \frac{1}{2}\int_\Omega |\Delta v|^2 - \int_\Omega f\,\Delta v. \end{aligned}$$

Galerkin Discretisation

- Let X_h be the finite dimensional discretisation space DNN_N.

Galerkin Discretisation

- Let X_h be the finite dimensional discretisation space DNN_N.
- Notice that ours choice of discretisation space respects conformity, i.e. $\boxed{X_h \subset X}$.

Galerkin Discretisation

- Let X_h be the finite dimensional discretisation space DNN_N.
- Notice that ours choice of discretisation space respects conformity, i.e. $\boxed{X_h \subset X}$.
- Consider the discrete problem,

$$\boxed{\textit{Find } u_h \in X_h \textit{ such that } Qu_h = F_h \in Y_h,}$$

where F_h is the projection of F on $Q(X_h)$ by the scalar product of Y.

Galerkin Discretisation

- Let X_h be the finite dimensional discretisation space DNN_N.
- Notice that ours choice of discretisation space respects conformity, i.e. $\boxed{X_h \subset X}$.
- Consider the discrete problem,

$$\boxed{\textit{Find } u_h \in X_h \textit{ such that } Qu_h = F_h \in Y_h,}$$

where F_h is the projection of F on $Q(X_h)$ by the scalar product of Y.

- In particular we focus our attention on the corresponding energy minimisation,

$$\boxed{u_h = \underset{v_h \in X_h}{argmin} \, \|Qv_h - F_h\|_Y^2 = \int_\Omega |\Delta v_h - f|^2.}$$

PINNs – Montecarlo Quadrature

If we use as a quadrature formula for the energy a **Montecarlo** scheme the discrete energy becomes,

$$J^h(u,f) = \frac{1}{N_\Omega^{\gamma_1}} \sum_i^N |\Delta u(\mathbf{x_i}) - f(x_i)|^2 + \frac{1}{N_{\partial\Omega}^{\gamma_2}} \sum_i^N |u(\mathbf{b_i})|^2$$

$$\mathbf{x_i} \sim \mathcal{U}(\Omega) \qquad \mathbf{b_i} \sim \mathcal{U}(\partial\Omega) \qquad \Omega = [0,1]^2$$

[1] At least shallow ones.

PINNs – Montecarlo Quadrature

If we use as a quadrature formula for the energy a **Montecarlo** scheme the discrete energy becomes,

$$J^h(u,f) = \frac{1}{N_\Omega^{\gamma_1}} \sum_i^N |\Delta u(\mathbf{x_i}) - f(x_i)|^2 + \frac{1}{N_{\partial\Omega}^{\gamma_2}} \sum_i^N |u(\mathbf{b_i})|^2$$

$$\mathbf{x_i} \sim \mathcal{U}(\Omega) \qquad \mathbf{b_i} \sim \mathcal{U}(\partial\Omega) \qquad \Omega = [0,1]^2$$

Searching for the minimizer of $J^h(u,f)$ in the space $X_h = DNN_N$ are precisely **PINNs**[1].

[1] At least shallow ones.

A Priori Error Estimates

Considering a conforming Galerkin approximation, including PINNs, for an abstract PDE we are interested in an a priori error estimate for,

$$\left\| u - \widetilde{u}_h \right\|_X$$

where u is the exact solution of the PDE and $\widetilde{u}_h$ is the computed solution and u_h the exact discrete solution.

A Priori Error Estimates

Considering a conforming Galerkin approximation, including PINNs, for an abstract PDE we are interested in an a priori error estimate for,

$$\left\| u - \widetilde{u}_h \right\|_X$$

where u is the exact solution of the PDE and $\widetilde{u}_h$ is the computed solution and u_h the exact discrete solution.

We begin splitting the error as follows,

$$\boxed{\left\| u - \widetilde{u}_h \right\|_X} = \boxed{\left\| u - u_h \right\|_X} + \boxed{\left\| u_h - \widetilde{u}_h \right\|_X}.$$

A Priori Error Estimates – LSQFEM

The main idea here presented is to use idea from **least square finite elements** in order to obtain an a priori error estimate for,

$$\boxed{\|u - u_h\|_X}.$$

A Priori Error Estimates – LSQFEM

The main idea here presented is to use idea from **least square finite elements** in order to obtain an a priori error estimate for,

$$\boxed{\|u - u_h\|_X}.$$

In order to do this we need to make some hypothesis on the discrete scalar product $(\cdot,\cdot)_h$ used which produce the discrete energy functional,

$$(\cdot,\cdot)_h : X_h \times X_h \to \mathbb{R}$$

$$(u_h, v_h)_h \mapsto \frac{1}{N_\Omega^{\gamma_1}} \sum_{i=1}^{N_\Omega} u_h(\omega_i) v_h(\omega_i) + \frac{1}{N_{\partial\Omega}^{\gamma_2}} \sum_{i=1}^{N_{\partial\Omega}} u_h(\beta_i) v_h(\beta_i).$$

A Priori Error Estimates – Assumptions

As we notice before the discrete lost function is given by,

$$J^h(u; F) = \|Qu - F\|_h^2, \qquad u_h = \arg\min_{v \in X_h} J^h(v; F).$$

A Priori Error Estimates – Assumptions

As we notice before the discrete lost function is given by,

$$J^h(u;F) = \|Qu - F\|_h^2, \qquad u_h = \operatorname*{arg\,min}_{v \in X_h} J^h(v;F).$$

