Mathematisches Forschungsinstitut Oberwolfach

Report No. 48/2023

DOI: 10.4171/OWR/2023/48

Mini-Workshop: Nonlinear Approximation of High-dimensional Functions in Scientific Computing

Organized by
Mathias Oster, Aachen
Janina Schütte, Berlin
Philipp Trunschke, Nantes

15 October – 20 October 2023

Abstract. Approximation techniques for high dimensional PDEs are crucial for contemporary scientific computing tasks and gained momentum in recent years due to the renewed interest in neural networks. It seems that especially nonlinear parametrizations will play an essential role in efficient and tractable approximations of high dimensional problems. We held a mini-workshop on the relation and possible synergy of neural networks and tensor product approximation. To reliably evaluate the prospect of different numerical experiments, the traditional talks were accompanied by live coding sessions.

Mathematics Subject Classification (2020): 15A69, 68T07, 35Q93, 65F55, 41A46.

Introduction by the Organizers

The workshop *Nonlinear Approximation of High-dimensional Functions in Scientific Computing*, organised by Mathias Oster (RWTH Aachen), Janina Schütte (WIAS Berlin) and Philipp Trunschke (École Centrale de Nantes) was attended by 17 people (16 on-site and 1 online) with affiliations for example in Germany, the US, the UK, France, Italy and the Netherlands. The program consisted of 16 talks (50 minutes) and three coding sessions (90 minutes), allowing for extended discussions throughout the workshop. Conversations with all participants lead to a positive conclusion. The workshop was a success fostering new collaborations, strengthening standing connections and providing the space to learn about other attendees research in the talks, while also having time to discuss new ideas during breaks and the coding sessions.

Topic. Numerous state-of-the-art applications in engineering and physics rely on the efficient solution of high-dimensional *partial differential equations* (PDEs) with controllable precision and reliable error bounds. But classical methods like finite differences, finite elements and finite volumes are limited to low dimensions due to an exponential growth in complexity. To circumvent this curse of dimensionality, new approximation methods such as sparse approximations, tensor product approximations and neural networks have been developed.

This mini-workshop explored the benefits and limitations of contemporary methods for neural network and tensor network approximations of high-dimensional functions and used the generated insights to discuss possible new and improved tools. Here, the coding session allowed the participants to explore some new ideas on-site, as for example using a combination of (global) linear transformation and tensor trains to reduce the ranks, exploring the implicit bias observed for linear networks as well as synthesising tensor trains with neural networks by using functional tensor trains whose basis functions are parametrised by neural networks. The following topics have been discussed in the workshop.

- **Theory-to-practice gap** The theory-to-practice gap describes two orthogonal phenomena in machine learning. On the one hand, it is often observed that neural networks outperform their theoretical expressivity bounds when the required accuracy is moderate. In particular, many proofs for approximation rates of neural networks show that certain network architectures are able to model classical approximation schemes. It is thus natural to ask when the trained networks can perform better than these classical algorithms and manifest the first interpretation of the theory-to-practice gap. On the other hand, the theory-to-practice gap describes the practical difficulty of estimating neural networks from point evaluations. Theoretical constructions demonstrate that the required sample size may suffer from the curse of dimensionality and practical experiments substantiate that even the approximation of "simple" functions, like the square $x \mapsto x^2$, is difficult to high-accuracy. This obviously depends on the distribution of the data and may be alleviated by model- and problem-dependent importance sampling schemes. However, theoretical results in this direction are currently sparse and first advances for the special case of tensor networks have been discussed in the workshop. As of now, it remains unclear if the theory-to-practice gap for general neural networks can be bridged or if it is a fundamental limitation of the model class akin to the concept of the "condition number" in numerical linear algebra.
- **Neural Operators** For neural operator techniques as Deep-O-Net or Fourier-Neural-Operators it is often claimed that they can approximate mappings from one functional space to another functional space with "discretisation invariant" schemes. These invariance claims have have been discussed and some counter examples have been presented. This also leads to interesting tasks of correct sampling of functional spaces ("Besov priors").

- **Mean-Field Limit** Two mean field generalisations of deep learning, based on *neural ordinary differential equations* (neural ODEs) have been discussed. The first approach considered the learning problem in the mean-field limit of the data. In this setting, the learning problem can be interpreted as an optimal control problem in Wasserstein space, where the initial data distributions is transported by means of a neural network (the control). Another approach, presented the infinite width and depth limit of neural networks as neural ODEs with Barron functions as vector fields and formulated an corresponding abstract optimal control problem with measure-valued controls.
- **Optimisation** The abundance of local minima in learning tensor networks and neural networks leads to an influence of the chosen optimisation scheme on the resulting generalisation performance. Of particular interest in this context is the implicit regularisation in the context of overparameterisation (more parameters than training data), i.e., which networks are favoured by such algorithms. This implicit bias was discussed for neural networks with linear and non-linear activation functions. In the optimisation of tensor methods optimal sampling strategies and active learning have been of interest.
- **Synthesising Techniques** Finally, part of the workshop was concerned with combining tensor decomposition methods with more classical approaches, such as sparse approximation schemes, for solving time-space discretisations of parabolic PDEs and model order reduction techniques for optimal controls of the Navier–Stokes equation.

As expected, these complex open problems were not solved in one week. Nevertheless, discussions in all considered areas were productive and new ideas and collaborations were found.

Mini-Workshop: Nonlinear Approximation of High-dimensional Functions in Scientific Computing

Table of Contents

Abstracts

Semi-global Optimal Control Problems and their Applications to Machine Learning

Mathias Oster
(joint work with Angela Kunoth, Reinhold Schneider)

Learning a function $f : \mathbb{R}^d \to \mathbb{R}$ by deep neural networks with activation function σ in for example the L^2 norm can be interpreted as an abstract optimal control problems with measure-valued controls $\mu(t)$ of the form

$$\min_{\mu(\cdot)} \mathcal{J}(\mu(\cdot)), \quad \mathcal{J}(\mu(\cdot)) = \int_{\mathbb{R}^d} \|f(x) - \int a\sigma(A\ z(T,x) + b)\ d\mu(t;a,A,b)\|^2 dx$$

$$s.t. \frac{d}{dt} z(t,x) = \int a\sigma(Az(t,x) + b)\ d\mu(t;a,A,b), \quad z(0,x) = x$$

and provides an interesting mathematical framework to analyse the expressivity and optimization of deep neural networks from a continuous point of view. This control problem can be seen as an infinitely deep neural network with distinguished last layer. Here we exploit the ideas of Barron spaces as continuous interpretation of infinitely wide shallow networks and neural odes as infinitely deep residual network architectures. This continuous interpretation might allow one to deduce new adaptive algorithms for neural network that change the depth and width of the neural network during the training process.

First, we show the existence of minimizers to the optimal control problem by using Prokhorov's theorem on tight measures and some regularity assumptions on the activation function and classical compactness and continuity arguments.
Secondly, we analyse analyse the gradient flows corresponding to optimizing the map $\mu(\cdot) \to \mathcal{J}(\mu(\cdot))$ in the space of probability measures. To that end, we introduce a fibered Wasserstein metric on probability measures with bounded second moment and fixed first marginal and define the notion of absolute continuous curves. Furthermore, we define a notion of Wasserstein gradient and exemplify it on the example of a potential functional $\mathcal{E}(\mu) = \int V(u) d\mu(u)$ for some twice continuously differentiable function V. By using the equivalence of absolute continuous curves and solutions to the continuity equation we can state the gradient flow equations for the optimal control problem and we sketch the proof of existence of gradient flows based on the so-called generalized minimizing movement.

Lastly, we propose a first naïve algorithm to deal with flexible architectures and provide some very first examples.

References

[1] L. Ambrosio, N. Gigli and G. Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*, Birkäuser (2005).

[2] Z. Ding, S. Chen, Q. Li and S.J. Wright. Overparameterization of DeepResnet: Zero Loss and Mean-field Analysis. Journal of Machine Learning Research **23.48** (2022), 1–65.

Approximation of high-dimensional functions with tensors and neural networks

Ivan Oseledets

Approximation of multivariate functions is a notoriously difficult task. In this talk, I discussed two different approaches: tensor decompositions and neural networks/operators.

The idea behind tensor decompositions is based on the separation of variables. Several tensor formats exist that utilize this idea: the simple canonical decomposition, which has well-known problems with stability if used as a general approximation tool, and SVD-based tensor formats such as tensor train and Hierarchical Tucker (H-Tucker). Using those formats, one can often approximate functions with high precision. Moreover, for a special class of functions written in the so-called inverse Polish notation, we can constructively represent tensors with optimal ranks. Some applications include computation of the matrix permanent and cooperative games, for more details see [3]. The idea of quantized tensor train (QTT) uses the procedure of tensorization. For example, given a function $f(x) = \sin(x)$ we can create a vector $v = 2^d$ of length d of values of this function on a uniform grid and reshape it into a $2 \times 2 \times \ldots \times 2$ d-dimensional tensor. For this example, the QTT-ranks will be equal to 2, giving logarithmic complexity. Moreover, one can show that for a certain class of functions QTT-representation gives the approximation of a function with complexity $\mathcal{O}(\log^{\alpha} \varepsilon)$, where ε is the approximation accuracy [4, 5].

However, it is also clear that there are important cases when tensor approximation fails, for example, for function with diagonal singularities like

$$f = e^{-x^2/2} e^{-y^2/2} e^{-|x-y|}.$$

A big alternative are neural networks, which are universal function approximators. However, the converger of the error with respect to the number of parameters is not well understood. A promising class of functions seems to be Deep-ReLU networks, especially due to the results of Yarotsky [7]. It can be shown, for example, that a function $f(x) = x^2$ can be well-approximated using DeepReLU network and the error decays exponentially with the depth. Based on this result, one can show that polynomials can be well-approximated and large classes of functions. In [6] we showed that even for the simplest one-dimensional example it is not possible to recover such a good Deep ReLU representation: instead of 10^{-6} we get $10^{-2} - 10^{-3}$ error of approximation at its best. The reason for that is the loss function is very "narrow" in this particular point. The current understanding of the situation is that deep feedforward networks can be very unstable in training, and we need to look for alternatives.

A promising direction is the approximation not of the solutions, but of the mappings using so-called *neural operators*. Neural operator is a parametrized mapping from a function (element of a Banach space) to another function (Banach space), and they are quickly gaining popularity. Popular approaches include DeepONet [8] and Fourier Neural Operator (FNO). All of them still can not be considered as real operators, and they do not improve with better discretizations, as standard methods. However, in many cases they provide an extremely fast surrogate model.

Among open problem for training neural operators, I want to highlight the following one. A standard approach is to construct a dataset of input-output pairs. The input pair (for example, coefficient in the diffusion equation) is sampled from a certain probability distribution over functions. But this distribution is taken empirically, like random mixture of Gaussians or random trigonometric polynomials. However, it is not clear why these functions are used for training, and what is the motivation for using such kind of functions. The research question, that needs an answer is what the optimal (or quasioptimal) way of sampling input data for different kinds of problems, where neural operators are used? Understanding and the solution of the problem may be the key for the generalization of such neural operators and their wider usage.

References

[1] I. V. Oseledets, *Tensor-train decomposition*, SIAM Journal on Scientific Computing **33** (2011), 2295–2317.

[2] L. Grasedyck, *Hierarchical singular value decomposition of tensors*, SIAM journal on matrix analysis and applications **31** (2010), 2029–2054.

