Parametric Complexity Bounds for Approximating PDEs with Neural Networks

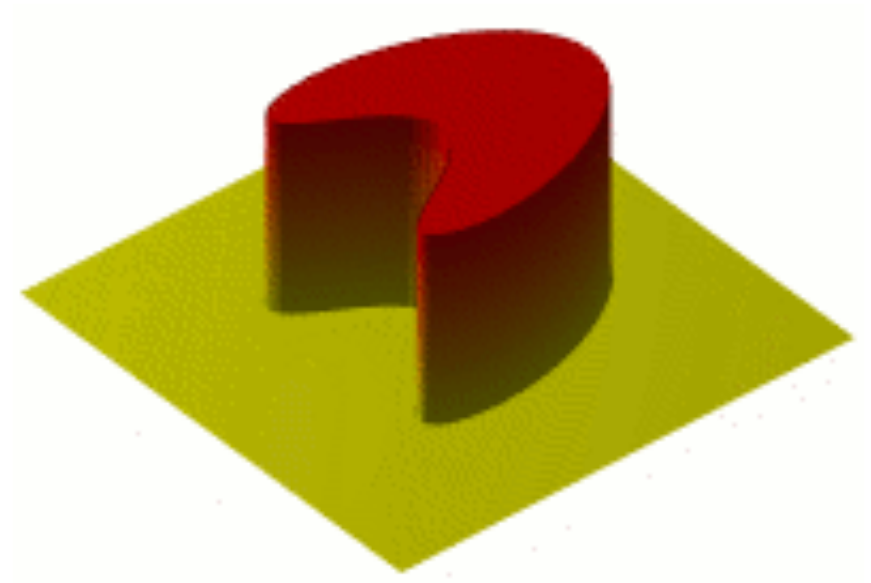
Tanya Marwah, Zachary C. Lipton, Andrej Risteski