Assumption 1

We will further assume that two positive semi-definite bilinear form $e(\cdot,\cdot)$ and $\epsilon(\cdot,\cdot)$ exist, such that,

$$J^h(u_h, F) = \frac{1}{2}\bigg((u_h, u_h)_h + (u,u)_h + \epsilon(u,u)\bigg) - (u, u_h)_h$$
$$-e(u, u_h) \qquad \forall u \in X, \ \forall u_h \in X_h,$$

A Priori Error Estimates – Assumptions

Assumption 2

Fixed $u_h \in X_h$ it exists a set of points $\left\{\delta_1, \ldots, \delta_{NDOF}\right\}$ such that, $u_h(\delta_1) = \cdots = u_h(\delta_{NDOF}) = 0 \Leftrightarrow u_h \equiv 0$.

Furthermore we require that $\{\delta_i\}_{i=1}^{NDOF} \subset \{\omega_i\}_{i=1}^{N_\Omega} \cup \{\beta_i\}_{i=1}^{N_{\partial\Omega}}$.

The above mentioned assumption can also be rephrased as $u_h(\delta_i)$ are **unisolvent** degrees of freedom for functions in X_h.

A Priori Error Estimates

Theorem (A Priori LSQFEM)[2]

Under the above previous assumption the following error estimate holds,

$$\boxed{\|u - u_h\|_h} \leq \boxed{\inf_{v \in X_h} \|u - v\|_h} + \boxed{\sup_{v \in X_h} \frac{e(u, v)}{\|v\|_h}}.$$

[2] Bochev, P. B., & Gunzburger, M. D. (2009). Least-squares finite element methods (Vol. 166). Springer Science & Business Media.

A Priori Error Estimates

Theorem (A Priori LSQFEM)[2]

Under the above previous assumption the following error estimate holds,

$$\boxed{\|u-u_h\|_h} \leq \boxed{\inf_{v\in X_h}\|u-v\|_h} + \boxed{\sup_{v\in X_h}\frac{e(u,v)}{\|v\|_h}}.$$

In particular if the least square functional J^h is r-consistent, i.e. it exist $r>0$ such that,

$$\sup_{v\in X_h}\frac{e(u,v)}{\|v\|_h} \leq C(u)N^{-r},$$

then the above error estimate becomes,

$$\|u-u_h\|_h \leq \inf_{v\in X_h}\|u-v\|_h + C(u)N^{-r}.$$

[2] Bochev, P. B., & Gunzburger, M. D. (2009). Least-squares finite element methods (Vol. 166). Springer Science & Business Media.

A Priori Error Estimates

Therefore given that the assumptions 1 and 2 are verified we get that,

$$\boxed{\left\|u - \widetilde{u}_h\right\|_h} = \boxed{\inf_{v \in X_h} \|u - v\|_h} + \boxed{\sup_{v \in X_h} \frac{e(u, v)}{\|v\|_h}} + \boxed{\left\|u_h - \widetilde{u}_h\right\|_h}.$$

Therefore we have the following source of errors,

A Priori Error Estimates

Therefore given that the assumptions 1 and 2 are verified we get that,

$$\boxed{\left\|u - \widetilde{u}_h\right\|_h} = \boxed{\inf_{v \in X_h} \|u - v\|_h} + \boxed{\sup_{v \in X_h} \frac{e(u, v)}{\|v\|_h}} + \boxed{\left\|u_h - \widetilde{u}_h\right\|_h}.$$

Therefore we have the following source of errors,

1. the **best approximation error** ,

A Priori Error Estimates

Therefore given that the assumptions 1 and 2 are verified we get that,

$$\boxed{\left\| u - \widetilde{u}_h \right\|_h} = \boxed{\inf_{v \in X_h} \|u - v\|_h} + \boxed{\sup_{v \in X_h} \frac{e(u,v)}{\|v\|_h}} + \boxed{\left\| u_h - \widetilde{u}_h \right\|_h}.$$

Therefore we have the following source of errors,

1. the **best approximation error** ,
2. the **consistency error**,

A Priori Error Estimates

Therefore given that the assumptions 1 and 2 are verified we get that,

$$\boxed{\left\| u - \widetilde{u}_h \right\|_h} = \boxed{\inf_{v \in X_h} \| u - v \|_h} + \boxed{\sup_{v \in X_h} \frac{e(u,v)}{\| v \|_h}} + \boxed{\left\| u_h - \widetilde{u}_h \right\|_h}.$$

Therefore we have the following source of errors,

1. the **best approximation error** ,
2. the **consistency error**,
3. the **training error** .

A Priori Error Estimates

Therefore given that the assumptions 1 and 2 are verified we get that,

$$\boxed{\left\| u - \widetilde{u}_h \right\|_h} = \boxed{\inf_{v \in X_h} \| u - v \|_h} + \boxed{\sup_{v \in X_h} \frac{e(u,v)}{\|v\|_h}} + \boxed{\left\| u_h - \widetilde{u}_h \right\|_h}.$$

Therefore we have the following source of errors,

1. the **best approximation error** ,
2. the **consistency error**,
3. the **training error** .

Furthermore we have an additional source of error, the **quadrature error**, which is given by the fact that we would like to replace the norm $\|\cdot\|_h$ with the norm $\|\cdot\|_X$.