[3] G. Ryzhakov and I. Oseledets, *Constructive TT-representation of the tensors given as index interaction functions with applications*, in *The Eleventh International Conference on Learning Representations*, 2022.

[4] V. Kazeev and C. Schwab, *Quantized tensor-structured finite elements for second-order elliptic PDEs in two dimensions*, Numerische Mathematik **138** (2018), 133–190

[5] I. V. Oseledets, *Constructive representation of functions in low-rank tensor formats*, Constructive Approximation **37** (2013), 1–18

[6] D. Fokina and I. Oseledets, *Growing axons: greedy learning of neural networks with application to function approximation*, arXiv preprint arXiv:1910.12686 (2019)

[7] D. Yarotsky, *Error bounds for approximations with deep ReLU networks*, Neural Networks **94** (2017), 103–114.

[8] Lu, Lu and Jin, Pengzhan and Karniadakis, George Em, *Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators*, arXiv preprint arXiv:1910.03193 (2019)

[9] Z. Li, N. B. Kovachki, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar et al., *Fourier Neural Operator for Parametric Partial Differential Equations*, International Conference on Learning Representations, 2020.

Optimal Sampling for Approximate Gradient Descent

Philipp Trunschke
(joint work with Robert Gruhlke, Charles Miranda, Anthony Nouy)

We consider the problem of minimising a *loss functional*

$$\operatorname{minimise}_{v\in\mathcal{M}} \mathcal{L}(v), \qquad \mathcal{L}(v) := \int \ell(v;x)\,\mathrm{d}\rho(x)$$

over a possibly nonlinear *model class* $\mathcal{M} \subseteq \mathcal{H}$ in a Hilbert space $\mathcal{H}$. When computing the integral is infeasible, a common approach is to replace the exact loss $\mathcal{L}$ with a Monte Carlo estimate before employing a standard gradient descent scheme. This results in the well-known *stochastic gradient descent* (SGD) method. However, using an estimated loss instead of the true loss can result in a "generalisation error". Rigorous bounds for this error usually require compactness of $\mathcal{M}$ and Lipschitz continuity of $\mathcal{L}$ while providing a very slow decay with increasing sample size. This slow decay is unfavourable in settings where high accuracy is required or sample creation is costly.

To address this issue, we propose a new approach that performs successive corrections on local linearisations of $\mathcal{M}$. To be specific, we suppose that in every step $t \in \mathbb{N}$ there exists a linear space $\mathcal{T}_t$ that approximates $\mathcal{M}$ locally around the current iterate u_t. Given the gradient $g_t := \nabla\mathcal{L}(u_t)$ and an estimator P_t^n of the $\mathcal{H}$-orthogonal projector P_t onto $\mathcal{T}_t$, we then perform a linear update $\bar{u}_{t+1} := u_t - s_t P_t^n g_t$ in direction of the (empirically) projected negative gradient $-P_t^n g_t$. This yields the intermediate iterate $\bar{u}_{t+1}$. Since the $\bar{u}_{t+1}$ is not guaranteed to lie in the original model class $\mathcal{M}$, we perform a recompression step $u_{t+1} := R_t(\bar{u}_{t+1})$, where $R_t : \mathcal{H} \to \mathcal{M}$ takes the linear update $\bar{u}_{t+1}$ back to the model class $\mathcal{M}$ with a controllable error in the loss $\mathcal{L}$. The proposed algorithm can thus be presented in the two equations

$$\begin{aligned} \bar{u}_{t+1} &:= u_t - s_t P_t^n g_t, \qquad g_t := \nabla\mathcal{L}(u_t), \\ u_{t+1} &:= R_t(\bar{u}_{t+1}). \end{aligned}$$

We show that under certain assumptions on the loss $\mathcal{L}$ and the sequences of projectors P_t^n, step sizes s_t and recompressions R_t, the resulting optimisation scheme converges almost surely to a stationary point of the true loss. The corresponding rates of convergence are displayed in Table 1. The proposed algorithm exhibits the same convergence rates as classical *gradient descent* (GD) in the best case but can never perform worse than SGD. We pay particular attention to the estimation of the projectors P_t^n, which must be carried out using optimally weighted samples in order to achieve the presented rates.

	GD	Best-case	Worst-case	SGD
L-smoothness	$\mathcal{O}(t^{-1})$	$\mathcal{O}(t^{-1+\varepsilon})$	$\mathcal{O}(t^{-1/2+\varepsilon})$	$\mathcal{O}(t^{-1/2+\varepsilon})$
strong convexity	$\mathcal{O}(a^t)$	$\mathcal{O}(a^t)$	$\mathcal{O}(t^{1-2\varepsilon})$	$\mathcal{O}(t^{1-2\varepsilon})$

Table 1. Almost sure convergence rates for different algorithms with $\varepsilon \in (0, \frac{1}{2})$ and $a \in (0,1)$ depending on the chosen step size.

Tensor train approximation of deep transport maps

Sergey Dolgov

(joint work with Tiangang Cui, Robert Scheichl, Olivier Zahm and workshop participants)

A challenging example of high-dimensional functions is joint probability density (or distribution) functions of multiple random variables. Sampling and computation of expectations of high-dimensional random variables is one of the fundamental challenges in stochastic computation. We develop a deep transport map that is suitable for sampling concentrated distributions defined by an unnormalised density function [1]. We approximate the target distribution as the pushforward of an easy reference distribution under a composition of inverse Rosenblatt transformations of coordinates. Each transformation is formed by a tensor-train (TT) decomposition of a bridging density, which is a simplified version of the target density. This composition of maps moving along a sequence of bridging densities alleviates the difficulty of approximating the concentrated target density directly. In contrast to neural network layers, each Rosenblatt map is fully defined by its bridging density, and can be computed independently of next layers by fast TT cross algorithms. We propose two bridging strategies suitable for wide use: tempering of the target density with a sequence of increasing powers [1], and smoothing of an indicator function with a sequence of sigmoids of increasing scales [3]. The latter strategy opens the door to efficient computation of rare event probabilities in Bayesian inference problems. Numerical experiments on problems constrained by differential equations show little to no increase in the computational complexity with the event probability going to zero, and allow to compute hitherto unattainable estimates of rare event probabilities for complex, high-dimensional posterior densities.

One drawback of the TT decomposition though is its sensitivity to the order of variables. Probability density functions with locally correlated variables exhibit typically low TT ranks [4], whereas if the same variables are permuted in such a way that strongly correlated variables are far apart in the random vector, the TT ranks may increase up to an exponential factor. Permutation (or even better, rotation) of variables may significantly expand the applicability of TT-driven approximation methods to higher dimensions and more complicated functions. In principle, this is the problem that is tackled by the Rosenblatt map. However, if the initial dimension is very high, it may still be daunting to compute a TT approximation, even for simple bridging densities. In this case it may be useful to identify unimportant variables (e.g. those in which the function is almost constant), and truncate them altogether. If the function to be approximated is a posterior density function of exponential family, the eigenvalue decomposition of the information matrix computed from the gradient of the log-likelihood can be used to inform the permutation or rotation of variables [2]. This allowed us to solve a Bayesian inverse problem constrained by an elasticity PDE with a thousand of random variables.

Both techniques outlined above require a function to be of a probability density form to compute the Rosenblatt map or the information matrix. Efficient tensor methods for very high dimensional functions which are neither positive nor easily differentiable are still lacking. During the workshop, we have come up with an idea of learning a matrix of linear change of variables simultaneously with a low-rank TT decomposition from data such as random samples of the function. Preliminary experiments with simple functions demonstrated that a nearly optimal rotation of variables is achievable using a moderate amount of function evaluations. However, further research is needed to make this technique useful for higher dimensions and concentrated functions, sampling of which is difficult.

References

[1] T. Cui, S. Dolgov, *Deep Composition of Tensor-Trains Using Squared Inverse Rosenblatt Transports*, Foundations of Computational Mathematics **22** (2022), 1863–1922.

[2] T. Cui, S. Dolgov, O. Zahm, *Scalable conditional deep inverse Rosenblatt transports using tensor trains and gradient-based dimension reduction*, Journal of Computational Physics **485** (2023), 112103.

[3] T. Cui, S. Dolgov, R. Scheichl, *Deep importance sampling using tensor trains with application to a priori and a posteriori rare event estimation*, SIAM Journal on Scientific Computing (to appear 2023), `https://arxiv.org/abs/2209.01941`

[4] P. B. Rohrbach, S. Dolgov, L. Grasedyck, R. Scheichl, *Rank Bounds for Approximating Gaussian Densities in the Tensor-Train Format*, SIAM/ASA Journal on Uncertainty Quantification **10** (2022), 1191–1224.

Curse-of-dimensionality-free deep-learning approaches to deterministic control problems

Lars Grüne

(joint work with Dante Kalise, Luca Saluzzi, and Mario Sperl)

It is known that deep neural networks have the ability to represent certain classes of high-dimensional functions without being affected by the curse of dimensionality. One of these classes are the so-called Barron functions. However, the usual way to check that a function falls into this class is by checking suitable smoothness properties, which cannot be expected to hold for the functions to be approximated in typical deterministic control problems.

Another prominent function class for which the curse of dimensionality can be avoided, the so-called compositional functions, have recently been shown to be a promising system class for problems involving deterministic dynamical systems

[2, 4]. In this talk, we have explained the ability of the simplest functions in this class, the so-called separable functions, to approximate control Lyapunov functions and optimal value functions.

For control Lyapunov functions, the requirement of separability is closely linked to the kind of Lyapunov functions that can be obtained from nonlinear small-gain theory, which is used for this purpose in a control context e.g. in [1]. While this approach is in principle constructive, it suffers from the fact that the construction of the resulting control Lyapunov functions is quite complicated. Here neural networks can provide a remedy, because the theory is only used for designing the architecture of the network, while the actual separable structure is learned in the training process of the network [3]. More precisely, small-gain theory ensures the *existence* of a control Lyapunov function V of the separable form

$$V(x) = \sum_{i=1}^{s} V_i(z_i), \qquad \begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix} = Tx,$$

where the low-dimensional subvectors z_i are obtained from the original high-dimensional state vector x by some coordinate transformation T, but the *computation* of T and of the individual V_i is left to the training process of the neural network.

For optimal value functions, separability is in general a too demanding property, as exploiting the interaction between different subsystems is usually a prerequisite for achieving optimality. However, when the subsystems are connected via a graph, it seems reasonable to expect that subsystems that are far away (in terms of the graph distance) only interact with each other very weakly. This heuristic expectation can be made rigorous in the framework of decaying sensitivity [5] and exploited for a curse-of-dimensionality-free approximation of optimal value functions V via *overlapping separable* functions

$$V(x) = \sum_{i=1}^{s} W_i(z_i), \quad z_i = \begin{pmatrix} x_{j_1} \\ \vdots \\ x_{j_k} \end{pmatrix},$$

where each component x_j of the state vector may occur in several of the subvectors z_i but the number k of components appearing in each z_i is bounded independent of the overall dimension. Under an exponential sensitivity assumption, first rigorous error estimates for such an overlapping separable approximation were obtained in [6].