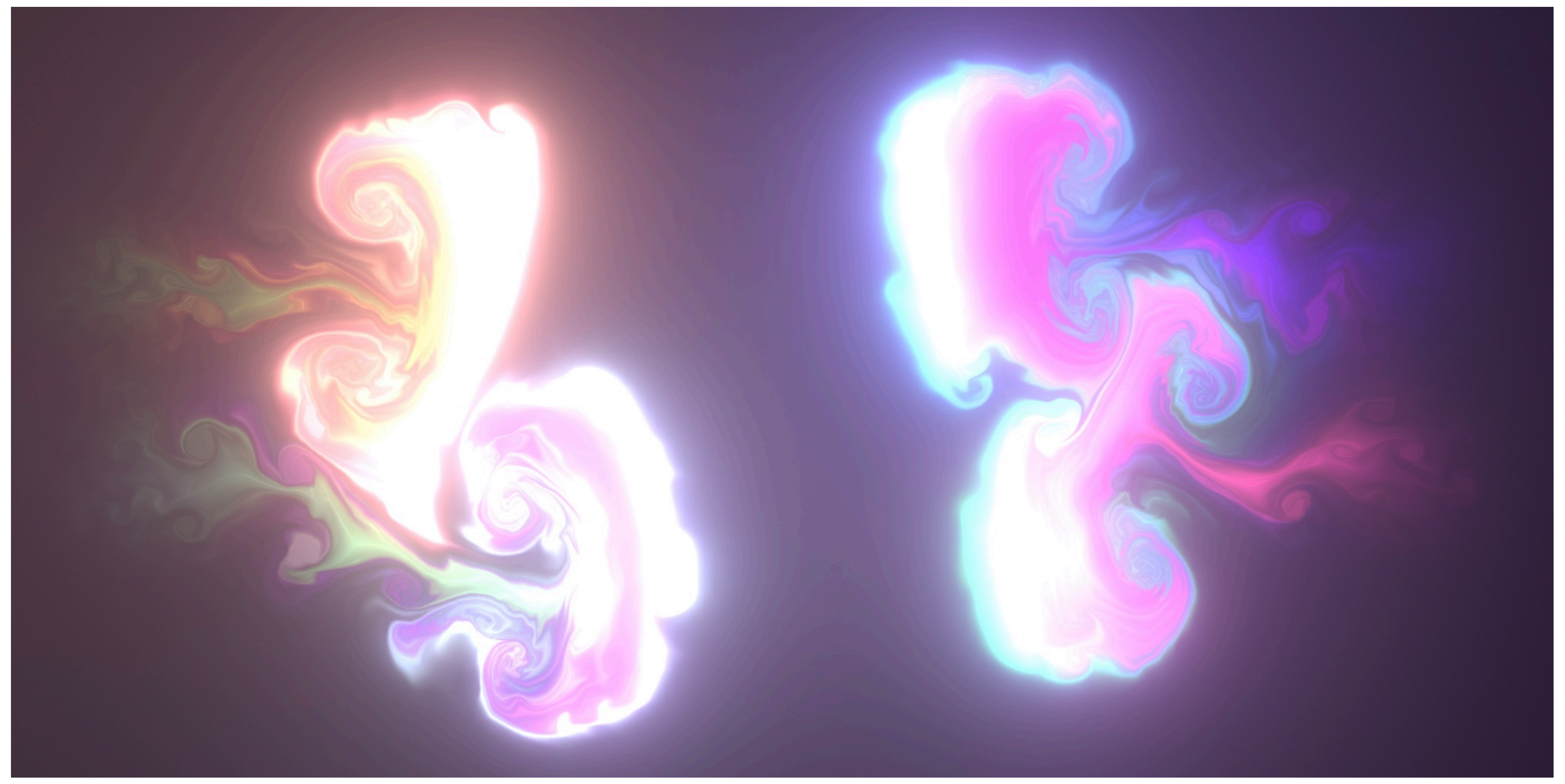
NEURAL INFORMATION PROCESSING SYSTEMS

Carnegie Mellon University

Partial Differential Equations

A partial differential equation (PDE) relates a multivariate function defined over some domain to its partial derivates.