RELU Network and FEM[a]

[a] Xu, J. (2020). The finite neuron method and convergence analysis. arXiv preprint arXiv:2010.01458.

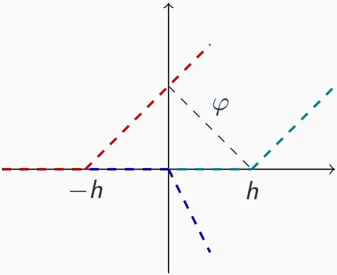
Rectified Linear Unit and FEM Shape Function

We now want to show that that one can reconstruct the linear finite elements space using RELU activation functions.

We know the finite element shape function on an uniformly spaced one dimensional mesh are defined as,

$$\varphi : [0,1] \to \mathbb{R}$$

$$x \mapsto \begin{cases} h^{-1}x & x \in [-h, 0] \\ 1 - h^{-1}x & x \in [0, h] \\ 0 & x \notin [-h, h] \end{cases}$$



now is just a matter of observing that one can rewrite the shape function φ as a linear combination of RELU functions with bias,

$$\varphi(x) = \boxed{h^{-1}\sigma(x+h)} - \boxed{2h^{-1}\sigma(x)} + \boxed{h^{-1}\sigma(x-h)}.$$

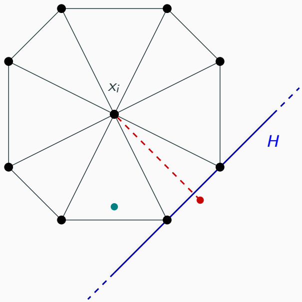
RELU and FEM – Representation of Shape Function

Now we would like to prove the same result for an n-dimensional case.

Lemma – Max Min Shape Function
Given a point x_i, we denote $G(i)$ the union of elements of the mesh having x_i ad vertex, then the tent function corresponding to x_i can be written as,

$$\varphi_i(x) = max\left\{0, \min_{T_k \subset G(i)} g_k(x)\right\}$$

where the function g_k is the linear function coinciding with the tent function on T_k, provided $G(i)$ is convex.



To explain this we consider two different cases, the first is when x belong to $T_{k_0} \subset G(i)$. Then we consider the hyper plane H and we focus our attention on g_{k_0}. In particular for all $y \in H \cap T_{k_0}$ we notice that,

$$g_k(y) \geq g_{k_0}(y).$$

RELU and FEM – Representation of Shape Function

Now $g_k(x_i) = g_{k_0}(x_i) = 1$ and therefore $g_k(x) \geq g_{k_0}(x)$.

Now let as assume that $x_i \notin G(i)$ then we consider the hyper plane connecting x and x_i, denoted in red. Now we consider the mesh element where the hyper plane lies to draw the H hyper plane. In particular since x_i and x are on different side of the hyper plane H and we know g_k is null on H then $g(x) < 0$ and therefore we need to impose the maximum with zero.

RELU and FEM – The Merge and Sort Proof

We are now ready to prove the inclusion of the FEM space with in shallow RELU neural network.

Theorem
Given a locally convex d-dimensional grid any P_1 spline on the grid characterised by $NDOF$ degrees of freedom can be also expressed as a RELU neural network with $log_2(k_{\mathcal{T}})+1$ hidden layers and $\mathcal{O}(k_{\mathcal{T}} NDOF)$ neuron, where $k_{\mathcal{T}}$ is the maximum number of neighborhood elements in the mesh.

Proof
We notice the following to begin with,

$$min\{a,b\} = \frac{a+b}{2} - \frac{|a-b|}{2} = v\sigma\left(W \cdot \begin{bmatrix} a \\ b \end{bmatrix}\right),$$

where v and W are defined as,

$$v = \frac{1}{2}\begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \qquad W = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

RELU and FEM – The Merge and Sort Proof

Now from the previous lemma we know we can represent each shape function as a max min problem, in particular the idea is to split each minimisation into sub minimisation until we are only comparing two elements. This operation can be represented using one RELU mapping. Therefore we need to concatenate $log_2(k_{\mathcal{T}})$ operations in this process. Now since the structure of the tree we have constructed in binary we can easily see that the number of nodes needed in the network is 2^k where k is the depth of the tree, this results in $\mathcal{O}(k_{\mathcal{T}})$ nodes per shape function.

RELU and FEM – Approximation Theory

Now that we have established a connection between linear finite elements and shallow RELU neural network we can use this connection to develop an interesting approximation estimate property,

Theorem
Given $\Omega = [0, 1]$ and a function $f \in H^2(\Omega)$ it exists a function $f_n \in DNN_1^N$ such the following estimate works,

$$||f - f_n||_{L^2(\Omega)} \leq C_f N^{-2}.$$

Proof
The idea is the following, we know that for a function $f : [0, 1] \to R$ that lives in $H^2(\Omega)$, it exists a linear FEM function $f_N : [0, 1] \to R$ such that:

$$||f - f_N||_{L^2(\Omega)} \leq C|f|_{H^2} \left(\frac{N}{3}\right)^{-2},$$

and this function can be represented as a DNN_1^N with RELU activation function.