References

[1] Kaiwen Chen and Alessandro Astolfi, *On the Active Nodes of Network Systems*, Proceedings of the 59th IEEE CDC, Jeju Island, Republic of Korea, 2020, 5561–5566

[2] Lars Grüne, *Computing Lyapunov functions using deep neural networks*, Journal of Computational Dynamics **8** (2021), 131–152

[3] Lars Grüne and Mario Sperl, *Examples for existence and non-existence of separable control Lyapunov functions*, Proceedings of NOLCOS 2022, IFAC-PapersOnLine **56** (2023), 19–24

[4] Wei Kang and Qi Gong, *Feedforward Neural Networks and Compositional Functions with Applications to Dynamical Systems*, SIAM Journal on Control and Optimization **60** (2022), 786–813

[5] Sungho Shin, Mihai Anitescu, and Victor M. Zavala, *Exponential decay of sensitivity in graph-structured nonlinear programs*, SIAM Journal on Optimization **32**, (2023), 1156–1183

[6] Mario Sperl, Luca Saluzzi, Lars Grüne, and Dante Kalise, *Separable approximations of optimal value functions under a decaying sensitivity assumption*, Proceedings of the 62nd IEEE CDC 2023, to appear; arXiv 2304.06379, 2023

A statistical Tensor Train - POD approach for feedback boundary optimal control in fluid dynamics

Luca Saluzzi
(joint work with Sergey Dolgov and Dante Kalise)

Consider the optimal control problem

$$\begin{cases} \inf\limits_{u\in\mathcal{U}} J(u(\cdot,x)) := \int_0^{+\infty} y(s)^\top Q y(s) + u^\top(s) R u(s)\, ds\,, \\ \text{subject to } \dot y(s) = f(y(s)) + B(y(s))u(s), \quad s\in(0,+\infty), \end{cases} \tag{1}$$

where $y(0)=x$ and $\mathcal{U}=L^\infty([0,+\infty);U)$ is the set of admissible controls. For a given initial condition $x\in\mathbb{R}^d$, we define the value function associate to the optimal control problem (1) as

$$V(x) = \inf_{u\in\mathcal{U}} J(u(\cdot,x))$$

which, by standard dynamic programming arguments, satisfies the following Hamilton-Jacobi-Bellman PDE

$$\min_{u\in U}\left\{(f(x)+B(x)u)^\top \nabla V(x) + x^\top Q x + u^\top R u\right\} = 0, \quad x\in\mathbb{R}^d. \tag{2}$$

The HJB PDE (2) is a challenging first-order fully nonlinear PDE cast over $\mathbb{R}^d$, where d can be arbitrarily large, and thus intractable through conventional grid-based methods. However, in the unconstrained case, *i.e.* $U=\mathbb{R}^m$, the minimizer of the l.h.s. of eq. (2) can be computed explicitly as

$$u^*(x) = -\frac{1}{2} R^{-1} B(x)^\top \nabla V(x). \tag{3}$$

In this context we propose to approximate the value function together with its gradient in a *data-driven* approach, learning a surrogate model for the value function via adaptive sampling of the solution of the HJB (2). The synthetic data are generated via the so-called State-Dependent Riccati Equation (SDRE), an extension of the Riccati solution to nonlinear dynamics. By writing the dynamics in semilinear form

$$\dot y = A(y(t))y(t) + B(y(t))u(t), \tag{4}$$

equation (2) can be approximated as

$$A^\top(x)\Pi(x) + \Pi(x)A(x) - \Pi(x)B(x)R^{-1}B(x)^\top\Pi(x) + Q = 0\,, \tag{5}$$

which is obtained by applying the ansatz $V(x) = x^\top \Pi(x) x$ with a gradient approximation $\nabla V(x) \approx 2\Pi(x)x$. At this point, similarly to [1], the value function is represented in Functional Tensor Train (FTT) format

(6)

$$V(x) \approx \tilde{V}(x) := \sum_{\alpha_0=1}^{r_0} \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_d=1}^{r_d} G^{(1)}_{(\alpha_0,\alpha_1)}(x_1) \cdots G^{(k)}_{(\alpha_{k-1},\alpha_k)}(x_k) \cdots G^{(d)}_{(\alpha_{d-1},\alpha_d)}(x_d),$$

with

$$G^{(k)}_{(\alpha_{k-1},\alpha_k)}(x_k) = \sum_{i=1}^{n_k} \Phi^{(i)}_k(x_k) H^{(k)}_{(\alpha_{k-1},i,\alpha_k)},$$

where $\{\Phi^{(i)}_k(x_k)\}_{i=1}^{n_k}$ are prescribed basis functions and $\{r_k\}_{k=1}^{d}$ are called TT ranks.

Given certain sample points $\{x_i\}_{i=1}^N$ and the dataset $\{\, V(x_i),\, \nabla V(x_i)\}_{i=1}^N$ computed by SDRE, we are interested in determining the coefficient tensors

$\{H^{(1)}, \dots, H^{(d)}\}$ which characterize the FTT representation $\tilde{V}(x)$ introduced in (6), solving the regression problem

$$\min_{H^{(1)},\dots,H^{(d)}} \sum_{i=1}^{N} |\tilde{V}(x_i) - V(x_i)|^2 + \lambda \|\nabla \tilde{V}(x_i) - \nabla V(x_i)\|^2,$$

which is approximated by an alternating direction strategy and a TT *cross interpolation* technique [2, 5]. The TT Cross enables to adapt the sampling sets to minimize the conditioning of the interpolation problem, avoiding the evaluation of the function on the whole tensorial grid. The methodology has been successfully applied to the optimal control of a multi-agent system, where the TT ranks of the approximation of the value function presented a constant behaviour varying the dimension of the system, yielding an effective mitigation of the curse of dimensionality. However, the dimension of the value function is still that of the state space, leading to a very large number of unknowns in the approximation ansatz and training data. A possible way to tackle this problem is given by the application of Model Order Reduction (MOR) techniques. One of the most famous MOR method is the Proper Orthogonal Decomposition (POD), which synthesizes a set of snapshots capturing the behaviour of the system and looks for basis functions that capture the major variations in the data. In contrast to existing techniques, we propose a Statistical Proper Orthogonal Decomposition (SPOD) which takes into account controlled trajectories treating boundary conditions and initial condition as random variables. The corresponding reduced basis is chosen to minimize the empirical risk for the controlled solution, avoiding any linearisation of the dynamical system. Once computed the basis and projected the system, the reduced dynamics can be employed for either a fast online computation of the optimal control or an efficient synthesis of a dataset for the construction of a TT surrogate model. The methodology has been tested on the vorticity stabilization of the 2D Navier-Stokes equations, whose discretization employs several thousands of degrees of freedom.

References

[1] Dolgov, D. Kalise, and K. K. Kunisch, *Tensor Decomposition Methods for High-dimensional Hamilton–Jacobi–Bellman Equations*, SIAM Journal on Scientific Computing **43** (2021), A1625–A1650.

[2] S. Dolgov, D. Kalise and L. Saluzzi, *Data-Driven Tensor Train Gradient Cross Approximation for Hamilton–Jacobi–Bellman Equations*, SIAM Journal on Scientific Computing **45** (2023), A2153–A2184.

[3] S. Dolgov, D. Kalise and L. Saluzzi, *Statistical Proper Orthogonal Decomposition for model reduction in feedback control*, preprint.

[4] K. Kunisch, S. Volkwein, and L. Xie, *HJB-POD-based feedback design for the optimal control of evolution problems*, SIAM Journal on Applied Dynamical Systems, **3** (2004), 701–722.

[5] I. V. Oseledets and E. E. Tyrtyshnikov, *TT-cross approximation for multidimensional arrays*, Linear Algebra Appl. **432** (2010), 70–88.

A Mean-Field Optimal Control Approach to the Training of NeurODEs & AutoencODEs

Cristina Cipriani

(joint work with Benoît Bonnet, Massimo Fornasier, Hui Huang, Alessandro Scagliotti and Tobias Wöhrer)

In recent years, neural networks have emerged as a significant tool in artificial intelligence. However, there exists a pressing need for a robust mathematical framework to systematically analyze their intricate characteristics. A key theoretical advancement involves interpreting deep neural networks with residual connections (or shortcut connections) as dynamical systems, as outlined in the works [1] and [2]. The information flow from input to output in a network with an infinite number of layers can be expressed in the continuum limit as:

$$\dot{X}(t) = \mathcal{F}(X(t), \theta(t)),$$

This leads to nonlinear *neural ODEs* (NeurODEs), where time takes the role of the continuous-depth variable. This perspective allows the interpretation of neural network learning problems as continuous-time control problems, which provides access to the extensive literature of mathematical control theory, potentially enhancing the overall explainability of learning algorithms. Relevant works in this direction include [3] and [4].

Our work in [5] focuses on the mean-field formulation of the control problem, specifically addressing the scenario of an infinitely large dataset. We examine the evolution of the distribution μ_0 of initial data through the network as a partial differential equation, subsequently considering the corresponding mean-field optimal control problem. In [5], we establish first-order optimality conditions through a mean-field Pontryagin Maximum Principle, derived as a consequence of an abstract Lagrange multiplier rule in the Banach space of Radon measures.

However, it is crucial to note that NeurODEs encounter limitations when modeling neural networks with discrepancies in dimensionality between consecutive layers. Skip connections with identity mappings necessitate a ”rectangular” network shape, where the width of layers is uniform. To address this limitation

and enhance the network's capacity, we introduce a novel design of the vector field driving the dynamics in [6]. This continuous-time model accommodates various width-varying neural networks and builds upon insights from our previous work [5]. Furthermore, in [6] we extend our framework to encompass the low-Tikhonov regularization regime. For the continuous-time version of Autoencoders (AutoencODEs), we propose a novel discrete architecture and an alternative training method based on the Pontryagin Maximum Principle. To demonstrate the effectiveness of our approach, we present informative numerical examples offering valuable insights into the resulting algorithm.
Finally, we leverage the well-established theory of optimal control to address the lack of robustness in neural networks against data manipulation, commonly known as adversarial attacks. These attacks involve small changes of the inputs, which lead to significant modifications in the model outputs. In [7], we interpret the adversarially robust learning problem arising in machine learning as a minimax control problem

$$\min_u \mathbb{E}_{(x^0,y)\sim\mu}\left[\max_{\|\alpha\|\leq\epsilon} \text{Loss}(\theta, x^0+\alpha, y)\right],$$

where the initial data and labels (x^0, y) are drawn from an underlying data distribution μ, and $\text{Loss}(u, x^0, y)$ quantifies the prediction accuracy. We derive the Pontryagin Maximum Principle for this problem using separation of Boltyanski approximating cones, as presented in [8], and develop a numerical method to address the robust learning problem, which is used for low-dimensional examples.

References

[1] W. E. *A proposal on machine learning via dynamical systems.* Comm. Math. Stat., **1**1–11 (2017).

[2] E. Haber, and L. Ruthotto. *Stable architectures for deep neural networks.* Inverse problems, **34** (2018).

[3] A. Scagliotti. *Deep Learning Approximation of Diffeomorphisms via Linear Control Systems.* Mathematical control and related fields, 1-32 (2022).

[4] J.-F. Jabir, D. Šiška and Ł. Szpruch. *Mean-Field Neural ODEs via Relaxed Optimal Control.* arxiv preprint arXiv:1912.05475 (2021).

[5] B. Bonnet, C. Cipriani, M. Fornasier, and H. Huang. *A measure theoretical approach to the Mean-field Maximum Principle for training NeurODEs.* Nonlinear Analysis **227** (2023), 131-161.