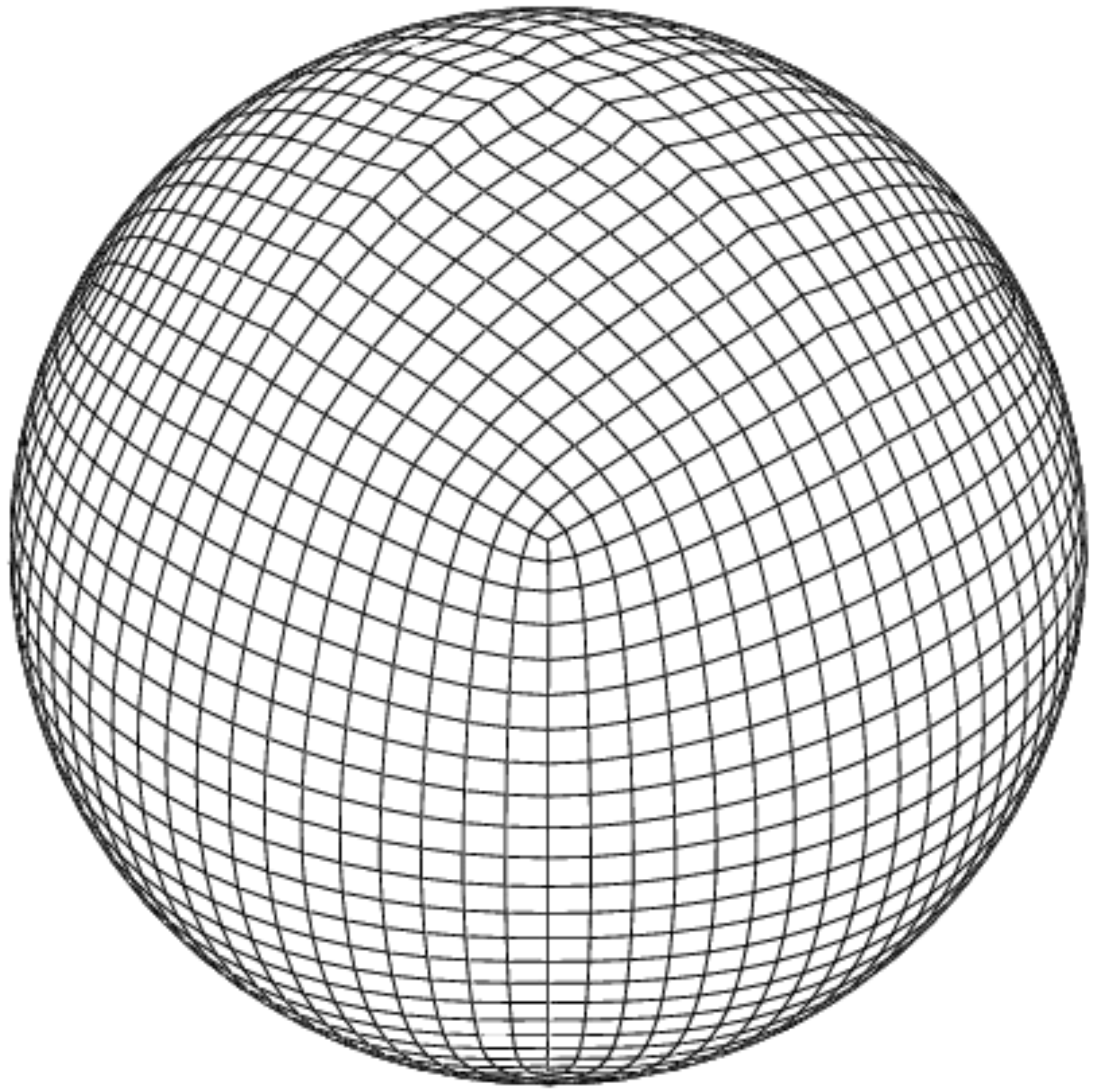
Numerical Methods

Numerical methods such as finite element or finite differences methods discretize the input domain.

Reduces the problem to solving a system of linear equations.

Computation cost for $O(\epsilon)$ error is $\sim \left(\frac{1}{\epsilon}\right)^d$

Prohibitive in high dimensions (curse of dimensionality)



Solutions using Neural Networks

Use Neural networks to represent the solutions to PDEs.

Empirical benefits:

Mesh free (E and Yu. 2017, Raissi et al. 2017)

Expressivity (Li et al. 2020)

Do not scale exponentially in the input dimension (Grohs et al. 2018)

Theoretical Analysis:

Previous work (Sirignano et al. 2018, Khoo et al. 2017) prove universal approximation based bounds. They do not analyze when can neural networks improve upon grid based methods.

Our Result

For the class of linear elliptic PDEs, if the coefficients of the PDE are approximable by neural nets with at most N parameters, then the solution to the PDE can be approximated by a neural network with $O(d^{\log(\frac{1}{\epsilon})}N)$ parameters.

We introduce a technique wherein we simulate gradient descent in an appropriate function (Hilbert) space through the very architecture of a neural network. Each iterate, given by a neural network is subsumed into the (slightly larger) network representing the subsequent iterate.

Linear Elliptic PDE

A **linear elliptic PDE** is defined as:

$$\begin{cases} (Lu)(x) \equiv \big(-\text{div}\,(A\,\nabla u) + cu\big)(x) = f(x), \forall x \in \Omega, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad u(x) = 0, \forall x \in \partial\Omega, \end{cases}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded open set with boundary $\partial\Omega$. Further, for all $x \in \Omega$, $A : \Omega \to \mathbb{R}^{d\times d}$ is a matrix valued function, such that $A(x) \succ 0$, and $c : \Omega \to \mathbb{R}$, s.t, $c(x) \geq 0$.