RELU and FEM – Approximation Theory

Theorem (Petrushev)[3]
Let Ω be the unit ball in $\mathbb{R}^d$ and $f \in H^2(\Omega)$ then it exists $f_N \in DNN_1^N$ sucht that the folloing estimate holds,

$$\|f - f_N\|_{\mathcal{L}^2(\Omega)} \leq C\|f\|_{H^2} \left(\frac{N}{3}\right)^{-\frac{1}{2}-\frac{3}{2d}}.$$

[3]Petrushev, P. P. (1998). Approximation by ridge functions and neural networks. SIAM Journal on Mathematical Analysis, 30(1), 155-189.

Barron-Xu Space

Definition (Barron Space)
Let us consider the closure of the convex symmetric hull of $\mathbb{D}$,

$$B_1(\mathbb{D}) = \overline{\left\{ \sum_{j=1}^{n} a_j d_j \ : \ n \in \mathbb{N}, d_j \in \mathbb{D}, \ \left\| \{a_j\}_{j=1}^{n} \right\|_{\ell_1} \leq 1 \right\}}.$$

we define the Barron space and the associated norm as follow:

$$\|\cdot\|_{\mathcal{K}(\mathbb{D})} = \inf \left\{ c > 0 \ : \ cf \in B_1(\mathbb{D}) \right\},$$
$$\mathcal{K}(\mathbb{D}) = \left\{ f \in X \ : \ \|f\|_{\mathcal{K}(\mathbb{D})} < \infty \right\}.$$

In particular, given the fact that X is a Hilbert space, it is obvious that $\mathcal{K}_{\mathbb{D}}$ is a subset of X.

Barron Space – Dictionary

We now introduce the dictionary specific for shallow neural network, i.e.

$$\boxed{\mathbb{D}_\sigma = \left\{ \sigma(\mathbf{w}_i \cdot x + \mathbf{b}_i) \; : \; \mathbf{w}_i \in \mathbb{R}^d \textit{ and } \mathbf{b}_i \in \mathbb{R} \right\}}.$$

Barron Space – Dictionary

We now introduce the dictionary specific for shallow neural network, i.e.

$$\boxed{\mathbb{D}_\sigma = \left\{\sigma(\mathbf{w}_i \cdot x + \mathbf{b}_i) \ : \ \mathbf{w}_i \in \mathbb{R}^d \ \textit{and}\ \mathbf{b}_i \in \mathbb{R}\right\}}.$$

We can also define the space of shallow neural network with a small, but important, additional requirement,

$$\boxed{\Sigma_{N,M}(\mathbb{D}) = \left\{ \sum_{j=1}^{N} a_j d_j \ : d_j \in \mathbb{D} \ \textit{and}\ \left\|\{a_j\}_{j=1}^{N}\right\|_{\ell_1} \leq M\right\}}.$$

Barron Space – Approximations

Theorem (Maurey and Pisier – DeVore[4])
Let X be a Hilbert space and given $f \in \mathcal{K}(\mathbb{D}_\sigma)$ we have the following approximation estimate:

$$\inf_{f_N \in \Sigma_{N,M}(\mathbb{D}_\sigma)} \|f - f_N\|_X \leq C_X K_{\mathbb{D}_\sigma} \|f\|_{\mathcal{K}_1(\mathbb{D}_\sigma)} N^{-\frac{1}{2}},$$

where C_X is the type-2 constant for the space X. Furthermore when σ is a bounded activation function then $\mathbb{D}_\sigma$ is uniformly bounded in $\mathcal{K}(\mathbb{D}_\sigma)$

[4] DeVore, R. A. (1998). Nonlinear approximation. Acta numerica, 7, 51-150.

A Priori Error Estimates – Rademacher Complexity Analysis

Theorem (Quadrature Error)[5]
Given a class of functions $\mathscr{F} : \Omega \to \mathbb{R}$ and a collection of sample points $\{\omega_i\}_{i=1}^{N_\Omega}$,

$$\mathbb{E}_{\omega_i \sim \mathcal{D}(\Omega)} \sup_{h \in \mathscr{F}} \left| \sum_{i=1}^{N_\Omega} \frac{h(\omega_i)}{N_\Omega} - \int_\Omega h(x)\, d\mathcal{D}(\Omega) \right| \leq 2R_{N_\Omega}(\mathscr{F})$$
$$:= \mathbb{E}_{\omega_i \sim \mathcal{D}(\Omega)} \mathbb{E}_{\xi_i} \left[\sup_{h \in \mathscr{F}} \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \xi h(\omega_i) \right],$$

where ξ_i are Rademacher random variables and ω_i are uniformly distributed points that includes the $\delta_i, \ldots, \delta_{NDOF}$ from Assumption 1.

[5] Wainwright, M. J. (2019). High-dimensional statistics: A non-asymptotic viewpoint (Vol. 48). Cambridge University Press.

Rademacher Complexity Analysis

Corollary
Let $\mathscr{F}, \mathscr{G}$ be as in the previous slides. If we consider $f \in \mathscr{F}$ and $g \in \mathscr{G}$ then the following inequalities holds:

$$\mathbb{E}_{\omega_i \sim D(\Omega)} \|f\|_h^2 \leq \mathbb{E}_{\omega_i \sim D(\Omega)} \|f\|_X^2 + 2R_{N_\Omega}(\mathscr{F}),$$
$$\mathbb{E}_{\omega_i \sim D(\Omega)} \|f\|_X^2 \leq \mathbb{E}_{\omega_i \sim D(\Omega)} \|f\|_h^2 + 2R_{N_\Omega}(\mathscr{F}),$$

$$\mathbb{E}_{\beta_i \sim D(\partial\Omega)} \|g\|_h^2 \leq \mathbb{E}_{\beta_i \sim D(\partial\Omega)} \|g\|_X^2 + 2R_{N_{\partial\Omega}}(\mathscr{G}),$$
$$\mathbb{E}_{\beta_i \sim D(\partial\Omega)} \|g\|_X^2 \leq \mathbb{E}_{\beta_i \sim D(\partial\Omega)} \|g\|_h^2 + 2R_{N_{\partial\Omega}}(\mathscr{G}).$$