[6] C. Cipriani, M. Fornasier, A. Scagliotti. *From NeurODEs to AutoencODEs: a mean-field control framework for width-varying neural networks.* arXiv:2307.02279 (2023).

[7] C. Cipriani, A. Scagliotti, and T. Wöhrer. *A minimax optimal control approach for robust neural ODEs.* arXiv preprint arXiv:2310.17584 (2023).

[8] M. Motta, and F. Rampazzo. *An Abstract Maximum Principle for con- strained minimum problems.* arXiv:2310.09845 (2023).

Spectral approximation of Lyapunov operator equations with applications in non-linear feedback control

Bernhard Höveler
(joint work with Tobias Breiten)

Let a (non-linear) dynamical system be given as

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}x(t) &= f(x(t)), \qquad \text{for } t \in (0,\infty) \\ \quad x(0) &= z \end{cases}$$

for some $f\colon \mathbb{R}^n \to \mathbb{R}^n$ and let us define the Lyapunov function v to a given cost $g\colon \Omega \subseteq \mathbb{R}^n \to \mathbb{R}_+$ as follows

$$v(z) := \int_0^\infty g(\Phi^t(z))\,\mathrm{d}t \qquad \text{for } z \in \Omega$$

where the flow $\Phi^t(z)$ is defined as the mapping from the initial value z to the state $x(t)$ with $x(0) = z$ at time t, i.e $\Phi^t(z) := x(t)$. Computing such a function is a challenging task both from the numerical as from the analytical side. One of the main numerical challenges arises, when n is large and therefore the system is high dimensional.
One of the main results of this talk is that we can define a weak-* continuous semigroup

$$\begin{array}{rccc} S^*(t)\colon & X^* & \to & X^* \\ & \phi & \mapsto & \phi \circ \Phi^t \end{array}$$

and that there exists a preadjoint $S(t)$. Here X and X^* are some specially weighted $\mathrm{L}^p(\Omega)$ spaces. The weighting assures the exponential decay under some assumptions. It is shown that – if the cost function g admits the decomposition $g(x) = \sum_{i=1}^\infty c_i(x)^2$ – the Lyapunov function v can be written as

$$v(x) = \sum_{i=1}^\infty p_i(x)^2$$

where p_i are the eigenfunctions of the symmetric bilinear form

$$\langle \phi, \psi \rangle_P = \int_0^\infty \langle C\phi, C\psi \rangle_{\ell_2}\ \mathrm{d}t \quad \text{with} \quad C\phi := \Big(\langle \phi, c_i \rangle_{X,X^*} \Big)_{i \in \mathbb{N}}.$$

Furthermore, it can be shown that the error to a finite rank approximation decays with a rate that is depending on the regularity of the c_i and f. Lastly, the generator A of the semigroup S can be used to show that P is the solution to an operator Lyapunov equation of the form

$$\langle A\phi, \psi \rangle_P + \langle \phi, A\psi \rangle_P + \langle C\phi, C\psi \rangle_{\ell_2} = 0 \qquad \text{for all } \phi, \psi \in \mathcal{D}(A) \subseteq X$$

which can be exploited for a numerical method. The proposed scheme relies on a low rank approximation and a splitting integrator to solve a corresponding time

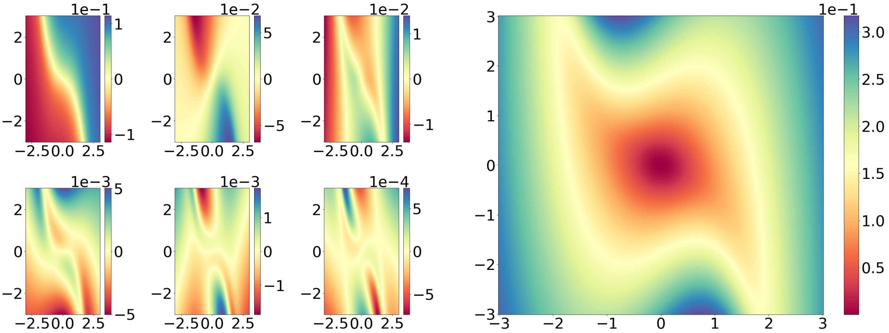

FIGURE 1. First six eigenfunctions (left) and the Lyapunov function (right) of a modified van der pool oscillator.

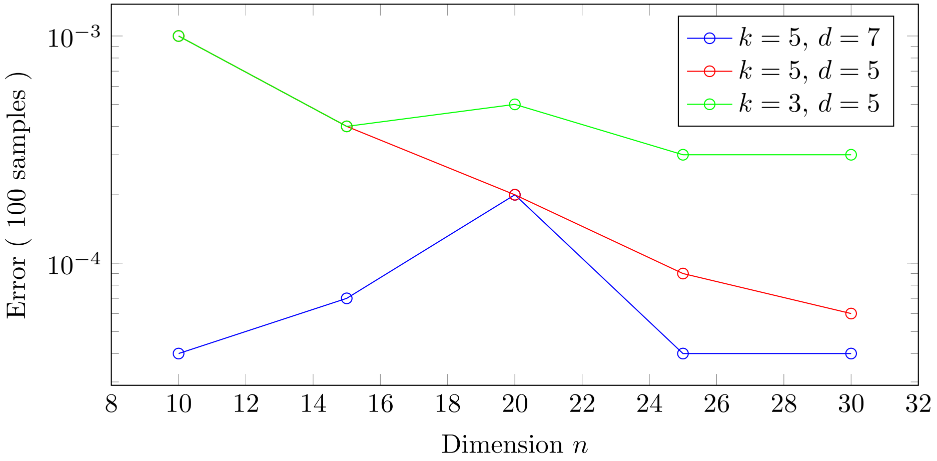

FIGURE 2. Maximum error of the proposed scheme applied to the discretized Allen Cahn model.

dependent problem. To overcome the curse of dimensionality tensor trains (TT) are used. This leads to an approximation of the Lyapunov function of the form

$$v_h(x_1, \dots, x_n) := \operatorname{Re} \sum_{j,j'}^{k} \prod_{i=1}^{n} G_i^{(j)}(x_i) \tilde{M}_{j,j'} \prod_{i=1}^{n} G_i^{(j')}(x_i)$$

where $G_i^{(j)} : [-1, 1] \to \mathbb{R}^{r_i^{(j)} \times r_{i+1}^{(j)}}$ are matrix valued functions for $j = 1, \dots, k$ and $i = 1, \dots, n$ while $\tilde{M} \in \mathbb{C}^{k \times k}$.

However, in contrast to a neural network the TT-approximation depends on the chosen basis and from an analytical standpoint it is not immediately clear what a good choice of basis might be. A possible mitigation might be to optimize over the choice of basis as well, which leads to an optimization over the Stiefel manifold.

Another area of interest is the inclusion of control. Ongoing research suggests that a non-linear operator equation similar to the Riccati equation is suitable.

$$\langle A\phi, \psi\rangle_P + \langle \phi, A\psi\rangle_P - \frac{1}{2}\sum_{k=1}^{\infty}\left(\langle M_k\phi, B_k\psi\rangle_P + \langle B_k\phi, M_k\psi\rangle_P\right) + \langle C\phi, C\psi\rangle_{\ell_2} = 0$$

Where:

$$B_k^*\phi := p_k b^\top \nabla\phi \quad \text{and} \quad M_k\phi := b^\top \nabla p_k \phi$$

However, the non-linear nature makes the analysis of this equation much more difficult.

References

[1] I. Osledets, *Tensor-Train Decomposition*, SIAM Journal on Scientific Computing, Vol. 33, Iss. 5 (2011)

[2] G. Certui,C. Lubich, *Time integration of symmetric and anti-symmetric low-rank matrices and Tucker tensors*, BIT Numerical Mathematics (2020)

[3] T. Breiten, B. Höveler, *On the Approximability of Koopman based Operator Lyapunov Equations*, SIAM Journal on Control and Optimization, Vol. 61, Iss. 5 (2023)

Functional SDE approximation inspired by a deep operator network architecture

Martin Eigel

(joint work with Charles Miranda)

We are concerned with the efficient generation of solution trajectories of SDEs by training a specific neural network (NN) architecture called SDEONet. This architecture is inspired by recent development in the area of operator learning, where operators in infinite dimensional spaces are represented with NNs. In particular, we refer to the analysis on deep operator networks (DeepONets) in [1]. These are composed of two NNs, a branch and a trunk network, representing learned basis coefficients (branch) of a linear combination of a learned reduced basis (trunk), respectively. To transfer this functional framework to the task of solving SDEs, we make use of the representability of any process $X_t \in L^2(\Omega, \mathcal{F}, \mathbb{P})$ in terms of a Wiener chaos expansion

$$X_t = \sum_{k\geq 0}\sum_{|\alpha|=k} x_\alpha(t) \underbrace{\prod_{i=1}^{\infty} H_{\alpha_i}\left(\int_0^T e_i(s)\,\mathrm{d}W_s\right)}_{\Psi_\alpha}, \tag{1}$$

with univariate Hermite polynomials H_n of degree n and a basis $(e_i)_{i\geq 1}$ of $L^2([0,T])$, which we choose to be the Haar basis. The coefficients x can be obtained by projection onto the Wiener chaos but also follow the dynamics of an

ODE [2]

$$\frac{\mathrm{d}x_\alpha}{\mathrm{d}t}(t) = \mu(t, X_t)_\alpha + \sum_{j=1}^{\infty} \sqrt{\alpha_j} e_j(t) \sigma(t, X_t)_{\alpha^-(j)}, \tag{2}$$

$$x_\alpha(0) = 1_{\alpha=0} x_0. \tag{3}$$

Our SDEONet architecture is a mapping from Brownian increments to the realization of the respective SDE trajectory as depicted in Figure 1 with input G consisting of integrals of e_i as in (1). It can hence be seen as an alternative approach to the standard Euler-Maruyama simulation scheme.

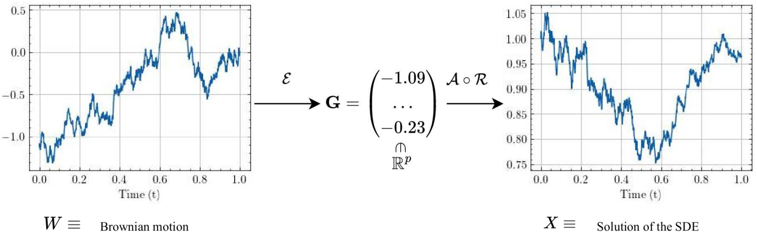
0.5
0.0
−0.5
−1.0
0.0 0.2 0.4 0.6 0.8 1.0
Time (t)
$W \equiv$ Brownian motion
$\mathcal{E}$
$\mathbf{G} = \begin{pmatrix} -1.09 \\ \cdots \\ -0.23 \end{pmatrix}$
$\Cap$
$\mathbb{R}^p$
$\mathcal{A} \circ \mathcal{R}$
1.05
1.00
0.95
0.90
0.85
0.80
0.75
0.0 0.2 0.4 0.6 0.8 1.0
Time (t)
$X \equiv$ Solution of the SDE

Figure 1. Sketch of SDE trajectory generation by the SDEONet architecture.

We consider the continuous stochastic process $(X_t)_{t\in[0,T]}$ that satisfies the stochastic differential equation (SDE)

$$\mathrm{d}X_t = \mu(t, X_t)\,\mathrm{d}t + \sigma(t, X_t)\,\mathrm{d}W_t, \text{ with } X_0 = x_0, \tag{4}$$

and $(W_t)_{t\in[0,T]}$ a Brownian motion on a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{P})$.