For the operator L: $(\lambda, \varphi)_{i=1}^{\infty}$ are the (eigenvalue, eigenfunction) pairs, where $0 < \lambda_1 \leq \lambda_2 \leq \cdots$.

Here div denotes the divergence operator: Given a vector field $F : \mathbb{R}^d \to \mathbb{R}^d$, div$F = \nabla \cdot F = \sum_{i=1}^{d} \frac{\partial F_i}{\partial x_i}$

Assumptions

Functions A, c are *infinitely differentiable* and can be ϵ-approximated by neural networks with infinitely differentiable activations and N_A and N_c parameters respectively.

The function f can be approximated by the neural network f_{nn} with N_f parameters such that $\|f - f_{nn}\|_{L^2(\Omega)} \leq \epsilon_{nn}$.

There exists a function f_{span} that lies within the *span of the first-k eigenfunctions* of L such that $\|f - f_{\text{span}}\|_{L^2(\Omega)} \leq \epsilon_{\text{span}}$.

Main Theorem

Theorem (informal): If there exists a neural network u_0 with N_0 parameters such that $\|u^\star - u_0\|_{L^2(\Omega)} \leq R$, for some $R < \infty$, then for every $T \in \mathbb{N}$ there exists a neural network with size

$$O\left(d^{2T}(N_0 + N_A) + T(N_f + N_c)\right)$$

such that

$$\|u^\star - u_T\|_{L^2(\Omega)} \leq \epsilon + \tilde{\epsilon}$$

where $\epsilon := \left(1 - \dfrac{2}{\lambda_k + \lambda_1}\right)^T R$ and $\tilde{\epsilon} = O(\epsilon_{\text{span}} + \epsilon_{\text{nn}})$.

Remarks

The number of parameters in the final network depends upon how close the initial estimate is to the solution $u^\star$ and its number of parameters N_0. Therefore there will be a trade-off , where better approximation may require more parameters.

While $\epsilon \to 0$ as $T \to \infty$, $\tilde{\epsilon}$ is a bias error term that does not go to 0 as $T \to \infty$. It contains terms that depends upon the approximation errors for f not entirely lying in the span of the first k eigenfunctions of L.

The relation $\epsilon = \left(1 - \dfrac{2}{\lambda_k + \lambda_1}\right)^T R$ comes from the fact that we are stimulating T steps of gradient descent on a strongly convex loss in a function space. λ_k and λ_1 can be thought of as the effective Lipschitz and strong convexity constants of the loss.

Proof Sketch

Define convergent sequence:

- We show that for operator L, we can define a sequence of functions that converges to ϵ optimal function approximation (in $L^2(\Omega)$ norm) after $O(\log(1/\epsilon))$ steps.
- The updates take the following form

$$u_{t+1} \leftarrow u_t - \frac{2}{\lambda_k + \lambda_1}(Lu_t - f)$$

- By ensuring that each iterate remains close to the span of the top-k eigenfunctions of L, we make sure that all the functions in sequence and hence the solution satisfy the boundary condition.

Proof Sketch

Approximating iterates by neural networks:

We show that if at step t, the function u_t is a neural network with N_t parameters, then u_{t+1} is a neural network with $O(d^2(N_A + N_t) + N_t + N_f)$ parameters.

We use the following results to show the above recurrence:

- *Backpropagation*: If $f : \mathbb{R}^m \to \mathbb{R}$ is a neural network with N parameters, then the network that calculates its the gradient $\frac{df}{di}$ for all $i \in [m]$ is a neural network with $O(2N)$ parameters.
- The *addition*—or *multiplication*—of two functions representable as neural networks with sizes N_1, N_2 can be represented as neural network with size $O(N_1 + N_2)$

Conclusion

Our key contribution is to show that the solution of a linear elliptic PDE can be approximated by a neural network with $O(\text{poly}(d)N)$ parameters if the coefficients of the PDE are approximable by neural networks with at most N parameters.

Future work:

- Extension to other boundary conditions.
- Lower bounds.
- Extension to other PDEs, for example the Helmholtz PDE.