Rademacher Complexity Analysis

Theorem[6]
We will assume that $\|a_\alpha\|_{\mathcal{L}^\infty(\Omega)} \leq K_a$, $\|b_\alpha\|_{\mathcal{L}^\infty(\Omega)} \leq K_b$, $f \in \mathcal{L}^\infty(\Omega)$ and $g \in \mathcal{L}^\infty(\partial\Omega)$ then we have the following bounds,

$$R_{N_\Omega}(\mathscr{F}^{\Omega}_{N,M}) \leq MK_a\Big(K_{\mathbb{D}_\sigma} + 2\|f\|_{\mathcal{L}^\infty(\Omega)}\Big) \sum_{|\alpha|\leq m} R_{N_\Omega}(\partial^\alpha \mathbb{D}_\sigma),$$

$$R_{N_{\partial\Omega}}(\mathscr{F}^{\partial\Omega}_{N,M}) \leq MK_b\Big(K_{\mathbb{D}_\sigma} + 2\|g\|_{\mathcal{L}^\infty(\partial\Omega)}\Big) \sum_{|\alpha|\leq m} R_{N_{\partial\Omega}}(\partial^\alpha \mathbb{D}_\sigma),$$

provided that $u \mapsto (u,u)_y$ is a locally Lipschitz function. Assuming the activation function σ lives in $\mathcal{W}^{m+1,\infty}$, then for any α such that $|\alpha| \leq m$ one has the following estimate for the Rademacher complexity,

$$R_{N_p}(\partial^\alpha \mathbb{D}_\sigma) \leq C {N_p}^{-\frac{1}{2}}.$$

[6] Hong, Q., Siegel, J. W., & Xu, J. (2021). A priori analysis of stable neural network solutions to numerical pdes. arXiv preprint arXiv:2104.02903.

Main Result

Combining what we have seen up until now we get the following result,

Theorem
Let u and u_h be the usual continuous and discrete minimizer of the energy functional corresponding to the PDE, furthermore assuming $u \in \mathcal{K}(\mathbb{D}_\sigma)$, $F \in \mathcal{L}^\infty(\Omega)$ and $\sigma \in \mathcal{W}^{m+1,\infty}$ then,

$$\boxed{\mathbb{E}_{\omega_i,\beta_i}\|u - u_h\|_X} \leq \boxed{2C_X K_{\mathbb{D}_\sigma}\|u\|_{\mathcal{K}_1(\mathbb{D}_\sigma)} N^{-\frac{1}{2}}}$$
$$+ \boxed{3C_a N_\Omega^{-\frac{1}{4}} + 3C_b N_{\partial\Omega}^{-\frac{1}{4}}},$$

where in this case N is the number of neurons in the shallow layer, N_Ω and $N_{\partial\Omega}$ are respectively the number of evaluation points on the boundary and inside of Ω.

Regularity Theory

Regularity – An Example

Example[7]
We consider the following PDE on the all $\mathbb{R}^d$,

$$-\Delta u = \max\{0, x_1\}$$

the above PDE has solution $u(\mathbf{x}) = -\frac{\max\{0,x_1\}^3}{6} + h(\mathbf{x})$ where h is an harmonic function. It is possible to show that u grows so quickly that $\|u\|_{\mathcal{K}(\mathbb{D}_\sigma)}$ can't be finite if σ is the RELU activation function.

[7] Weinan, E., & Wojtowytsch, S. (2022, April). Some observations on high-dimensional partial differential equations with Barron data. In Mathematical and Scientific Machine Learning (pp. 253-269). PMLR.

Regularity – Conforming Activation Function

An alternative idea would be to consider $\sigma \in C^{2,\alpha}(\mathbb{R})$ and observe that,

$$\Delta\left[a\sigma(\omega^T x + b)\right] = a|\omega|^2 \partial^2 \sigma(\omega^T x + b).$$

Therefore if $\sigma \in C^{2,\alpha}(\mathbb{R})$ and $f \in \mathcal{K}(\mathbb{D}_{\partial^2 \sigma})$ then u solution of the Poisson problem over the all domain lives in $u \in \mathcal{K}(\mathbb{D}_\sigma)$. This argument comes from Weinan, E., & Wojtowytsch, S. (2022, April). Some observations on high-dimensional partial differential equations with Barron data. In Mathematical and Scientific Machine Learning (pp. 253-269). PMLR, where it has also been applied to other PDE.

Regularity – The Role of Boundary Conditions

Example [8]
Let u be the solution of the Laplace equation in d dimension,

$$\begin{cases} -\Delta u(\mathbf{x}) = 0 \; |x| < 1, \\ u(x) = \max\{0, x_1\} \; |x| = 1. \end{cases}$$

Assuming $u \in \mathcal{K}(\mathbb{D}_{\sigma_3})$ then we can extend u to the all space and observe that u is discontinuous on the equator of the ball $|x| \leq 1$. Using the fact that singular set of a Barron functions is the countable union of affine spaces we found an absurd when $d > 2$.