Figure 2 illustrates the representation of the functional mapping (of the stochastic process operator $\mathcal{G}$) by the SDEONet architecture. First, the encoder maps the Brownian increments W to $(G_i)_{i=0}^{m-1}$ with $G_i := \int_0^T e_i(t)\,\mathrm{d}W_t$. Second, the approximator maps $(G_i)_{i=0}^{m-1}$ to approximate polynomial chaos $\Psi_{k_j^*}$. These two operations constitute the branch net. The trunk net approximates the coefficient functions $x_{k_j^*}(t)$. The reconstructor uses branch and trunk to approximate the trajectory $(X_t^{m,p^*})_{t\in[0,T]}$.

For the convergence and complexity analysis, we consider a decomposition of the error E into (Wiener chaos) truncation [2], NN (Hermite) polynomial approximation [4] and reconstruction (with approximate ODE coefficients) [3],

$$E := \left(\int_0^T \mathbb{E}[|X_t - \widetilde{X}_t^{m,p}|^2]\,\mathrm{d}t \right)^{1/2} \leq E_{\text{Trunc}} + E_{\text{Approx}} + E_{\text{Recon}}.$$

For all three terms, convergence rates and NN complexity bounds can be derived.

$$\begin{tikzcd}
L^2([0,T]\times\Omega) \arrow[r, "\mathcal{G}"] \arrow[d, dashed, "\substack{\text{Encoder}\\ W\mapsto \mathbf{G}:=(G_1,\dots,G_m)}"'] & L^2([0,T]\times\Omega) \\
L^2(\Omega,\mathbb{R})^m \arrow[r, dashed, "\substack{\text{Approximator}\\ \mathbf{G}\mapsto(\widetilde{\Psi_1},\dots,\widetilde{\Psi_p})}"'] & L^2(\Omega,\mathbb{R})^p \arrow[u, dashed, "\substack{\text{Reconstructor}\\ \widetilde{\Psi}\mapsto\sum_{i=1}^{p}\tilde{x}_i\widetilde{\Psi_i}}"']
\end{tikzcd}$$

Figure 2. Diagram of the SDEONet operator mapping.

References

[1] S. Lanthaler, S. Mishra, Siddhartha, G. Karniadakis, *Error Estimates for DeepONets: A Deep Learning Framework in Infinite Dimensions*, Transactions of Mathematics and Its Applications **6** (2022).

[2] T. Huschto, M. Podolskij, S. Sager, *The Asymptotic Error of Chaos Expansion Approximations for Stochastic Differential Equations*, Modern Stochastics: Theory and Applications **32** (2019), 145–165.

[3] P. Petersen, F. Voigtlaender, *Optimal approximation of piecewise smooth functions using deep ReLU neural networks*, Neural Networks **108** (2018), 296–330.

[4] C. Schwab, J. Zech, *Deep Learning in High Dimension: Neural Network Expression Rates for Analytic Functions*, SIAM Journal on Uncertainty Quantification **11** (2023), 199–234.

Approximating Langevin Monte Carlo with ResNet-like Neural Network architectures

Charles Miranda

(joint work with Martin Eigel, Janina Schütte, David Sommer)

Deep Neural Networks (DNNs) have demonstrated their success in solving complex numerical problems, such as image classification, regression, kernel learning and solving partial differential equations (PDEs). Therefore, significant attention is given to establishing theoretical guarantees on the expressive abilities of DNNs. Deep neural networks (DNNs) have overcome the curse of dimensionality, especially when it comes to approximating Kolmogorov partial differential equations (PDEs) [1]. The latter demands a polynomial growth of parameters with the increase in dimension and expected precision, yet DNNs offer a potent workaround, thus presenting a remarkable achievement. Our study aims to sample from smooth log-concave probability distributions $\mathrm{d}\mu_\infty(x) \propto \exp(-V(x))\mathrm{d}x$. The primary objective is to create a deep neural network (DNN) with the ability to generate samples from the target distribution. The DNN's performance is evaluated based on the 2-Wasserstein distance, using input samples from a simple reference distribution such as the standard normal distribution. Our investigation is focused on the approximation of the Langevin Monte Carlo (LMC) algorithm, which is the

Euler-Maruyama discretisation of the stochastic differential equation

$$\mathrm{d}X_t = -\nabla V(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}W_t$$

through ResNet-like neural network structures

$$\begin{aligned} &x_0 := X_0 \\ &x_k := x_{k-1} + \phi_k(x_{k-1}) + \xi_k,\ x \in 1,\dots,K \end{aligned}$$

where ϕ_k are fully connected neural networks and ξ_k are i.i.d. standard normal random variable. Notably, we pay special attention to error analysis in the context of the 2-Wasserstein distance. The suggested approach emulates LMC by connecting feed-forward neural networks as above. The approximation of the drift term with epsilon accuracy occurs in an appropriate L^2 space established by the current law of the process. Namely, our analysis is done on the quantity

$$\| -\nabla V - \phi_{k+1}\|_{L^2_{\nu_k}(\mathbb{R}^d;\mathbb{R}^d)}$$

where ν_k is the law of x_k.

We demonstrate that if the above quantity is smaller than ε

$$\mathcal{W}_2(\mu_\infty, \nu_K) \leq (1-mh)^K \mathcal{W}_2(\mu_\infty, \mu_0) + \frac{7\sqrt{2}}{6}\frac{M}{m}\sqrt{hd} + \frac{1-(1-mh)^K}{m}\varepsilon$$

where m is the strong-convexity parameter of V, M the Lipschitz constant of ∇V and $h \leq 2/(m+M)$ the step size in the LMC algorithm. By exploiting the properties of the initial distribution, which is the standard Normal distribution, and of the ResNet-like architecture, we are able to show that the measures ν_k are sub-Gaussian. We show that the ε-accuracy can be achieved for all ϕ_k such that the number of parameters for the ResNet-like architecture is bounded by $K(N(d,r,\varepsilon/\sqrt{d},m,M) + 2d^2 + 2)$, where $N(d,r,\varepsilon/\sqrt{d},m,M)$ is the number of parameters for a single fully connected neural network ϕ to satisfy

$$\| -\nabla V - \phi\|_{L^\infty(B_r(0);\mathbb{R}^d)} \leq \frac{\varepsilon}{\sqrt{2d}}$$

where

$$r \in \mathcal{O}\left(d^{7/4}\varepsilon^{-1}(d^{9/4}\varepsilon^{-1})^{3(1.5^K-1)}\right)$$

Unfortunately, the aforementioned result indicates that the radius of the ball must increase exponentially in the number of steps. We conjecture that due to the strong convexity of V and the Lipschitz continuity of ∇V, there exists a neural network capable of approximating $-\nabla V$ with a linear error growth. The experiments also indicate that the proposed architecture can sample from μ_∞ with the same convergence rate even if the potential V is no longer strongly convex, such as in a Gaussian mixture.

References

[1] Jentzen A, Salimova D, Welti T., *A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients*, Communications in Mathematical Sciences 2021;19(5):1167–1205.

The implicit bias phenomenon in deep learning

Holger Rauhut

It is common in deep learning to use many more parameters than training examples. Despite traditional statistical wisdom, which would predict overfitting, the learned neural networks usually generalize well to new unseen data [17]. In this overparameterized setting many networks exist that interpolate the data exactly. They all lead to global minimizers of the empirical loss function, which sums up the losses of a neural network over the training data. In this situation, the employed optimization algorithm (usually variants of gradient descent or stochastic gradient descent), including hyperparameters such as initialization, step sizes etc., significantly influences the computed minimizer. This phenomenon is called **implicit bias** of the learning algorithm. It is puzzling that the implicit bias of (stochastic) gradient descent and variations is often towards solutions that generalize well. Although there is a growing research literature available, see e.g. [1, 2, 4, 3, 5, 9, 10, 11, 13, 15, 16, 17], many aspects of this phenomenon are not well understood yet.

One working hypothesis is that (stochastic) gradient descent with suitable initialization favors networks of low complexity, i.e., networks that can be represented with much fewer parameters than the number of trainable network weights. Low complexity may be understood in a broad sense here and it may be a challenge to determine suitable low complexity models for concrete types of data and network models. Examples may be sparse representations [3, 5, 7] as well as low rank matrix [1, 4] and tensor representations [14].

In order to gain theoretical understanding of the implicit bias phenomenon, it is useful to study simpler optimization problems that share two characteristics with the overparameterized deep learning scenario:

- many (infinite number of) global minimizers;
- a factorization/compositional structure.

In [3, 7] the problem of minimizing the function

$$L(x) = \frac{1}{2}\|Ax - y\|_2^2$$

is considered where $A \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$ with $m < n$. In this case, L has infinitely many global minimizer. In fact, if A has full rank, they form the affine subspace of solutions x to $Ax = y$. In order to induce a factorization structure we set

$$x = w^{(1)} \odot w^{(2)} \odot \cdots \odot w^{(N)},$$

where $(w^{(1)} \odot w^{(2)})_j = w_j^{(1)} w_j^{(2)}$ is the Hadamard product. This structure can be interpreted as a linear diagonal neural network. Plugging into the function L, we define

$$L^N(w^{(1)}, \ldots, w^{(N)}) = L(w^{(1)} \odot w^{(2)} \odot \cdots \odot w^{(N)}) \tag{1}$$

$$= \frac{1}{2}\|A(w^{(1)} \odot w^{(2)} \odot \cdots \odot w^{(N)}) - y\|_2^2. \tag{2}$$

Initializing identically with $w^{(\ell)}(0) = \alpha\mathbf{1}$, $\ell = 1, \dots, N$, where $\mathbf{1} = (1, 1, \dots, 1)^T$, we consider the gradient flow

$$\frac{d}{dt} w^{(\ell)}(t) = -\nabla_{w^{(\ell)}} L^N(w^{(1)}(t), \dots, w^{(N)}(t)), \quad \ell = 1, \dots, L.$$

We are interested in the convergence behavior and implicit bias of the product flow

$$v(t) = \prod_{\ell=1}^{N} w^{(\ell)}(t).$$

For identical initialization (as assumed) the vectors $w^{(\ell)}(t)$ remain identical, $w^{(1)}(t) = \cdots = w^{(N)}(t) = w(t)$, so that $v(t) = w^{\odot N}(t)$, where $w(t)$ is the gradient flow for

$$\widetilde{L}^N(w) = \frac{1}{2}\|Aw^{\odot N} - y\|_2^2.$$

Theorem. Assume that $S_+ = \{z \geq 0 : Az = y\}$ is nonempty, and let $N \geq 3$. Then $v_\infty = \lim_{t\to\infty} v(t) = w^{\odot N}(t)$ exists and $v_\infty \in S$. Let $Q = \min_{z \in S_+} \|z\|_1$ and $\beta = \|v(0)\|_1 = \alpha\sqrt{n}$. If $\beta < Q$ then

$$\|v_\infty\|_1 - Q \leq N \left(\frac{\beta}{Q}\right)^{1-\frac{2}{N}} Q.$$

Since ℓ_1-minimization promotes sparse solutions, see e.g. [8], this result basically states that the implicit bias of gradient flow is towards sparse solutions if the initialization scale α is small enough compared to the ℓ_1-norm of the ℓ_1-minimizer.