[8] Weinan, E., & Wojtowytsch, S. (2022, April). Some observations on high-dimensional partial differential equations with Barron data. In Mathematical and Scientific Machine Learning (pp. 253-269). PMLR.

Second Order Methods For Training

Second Order Training

In this section of the presentation I'd like to make a case for second order training methods when training neural networks for solving PDE. In particular we are concerned with a part of the error never explored before

$$\boxed{\left\| u_h - \widetilde{u_h} \right\|_X}.$$

One way of obtaining a minimzer $\widetilde{u_h}$ such that the above quantity can be bounded a priori is to use a greedy algorithm[9] but we will not concern our self with this problem at the moment.

[9] Hao, W., Jin, X., Siegel, J. W., & Xu, J. (2021). An efficient greedy training algorithm for neural networks and applications in PDEs. arXiv preprint arXiv:2107.04466.

The Problem – Network and Loss

To make a case for second order training method we will consider a very particular example of elliptic PDE, i.e. the L^2 regression applied on functions in 2D and 3D.

The idea behind the problem we are considering is to construct a neural network with the shape to the one consider until now but deeper.
Number of Hidden Layers: 3
Number of Neurons per Hidden Layers: 100
Dimension of parameter space: ~ 20000

In particular the loss functional taken into consideration is,

$$\mathcal{L}(u, f) = \frac{1}{N} \sum_{i=1}^{N} |u(\mathbf{x_i}) - f(x_i)|^2, \qquad \mathbf{x_i} \sim \mathcal{U}(\Omega), \qquad \Omega = [0, 1]^2.$$

The Problem – Interpolation

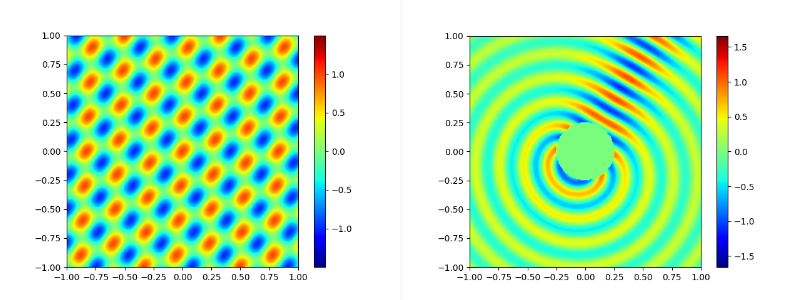


Figure 1: The figure displays the two function we would like to interpolate. In particular on the left we have our 2D problem, while on the right we have one time frame of the 3D problem.

The Problem – Why this problem ?

The particular reason why we chose this problem are,

- The problem is **big enough**, in the sense that in the 2D case we train with 10^4 random points, while in the 3D case we train with 10^6 random points.
- The problem is **easy to scale**, we can easily considered more sampled points and deeper/larger newtorks.
- The problem is **highly non convex**, as many of the problems that appear in practice.

The Standard Methods

We will compare different methods,

- **ADAM**,
- **BFGS**[10],
- Second order methods

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma \widetilde{H}f(\mathbf{x}_n)^{-1} \nabla f(\mathbf{x}_n),$$

  where $\widetilde{H}$ is a matrix containing second order information, in particular we considered,

  - **Newton methods**, $\widetilde{H}$ is the exact Hessian,
  - **Gauss-Newton methods**, $\widetilde{H} = J^T(\partial_y^2 \mathcal{L})J$.

The linear system $\widetilde{H}f(\mathbf{x}_n)\delta = \nabla f(\mathbf{x}_n)$, is solved using a **Krylov iterative method**.

[10] Wright, S., & Nocedal, J. (1999). Numerical optimization. Springer Science, 35(67-68), 7.

The Standard Methods – Loss History

There is evidence that second order method are superior then first order method to train our network to interpolate the function.

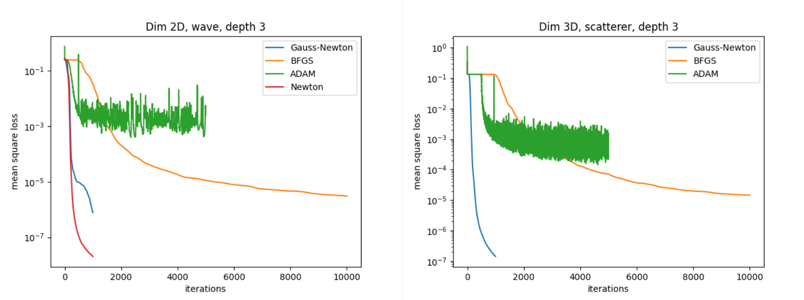


Figure 2: The figure show how the objective function decays for four different method Newton,Gauss-Newton, BFGS and ADAM.

The Standard Methods – Loss History

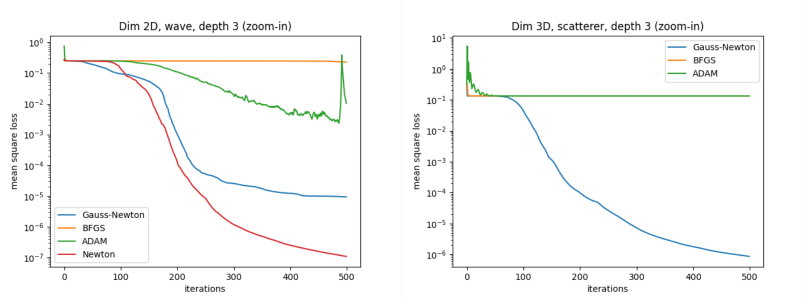


Figure 3: The figure show how the objective function decays for four different method Newton,Gauss-Newton, BFGS and ADAM.

Result – ADAM

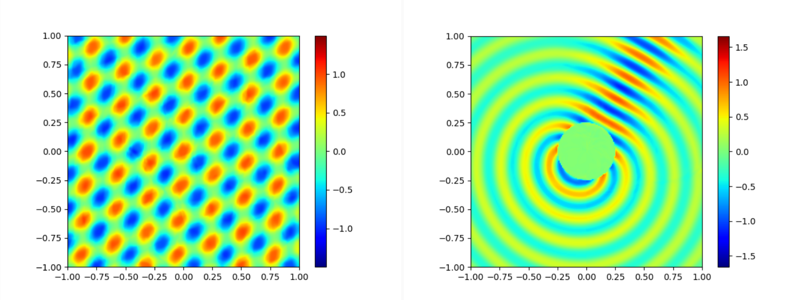


Figure 4: The figure shows the outcome of the training for both our toy problem, when as training method we used ADAM. On the left we have our 2D problem, while on the right we have one time frame of the 3D problem. In particular we notice that small features can't be resolved with ADAM.

Result – Gauss-Newton

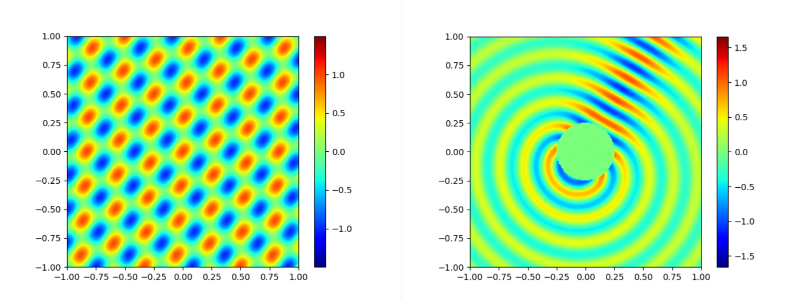


Figure 5: The figure shows the outcome of the training for both our toy problem, when as training method we used Newton. On the left we have our 2D problem, while on the right we have one time frame of the 3D problem. In particular we notice that small features can be resolved sing Newton method.

Learning Coordinates Matters

We now observe the difference if we train the first 3D problem using a data set in polar coordinates or in Cartesian coordinates.

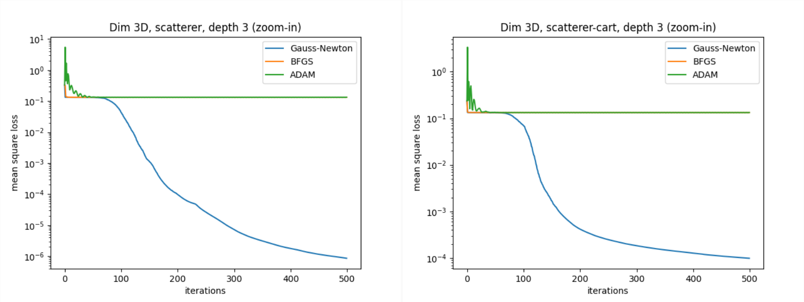


Figure 6: On the left we have considered polar coordinates while on the right we have used Cartesian coordinates in the data set used to train the neural network.

An Example – Physically Informed Neural Network

Physically Informed Neural Networks – PINNs

We will focus our attention on a particular PDE form now on, ie.

$$\Delta u = f \ in \, \Omega$$

$$u \equiv 0 \ on \, \partial\Omega$$

If we consider as energy functional $J(u, F)$ and $J(u_h, F)$ the continuous and discrete residual we will obtain the so-called dumb least square formulation for the Laplace problem, i.e.

$$\boxed{\textit{find } u \in X \textit{ such that } \int_\Omega \Delta u \Delta v = \int f \Delta v \ \forall v \in X.}$$ [11]

In particular if we discretise the energy using a Montecarlo scheme we obtain the following energy functional,

$$\mathcal{L}(u, f) = \frac{1}{N_\Omega^2} \sum_i^N |\Delta u(\mathbf{x_i}) - f(x_i)|^2 + \frac{1}{N_{\partial\Omega}} \sum_i^N |u(\mathbf{b_i})|^2$$

$$\mathbf{x_i} \sim \mathcal{U}(\Omega) \qquad \mathbf{b_i} \sim \mathcal{U}(\partial\Omega) \qquad \Omega = [0, 1]^2$$

[11] Bochev, P. B., & Gunzburger, M. D. (2009). Least-squares finite element methods (Vol. 166). Springer Science & Business Media.

PINNs– Laplacian Eigenvalue

The minimization problem for the PINN requires a long training in particular the figures below are computed using 50000 iteration of ADAM.