This result can be extended to the recovery of vectors with not necessarily nonnegative coefficients by using a difference of two factorizations, i.e., $v = w_1^{\odot N} - w_2^{\odot N}$, see [3] for details. Furthermore, by splitting $w = ru$, where r is a scalar und u is a vector on the unit sphere, and considering the gradient flow for both r and w with different learning rates – also referred to as weight normalization – gives similar results [5] as stated in the theorem above, however, allowing for larger initialization scale α, which leads to faster convergence.

In order to make a step closer to realistic neural networks, deep linear fully connected networks of the form $V = W^{(N)} \cdots W^{(1)}$ are considered in several works [1, 4, 6, 11, 15]. The current results suggest implicit towards low rank solutions, but a theorem similar to the one stated above is not yet available.

Of course, the next step will be to extend to nonlinear networks. Preliminary results for two-layer networks are available, see e.g. [12], but in general the understanding of the implicit bias phenomenon in deep learning is widely open.

References

[1] S. Arora, N. Cohen, W. Hu, and Y. Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems* (2019), 7413–7424.

[2] S. Azulay, E. Moroshko, M. S. Nacson, B. E. Woodworth, N. Srebro, A. Globerson, and D. Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In *International Conference on Machine Learning* (2021), pages 468–477.

[3] H.-S. Chou, J. Maly, H. Rauhut. More is less: Inducing sparsity via overparameterization. *Information and Inference: A Journal of the IMA 12:3* (2023), 1437–1460.

[4] H.-S. Chou, C. Gieshoff, J. Maly, H. Rauhut. Gradient descent for deep matrix factorization: Dynamics and implicit bias towards low rank. *Applied and Computational Harmonic Analysis 68* (2024), DOI:https://doi.org/10.1016/j.acha.2023.101595.

[5] H.-S. Chou, H. Rauhut, R. Ward. Robust implicit regularization via weight normalization. *Preprint arXiv:2305.05448* (2023).

[6] N. Cohen, G. Menon, Z. Veraszto. Deep linear networks for matrix completion – An infinite depth limit SIAM Journal on Applied Dynamical Systems, to appear. arXiv:2210.12497.

[7] M. Evan, S. Pesme, S. Gunasekar, N. Flammarion (2023). (S)GD over diagonal linear networks: Implicit regularisation, large stepsizes and edge of stability. *Preprint arXiv:2302.08982* (2023).

[8] S. Foucart, H. Rauhut *A Mathematical Introduction to Compressive Sensing.* Birkhäuser (2013).

[9] D. Gissin, S. Shalev-Shwartz, and A. Daniely. The implicit bias of depth: How incremental learning drives generalization. *International Conference on Learning Representations (ICLR).*, 2020.

[10] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Characterizing implicit bias in terms of optimization geometry. *Proc. ICML, PMLR **80*** (2018), 1832–1841.

[11] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems* (2017), 6151–6159.

[12] Y. Kou, Z. Chen, Q. Gu. Implicit bias of gradient descent for two-layer ReLU and leaky ReLU networks on nearly-orthogonal data. *Preprint arXiv:2310.18935* (2023).

[13] B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro. Geometry of optimization and implicit regularization in deep learning. *Preprint arXiv:1705.03071* (2017).

[14] N. Razin, A. Maman, and N. Cohen. Implicit regularization in hierarchical tensor factorization and deep convolutional neural networks. *arXiv:2201.11729* (2022).

[15] N. Razin and N. Cohen. Implicit regularization in deep learning may not be explainable by norms. In *Advances in Neural Information Processing Systems* (2020), 21174–21187.

[16] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research* 19:1 (2018), 2822–2878.

[17] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals. Understanding deep learning requires rethinking generalization. *Communications of the ACM **64:3*** (2016), 107–115.

The Role of Statistical Theory in Understanding Deep Learning

Sophie Langer

(joint work with Alina Braun, Gabriel Clara, Michael Kohler,
Johannes+Schmidt-Hieber, Harro Walk)

In recent years, there has been a surge of interest across different research areas to improve the theoretical understanding of deep learning (see, e.g., [1] and [8]). A particular promising approach is the statistical one, which interprets deep learning as a nonlinear or nonparametric generalization of existing statistical models. For instance, a simple fully connected neural network is equivalent to a recursive generalized linear model with a hierarchical structure. Given this connection, many papers in recent years derived convergence rates of neural networks in a nonparametric regression or classification setting (see, e.g., [12], [3], [10]) . Nevertheless, phenomena like overparameterization seem to challenge the statistical principle of bias-variance trade-off (see [15]). Therefore, deep learning cannot only be explained by existing techniques of mathematical statistics but also requires a radical overthinking. In this talk, we will delve into the dual aspects of the role statistics plays in comprehending deep learning: its significance and its limitations, emphasizing the need to bridge with other research domains. Our discussion centeres around three distinct topics:

Empirical risk minimizers vs. estimators learned by gradient descent. The statistical performance of deep neural networks is often analyzed within a non-parametric regression framework. The objective here is to construct an estimator m_n for the true regression function m such that

$$\mathbf{E} \int |m_n(x) - m(x)|^2 P_X(dx) \tag{1}$$

is *small* with a particular interest in the behavior of the bound as the number of data points n increases - the rate of convergence. Previous studies (see, e.g., [12], [3], [10]) adopted the empirical risk minimizer

$$m_n \in \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} |f(X_i) - Y_i|^2,$$

based on a specific class of neural networks. For this kind of estimators rate of convergence results were derived under different assumptions on m, which all have in common that the rate, i.e., the bound on (1), no longer depends on the input dimension d of the problem but on a lower dimension d^* and thus promises fast convergence even in high-dimensional spaces. While these results show interesting approximation and generalisation results for neural networks, they are subject to a fundamental problem: they sidestep the optimization process of neural networks by assuming an empirical risk minimizer, limiting the holistic understanding of the procedure. To adress this gap, we showed in a simplified setting (see [4]), i.e., for regression functions with suitable decaying Fourier transform (similar to the so-called Barron class in [2]) and for shallow neural networks with sigmoidal activation function a rate of convergence of $n^{-1/2}$. While these results offer hope for a statistical analysis that considers training, they also underscore the indispensability of integrating optimization considerations, especially for deeper network structures and less restrictive assumptions on the regression function.

Understanding dropout in a linear model. Overparameterized neural networks have gained significant attention in recent years due to their remarkable ability to achieve high accuracy on complex tasks. However, these networks are prone to overfitting, where they memorize the training data rather than learning the underlying patterns. To address this issue, researchers have developed various regularization schemes. In addition to explicit regularization techniques such as ℓ_2- or ℓ_2-penalization, algorithmic regularization approaches have been employed. Among them, dropout has emerged as a technique that randomly drops neurons during training, and it has demonstrated its effectiveness in various applications (see [13]). However, despite its empirical success, a comprehensive theoretical understanding of how dropout achieves regularization is still somewhat limited.

In the case of a linear model, it was shown that under an averaged form of dropout the least squares minimizer performs a weighted variant of ℓ_2-penalization. In turn, the heuristic "dropout performs ℓ_2-penalization" has even made it in popular textbooks (see [6] and [7]). We challenge this relation by investigating the statistical behavior of iterates generated by gradient descent with dropout (see [5]). In particular, non-asymptotic convergence rates for the expectation and covariance matrices of the iterates are derived. While in expectation the connection between dropout and ℓ_2-penalization can be verified, we show sub-optimality of the asymptotic variance compared to the estimator resulting from direct minimization of averaged dropout. To us, this result highlights once again, that simplification in analyzing deep learning can also lead to wrong conclusions.

Statistical analysis of image classification. The availability of massive image databases resulted in the development of scalable machine learning methods such as convolutional neural network (CNNs) filtering and processing these data. While the very recent theoretical work on CNNs focuses on standard nonparametric denoising problems, the variability in image classification datasets does, however, not originate from additive noise but from variation of the shape and other characteristics of the same object across different images. To address this problem, we consider a simple supervised classification problem for object detection on grayscale images (see [11]). While from the function estimation point of view, every pixel is a variable and large images lead to high-dimensional function recovery tasks suffering from the curse of dimensionality, increasing the number of pixels in our image deformation model enhances the image resolution and makes the object classification problem easier. We propose and theoretically analyze two different procedures. The first method estimates the image deformation by support alignment. Under a minimal separation condition, it is shown that perfect classification is possible. The second method fits a CNN to the data. We derive a rate for the misclassification error depending on the sample size and the number of pixels. Both classifiers are empirically compared on images generated from the MNIST handwritten digit database. The obtained results corroborate the theoretical findings. To us, the introduced image deformation model offers a new way of analyzing image classification theoretically with rates of convergence that are in line with practical observations. Furthermore, it highlights the necessity of critically questioning and revising existing statistical models.

References

[1] P. Bartlett, A. Montanari and A. Rakhlin, *Deep learning: A statistical viewpoint*, Actia Numerica **30** (2021), 87–201.

[2] A. Barron, *Approximation and estimation bounds for artificial neural networks*, Machine Learning **14** (1994), 115–133

[3] B. Bauer and M. Kohler, *On deep learning as a remedy for the curse of dimensionality in nonparametric regression*, Annals of Statistics **47** (2019), 2261–2285.

[4] A. Braun, M. Kohler, S. Langer and H. Walk, *Convergence rates for shallow neural networks learned by gradient descent*, Bernoulli **30** (2024), 475–502.

[5] G. Clara, S. Langer and J. Schmidt-Hieber, *Dropout Regularization Versus ℓ_2-Penalization in the Linear Model*, Arxiv preprint, arXiv:2306.10529

[6] B. Efron and T. Hastie, *Computer Age of Statistical Inference* (2021), Cambridge University Press

[7] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning* (2016), MIT press

[8] P. Grohs and G. Kutyniok, (Eds.), *Mathematical Aspects of Deep Learning.* (2022), Cambridge University Press

[9] L. Györfi, M. Kohler, A. Krzyzak and H. Walk, *A Distribution-Free Theory of Nonparametric Regression*(2002), Springer

[10] M. Kohler and S. Langer, *On the rate of convergence of fully connected deep neural network regression estimates*, Annals of Statistics **49** (2021), 2231-2249.

[11] S. Langer and J. Schmidt-Hieber, *A statistical analysis of an image classification problem*, Arxiv Preprint (2022), arXiv:2206.02151

[12] J. Schmidt-Hieber, *Nonparametric regression using deep neural networks with ReLU activation function*, Annals of Statistics **48** (2020), 1875–1897

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, Journal of Machine Learning Research **15** (2014), 1929 – 1958

[14] C. J. Stone, *Optimal Global Rates of Convergence for Nonparametric Regression*, Annals of Statistics **10**(4)(1982), 1040–1053

[15] C. Zhang, S. Bengio, M. Hardt, B. Recht and O. Vinyals, *Understanding deep learning requires rethinking generalization*, in ICLR (2017).

Optimal sampling and tensor learning

Anthony Nouy
(joint work with Robert Gruhlke, Bertrand Michel, Charles Miranda, Philipp Trunschke)

We consider the approximation of functions in L^2 from point evaluations, using linear or nonlinear approximation tools. For linear approximation, recent results show that weighted least-squares projections allow to obtain quasi-optimal approximations with near to optimal sampling budget [1, 2]. This can be achieved by drawing i.i.d. samples from suitable distributions (depending on the linear approximation tool) and subsampling methods. In a first part of this talk, we review different strategies based on i.i.d. sampling and present alternative strategies based on repulsive point processes that allow to achieve the same task with a reduced sampling complexity. In a second part, we show how these methods can be used to approximate functions with nonlinear approximation tools, in an active learning setting, by coupling iterative algorithms and optimal sampling methods for the projection onto successive linear spaces. We particularly focus on the approximation using tree tensor networks, an approximation tool with high expressive power [3, 4] and with an architecture allowing for an efficient implementation of optimal sampling procedures within coordinate descent algorithms.