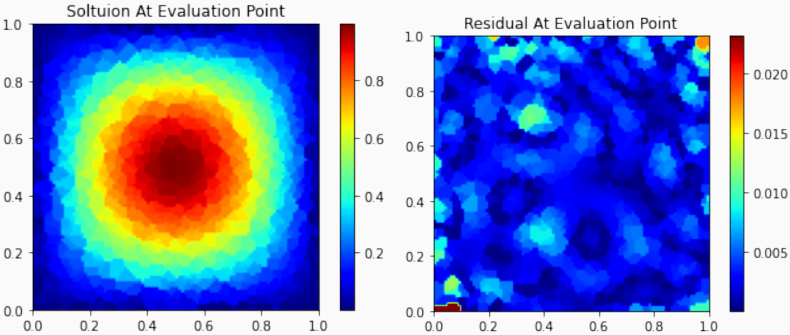


Figure 7: The figures above show the first eigenfunction of the Laplacian computed numerically and the residual both in the collocation points.

PINNs – Laplace Eigenfunctions

One strong advantage of PINNs is that they outperform all other method at inference time, by a very large factor.

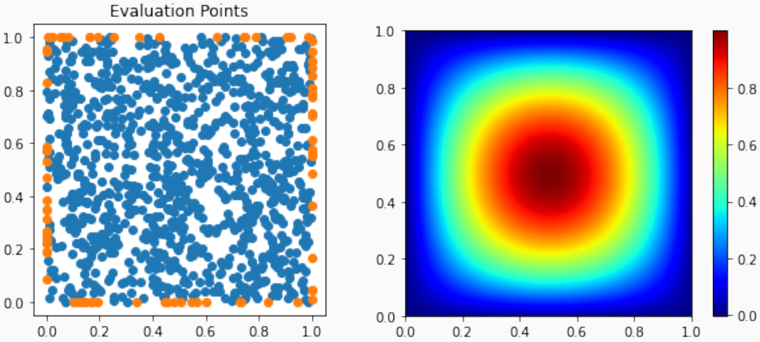


Figure 8: The figures shows on the left the points used to evaluate the collocation error, while the right one can see the solution at inference time on million degree of freedom.

An Other Example – The Infinity Laplacian

The Infinity Laplacian

Given a smooth function $\varphi : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ we define the infinity Laplacian operator as,

$$\Delta_\infty \varphi := \sum_{i,j=1}^{n} \varphi_{x_i} \varphi_{x_j} \varphi_{x_i x_j} = H\varphi \nabla\phi \cdot \nabla\phi.$$

- The infinity Laplacian is an operator not expressible in divergence form, i.e. we can't integrate by parts to obtain a weak formulation.
- We define the solution of

$$\begin{cases} \Delta_\infty \varphi = 0 \; in \; \Omega \\ \varphi_{|\partial\Omega} = g \end{cases}$$

  using the notion of **viscosity solution**.
- There regularity theory of the infinity Laplacian is **incomplete**.

The Infinity Laplacian – Why Neural Networks

- There is evidence that deep neural networks can well approximate the viscosity solution of partial differential equations.[12]
- We are interested in the Infinity Laplacian in high dimensions.
- We need a numerical scheme not based on a weak solution.

[12] See reference in Weinan, E., & Wojtowytsch, S. (2022, April). Some observations on high-dimensional partial differential equations with Barron data. In Mathematical and Scientific Machine Learning (pp. 253-269). PMLR.

We minimize the following loss functional within the set of discrete functions made by deep neural network in this case,

$$\mathcal{L}(u,f) = \frac{1}{N_\Omega}\sum_{i}^{N} |\Delta_\infty u(\mathbf{x_i})|^2 + \frac{1}{N_{\partial\Omega}}\sum_{i}^{N} |u(\mathbf{b_i}) - g(\mathbf{b_i})|^2$$

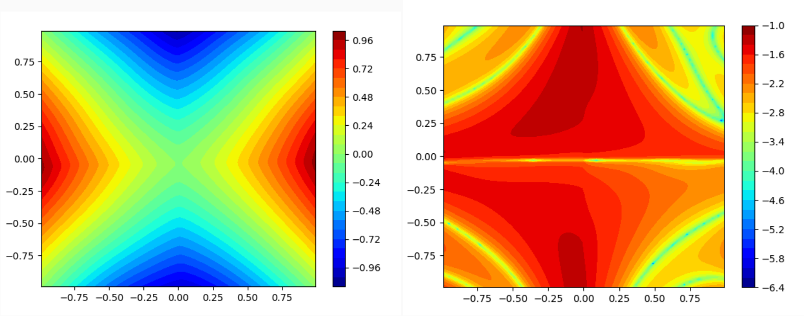


Figure 9: On the right the computed solution of the Aronsson example, on the left the error in log scale.

Conclusion

- We can use ideas developed in the context of **Least SQuares Finite Elements Methods** and **GAlerkin Least Squares** in order to analyse PINNs and other deep neural networks methods used in scientific computing.

Conclusion

- We can use ideas developed in the context of **Least SQuares Finite Elements Methods** and **GAlerkin Least Squares** in order to analyse PINNs and other deep neural networks methods used in scientific computing.
- The a priori estimates here presented is **far** (**really far** !) from being optimal.

Conclusion

- We can use ideas developed in the context of **Least SQuares Finite Elements Methods** and **GAlerkin Least Squares** in order to analyse PINNs and other deep neural networks methods used in scientific computing.
- The a priori estimates here presented is **far** (**really far** !) from being optimal.
- If we want to use PINNs in scientific computing we **need** to become **better at training**.
- Elliptic **regularity results** in terms of Barron spaces need to be developed.

Thank you !