References

[1] M. Dolbeault, D. Krieg, and M. Ullrich. *A sharp upper bound for sampling numbers in L^2*, Applied and Computational Harmonic Analysis, **63** (2023), 113–134.

[2] C. Haberstich, A. Nouy, and G. Perrin. *Boosted optimal weighted least-squares,* Mathematics of Computation, **91(335)** (2022), 1281–1315.

[3] M. Ali and A. Nouy. *Approximation theory of tree tensor networks: Tensorized univariate functions,* Constructive Approximation, (2023), pages 1–82.

[4] B. Michel and A. Nouy. *Learning with tree tensor networks: Complexity estimates and model selection.* Bernoulli, **28(2)** (2022), 910–936.

Low-rank tensor solvers for high-dimensional parabolic PDEs

Markus Bachmayr

(joint work with Henrik Eisenmann, Manfred Faldum, Emil Kieri, André Uschmajew)

In this talk, we consider two different approaches for numerically solving second-order parabolic initial value problems on high-dimensional product domains using low-rank tensor approximations. A typical model problem takes the form

$$\partial_t u - \nabla \cdot M \nabla u = f \quad \text{in } (0,T) \times \Omega = \Omega_1 \times \cdots \times \Omega_d, \tag{1}$$

subject to the initial condition $u(0,\cdot) = u_0$ in Ω and the boundary condition $u = 0$ on $(0,T) \times \partial\Omega$. As the following results show, using methods based on low-rank approximations of solutions this problem can be treated also for large d.

The two types of low-rank approximations that we consider are conceptually quite different, one based on dynamical low-rank approximation, the other on an adaptive solver for a space-time variational formulation. In both cases, we assume a Gelfand triplet $V \subset H \subset V'$, where in the case of (1), $V = H^1_0(\Omega)$ and $H = L_2(\Omega)$. In the first approach based on dynamical low-rank approximation, one obtains approximate dynamics under the additional constraint that for all times $t \in [0,T)$, one has $u(t) \in \mathcal{M}$, where $\mathcal{M}$ is a manifold of low-rank tensors such as $\mathcal{M} = \bigl\{ \sum_{i=1}^r \phi^1_k \otimes \phi^2_k : \ \phi^1_k \in L_2(\Omega_1), \phi^2_k \in L_2(\Omega_2) \bigr\} \subset H$ in the case $d = 2$.

The Dirac-Frenkel variational principle then yields an accordingly projected problem, which as shown in [3] can also be formulated in a weak formulation of (1): Given $f \in L_2(0,T;V')$ and $u_0 \in \mathcal{M} \cap H$, find $u \in W(0,T;V,V') = \{u \in L_2(0,T;V) \colon u' \in L_2(0,T;V')\}$ such that for almost all $t \in [0,T]$,

$$\begin{aligned} &u(t) \in \mathcal{M}, \\ &\langle u'(t) + A(t)u(t), v\rangle = \langle f(t), v\rangle \quad \text{for all } v \in T_{u(t)}\mathcal{M} \cap V, \\ &u(0) = u_0, \end{aligned} \tag{2}$$

where $T_{u(t)}\mathcal{M}$ denotes the tangent space at $u(t)$ and where $A(t) \colon V \to V$ is the elliptic part of the operator, assumed to be Lipschitz continuous with respect to t. Under natural conditions on $\mathcal{M}$ and the additional regularity requirements $f \in L_2(0,T;H)$ and $u_0 \in \mathcal{M} \cap V$, in addition to a splitting of $A(t) = A_1(t) + A_2(t)$ where $A_1(t)$ maps $\mathcal{M}$ to the respective tangent space and $A_2(t)$ satisfies a suitable boundedness condition as a mapping from $\mathcal{M} \cap V$ to H, in [3] we obtain existence and uniqueness of solutions $u \in W(0,T^*;V,H) \cap L_\infty(0,T^*;V)$ whenever u_0 has positive distance from the boundary of $\mathcal{M}$. Here either $T = T^*$ or $u(t)$ approaches the boundary of $\mathcal{M}$ as $t \to T^*$. In [4], this result is shown to be applicable to manifolds $\mathcal{M}$ of tensor trains and hierarchical tensors in H, and thus to problems with large d. We also show the resulting approximation to be stable with respect to perturbations of the problem data and that spatial semidiscretizations converge under natural assumptions.

Numerical solvers with favorable properties are available for the reduced problems on $\mathcal{M}$ defined by (2). However, with this approach in general one cannot ensure that the solutions of (2) are close to the unconstrained evolution given by $u'(t) + A(t)u(t) = f(t)$; as a simple example, one obtains a systematic error when $u_0 \perp_H f(0)$. Ensuring that such effects are avoided is difficult in practice. Such issues do not arise in the second approach that we consider.

This alternative construction of a low-rank solver for parabolic problems such as (1) is based on a space-time variational formulation. In the basic case of the heat equation, it reads: with $\mathcal{X} = W(0,T;V,V')$ and $\mathcal{Y} = L_2(0,T;V) \times H$, find $u \in \mathcal{X}$ such that for all $(v,w) \in \mathcal{Y}$,

$$\begin{aligned}(3)\quad \int_0^T \langle \partial_t u, v\rangle_{V',V} + \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x\,\mathrm{d}t + \int_\Omega \gamma_0 u\, w \,\mathrm{d}x \\ = \int_0^T \int_\Omega f\, v \,\mathrm{d}x\,\mathrm{d}t + \int_\Omega u_0\, w\,\mathrm{d}x,\end{aligned}$$

where $\gamma_0 u \in H$ denotes the inital trace of u. Here we restrict ourselves to the model case $\Omega = (0,1)^d$ for simplicity. Similarly to [6], our approximations of u are based on basis functions $\{\theta_\mu\}_{\mu\in\mathcal{I}}$ on $(0,T)$ with the Riesz basis properties $\|\mathbf{v}\| \eqsim \big\| \sum_{\mu\in\mathcal{I}} \mathbf{v}_\mu \frac{\theta_\mu}{\|\theta_\mu\|_S} \big\|_S$ for all $\mathbf{v} \in \ell_2(\mathcal{I})$ and $S \in \{L_2(0,T), H^1(0,T)\}$ and $\{\psi_\nu\}_{\nu\in\mathcal{J}}$ on $(0,1)$ such that $\|\mathbf{v}\|_{\ell_2(\mathcal{J})} \eqsim \big\| \sum_{\hat\nu\in\mathcal{J}} \mathbf{v}_{\hat\nu} \frac{\psi_{\hat\nu}}{\|\psi_{\hat\nu}\|_S} \big\|_S$ for all $\mathbf{v} \in \ell_2(\mathcal{J})$ and $S \in \{H^1_0(0,1), L_2(0,1), H^{-1}(0,1)\}$. A concrete example of suitable such basis functions is provided by spline (multi-)wavelets.

A novel aspect in the method that we obtain in [5] is that we combine a sparse expansion in time with adaptive low-rank approximations in the spatial variables. Specifically, we compute approximations of u in the form

$$(4)\quad u(t,x_1,\ldots,x_d) \approx \sum_{\mu\in\Lambda_{\mathrm{t}}\subset\mathcal{I}} \theta_\mu(t) \sum_{(\nu_1,\ldots,\nu_d)\in\Lambda_\mu} \mathbf{u}_{\mu,\nu_1,\ldots,\nu_d}\, d^{\mathcal{X}}_{\mu,\nu} \psi_{\nu_1}(x_1)\cdots\psi_{\nu_d}(x_d)$$

with finite $\Lambda_{\mathrm{t}} \subset \mathcal{I}$ and $\Lambda_\mu = \Lambda^1_\mu \times \cdots \times \Lambda^d_\mu \subset \mathcal{J} \times \cdots \times \mathcal{J}$ that are potentially different for each μ. Here the coefficient tensors $\mathbf{u}_\mu = (\mathbf{u}_{\mu,\nu_1,\ldots,\nu_d})_{\nu\in\Lambda_\mu}$ are represented in hierarchical tensor format separately for each μ.

Based on a generalization of the strategy with a single hierarchical tensor representation of the approximate solution developed in [2] (see also [1]) for elliptic problems, an adaptive solver operating on the Riesz basis representation of the problem is obtained in [5] that refines the index sets Λ_{t} and Λ_μ, $\mu \in \Lambda_{\mathrm{t}}$, while at the same time computing approximate coefficient tensors $\mathbf{u}_\mu$ with adaptively adjusted ranks. A central role is played by suitable low-rank approximations of the scaling factors $d^{\mathcal{X}}_{\mu,\nu}$ in (4) that yield the appropriate normalization to a Riesz basis of $\mathcal{X}$. These can be chosen as

$$d^{\mathcal{X}}_{\mu,\nu} = \frac{\|\psi_{\nu_1} \otimes \cdots \otimes \psi_{\nu_d}\|_{H^1}}{\|\psi_{\nu_1} \otimes \cdots \otimes \psi_{\nu_d}\|^2_{H^1} + \|\theta_\mu\|_{H^1}}.$$

For each fixed μ and $a_\mu = \|\theta_\mu\|_{H^1}$, low-rank approximations by exponential sums of these expressions are obtained by applying quadrature to the integral representations

$$\frac{\sqrt{s}}{s + a_\mu} = \int_0^\infty \frac{1}{\sqrt{\pi y}} \bigl(1 - 2\sqrt{a_\mu y} F(\sqrt{a_\mu y})\bigr) \exp\bigl(-ys\bigr)\, \mathrm{d}y, \quad s > 0,$$

where F is the Dawson function, and by setting $s = \|\psi_{\nu_1} \otimes \cdots \otimes \psi_{\nu_d}\|^2_{H^1} = \sum_{i=1}^d \|\psi_{\nu_i}\|^2_{H^1}$. This yields approximate low-rank diagonal preconditioning for (3).

The resulting method can always be guaranteed to converge in $\mathcal{X}$-norm to the exact solution u of (3). Under benchmark approximability assumptions on the problem data and on u, it is also shown to yield approximations with optimality properties analogous to those obtained for the elliptic case in [2], especially on the arising tensor ranks. In particular, the curse of dimension can be avoided both concerning the complexity of approximations and the required number of operations in their computation. This is confirmed by the numerical tests in [5], where the total computational costs are observed to grow polynomially in d in the case of the heat equation as in (3).

References

[1] M. Bachmayr, *Low-rank tensor methods for partial differential equations*, Acta Numerica **32** (2023), 1–121.

[2] M. Bachmayr and W. Dahmen, *Adaptive low-rank methods: Problems on Sobolev spaces*, SIAM J. Numer. Anal. **54** (2016), 744–796.

[3] M. Bachmayr, H. Eisenmann, E. Kieri and A. Uschmajew, *Existence of dynamical low-rank approximations to parabolic problems*, Mathematics of Computation **90** (2021), 1799–1830.

[4] M. Bachmayr, H. Eisenmann and A. Uschmajew, *Dynamical low-rank tensor approximations to high-dimensional parabolic problems: existence and convergence of spatial discretizations*, arXiv Preprint arXiv:2308:16720 (2023).

[5] M. Bachmayr and M. Faldum, *A space-time adaptive low-rank method for high-dimensional parabolic partial differential equations*, arXiv Preprint arXiv:2302.01658 (2023).

[6] C. Schwab, and R. Stevenson, *Space-time adaptive wavelet methods for parabolic evolution problems*, Mathematics of Computation **78** (2009), 1293–1318.

Parametric PDE-induced Neural Networks and Network Training by Hierarchical Tensors

Thong Le

(joint work with Martin Eigel, Lars Grasedyck, Janina Enrica Schütte)

In our research, we investigate the potential of integrating low-rank tensor decompositions in neural network training. Our approach involves discretizing the loss function

$$\mathcal{L}_\Phi \,:\, \mathbb{R}^d \mapsto \mathbb{R}, \quad W \mapsto \mathcal{L}_\Phi(W).$$

with a grid of size n^d and afterwards finding the position of the minimum absolute entry which corresponds to the weights of the neural network. Calculating all entries is not possible because of the curse of dimensionality so we make use of the Hierarchical Tucker format to circumvent the curse of dimensionality. This not only enhances the networks' ability to optimize but could also facilitate more effective weight initialization, potentially leading to better network training. There are two different approaches one could choose:

- First idea is to create a fine grid in order to better approximate the minimum loss value but this would lead to higher n,
- Second idea is to use a grid refinement strategy to adaptively approach the minimum loss value which could be done with small n.

In this workshop we focused on the latter idea.

Furthermore we propose an idea to construct Feedforward Neural Networks using hierarchical domain decompositions of the parameter field of the parametric PDE which in our case is a cookie-shaped domain.

Our focus throughout the workshop is the Darcy partial differential equation as the model problem within a cookie-shaped parameter domain. Using this model problem we provide numerical results.

References

[1] J. Ballani and L. Grasedyck, *Tree adaptive approximation in the hierarchical tensor format*, SIAM Journal on Scientific Computing, **36(4):A1415–A1431** (2014)

[2] J. Ballani and L. Grasedyck, *Hierarchical tensor approximation of output quantities of parameter-dependent pdes*, SIAM/ASA Journal on Uncertainty Quantification **3(1):852–872** (2015)

[3] L. Grasedyck, L. Juschka and C. Löbbert, *Finding entries of maximum absolute value in low-rank tensors*, arxiv **1912.02072** (2019)

Convolutional neural networks for parametric PDEs

Janina Schütte

(joint work with Martin Eigel)

Deep learning has emerged as a flexible tool, extending its reach beyond famous applications, such as in natural language processing and image recognition, into the realm of solving parametric partial differential equations (pPDEs).

The significance of solving pPDEs lies in their crucial role across diverse fields such as physics, engineering, finance, and environmental science. Understanding the impact of varying parameters on a system is essential for predicting outcomes and making informed decisions.

Deep learning offers a novel approach to tackle the complexity of pPDEs. By training neural networks on appropriate data sets, the models learn intricate relationships between parameters and the corresponding system behavior. This expedites the solution process and therefore provides a chance to observe different states of the system under the influence of many different parameters.

There exist well developed mathematical concepts to solve PDEs specifically finite element (FE) and finite volume methods. There are works incorporating these methods into the setting of parametric PDEs, such as the Adaptive Stochastic Galerkin FEM [2] or the Variational Monte Carlo method [3], which are based on a polynomial chaos expansion and tensor approximation. A method based on convolutional neural networks (CNNs) was proposed in [1].

Parametric Darcy problem. The introduced methods are sample based and can be applied to data generated with a large class of linear and nonlinear pPDEs. In the analysis, the focus lies on the *parametric Darcy problem*, or stationary diffusion equation, which we also use as a benchmark problem in the numerical experiments. We formulate it in the following way. Let $D \subset \mathbb{R}^d$ be a spatial domain and $\Gamma \subset \mathbb{R}^{\mathbb{N}}$ a possibly countable infinite parameter space. Let $f : D \to \mathbb{R}$. We approximate the map $u : \Gamma \times D \to \mathbb{R}$, which satisfies

$$\begin{cases} \nabla_x \cdot (\kappa(y,x)\nabla_x u(y,x)) = f(x) & \text{for } x \in D \text{ and} \\ u(x) = 0 & \text{for } x \in \partial D \end{cases} \tag{1}$$

for the parameter field $\kappa : \Gamma \times D \to \mathbb{R}$ and where the derivatives are applied to the variable x.
The dependence of the parameter field κ on the parameter vector y can be characterized in different ways. For instance, for the *cookie problem*, the parameter field is defined for $D = [0,1]^2$ and $\Gamma = [0,1]^p$. Let $y \in \Gamma$ with $y_k \sim U[0,1]$ for $k = 1, \dots, p$ and define

$$\kappa(x,y) = a_0 + \sum_{k=1}^{p} y_k \chi_{D_k}(x),$$

where D_k are disks with fixed centers and radii and $a_0 > 0$ is constant. A visualization of the cookie parameters and the corresponding solutions can be seen in figure 1 in the top and bottom row, respectively.

CNN approximating an adaptive finite element method. To solve this problem a CNN architecture is proposed, which maps the coefficients of a FE discretization of $\kappa(y, \cdots)$ to those of $u(y, \cdot)$. For a FE space V_h we denote the interpolation of $\kappa(y, \cdot)$ into V_h by κ_h and the Galerkin projection of the solution of problem (1) $u(y, \cdot)$ onto V_h by $h_h(y, \cdot)$. Well suited $P1$ finite element spaces V_h

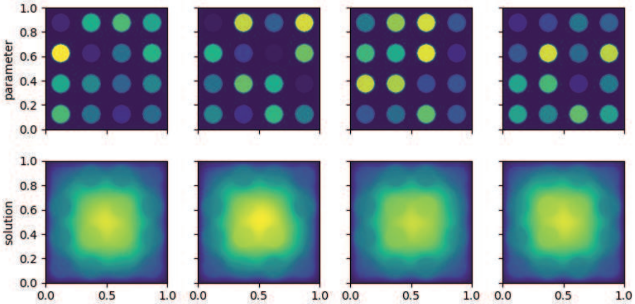

Figure 1. Realizations of parameter fields for the cookie problem and the corresponding solutions to the parametric Darcy problem

are build to control the discretization error

$$\mathcal{E} = \|u(y,\cdot) - u_h(y,\cdot)\|_{H^1_0(D)}$$

for any $y \in \Gamma$. The space is build in an adaptive manner by starting with a coarse FE space and repeating:

$$\begin{aligned}&\text{Solve on current space}\\ \to&\text{Estimate } \mathcal{E} \text{ locally}\\ \to&\text{Mark large error regions}\\ \to&\text{Refine marked regions}\end{aligned}$$

A CNN architecture is derived, which can approximate every step of the above iteration. There exists a constant $C > 0$ such that for any $\varepsilon > 0$ and V_h the final space of the described algorithm after $K \in \mathbb{N}$ steps with maximally $L \in \mathbb{N}$ refinement steps in every region, there exists a CNN $\Psi : \mathbb{R}^{2\times \dim V_L} \to \mathbb{R}^{\sum_{\ell=1}^{L} \dim V_\ell}$ such that the number of parameters is bounded by $CLK \log(\varepsilon^{-1})$ and

$$\|u(y,\cdot) - \mathcal{F}(\Psi(\kappa_L(y), f_L))\|_{H^1(D)} \leq \|u(y,\cdot) - u_h(y,\cdot)\|_{H^1(D)} + \varepsilon,$$

where $\mathcal{F}$ maps the coefficients of the CNN output to the corresponding FE function.

Approximation of corrections. As the derived CNN can approximate steps of an adaptive finite element method, individual parts of the network can be trained separately. A first part of the network can approximate the solution on a coarse grid, while the following parts of the network approximate correcitons of the solution on finer grids, as depicted in figure 2. The training of only few parameters at a time yields an advantage, when optimizing the network. Furthermore, the influence of later corrections quickly decreases, which gives a need for good approximations in the first steps and requires less accuracy in later corrections. This can be translated into smaller networks for later corrections or only few fine grid training samples.

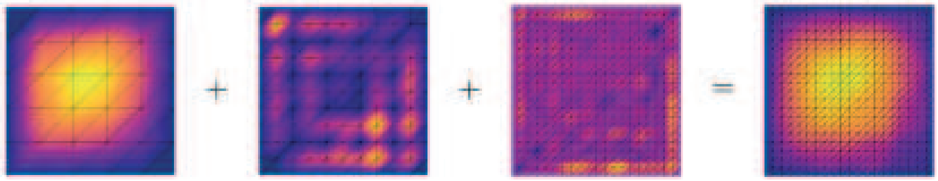

Figure 2. Visualization of the multilevel decomposition

Coclusions and outlook. Convolutional neural networks are an efficient tool to solve parametric partial differential equations. Theoretically small approximation errors can be achieved with network sizes growing only logarithmically with the the inverse of the required error bound. Numerically, the multilevel decomposition of the data allows for efficient training of small networks and with few expensive and many cheap data points. Solving a parametric PDE for a given parameter with the trained neural network only takes one forward pass through network, which can be evaluated quickly.

Applying this network to different applications, such as the inverse problem mapping the solution to the parameter, is of great interest.

References

[1] C. Heiß, I. Gühring, M. Eigel, *Multilevel CNNs for Parametric PDEs*, (2023), arXiv preprint arXiv:2304.00388, [cs.LG].

[2] M. Eigel, C. Jeffrey Gittelson, C. Schwab, E. Zander, *Adaptive stochastic Galerkin FEM*, Computer Methods in Applied Mechanics and Engineering **270**, pp. 247–269, (2014).

[3] M. Eigel,R. Schneider, P. Trunschke, S. Wolf, *Variational Monte Carlo–bridging concepts of machine learning and high-dimensional partial differential equations*, Advances in Computational Mathematics **45**, pp. 2503–2532 (2019).

Reporter: Cristina Cipriani

Participants

Prof. Dr. Boian Alexandrov
Los Alamos National Laboratory
Theoretical Division
Mail Stop B 265
Los Alamos, NM 87545
UNITED STATES

Prof. Dr. Markus Bachmayr
Institut für Geometrie und Praktische Mathematik
RWTH Aachen
Templergraben 55
52062 Aachen
GERMANY

Cristina Cipriani
Zentrum Mathematik
TU München
Boltzmannstr. 3
85748 Garching bei München
GERMANY

Dr. Nadav Cohen
School of Computer Science
Tel Aviv University
69978 Ramat Aviv, Tel Aviv
ISRAEL

Dr. Sergey Dolgov
Dept. of Mathematical Sciences
University of Bath
Claverton Down
Bath BA2 7AY
UNITED KINGDOM

Dr. Martin Eigel
Weierstraß-Institut für
Angewandte Analysis und Stochastik
Mohrenstraße 39
10117 Berlin
GERMANY

Prof. Dr. Lars Grüne
Mathematisches Institut
Lehrstuhl für Angewandte Mathematik
Universität Bayreuth
95440 Bayreuth
GERMANY

Bernhard Höveler
Fachbereich Mathematik
Technische Universität Berlin
Straße des 17. Juni 136
10623 Berlin
GERMANY

Prof. Dr. Sophie Langer
Department of Applied Mathematics
University of Twente
P.O. Box 217
7500 AE Enschede
NETHERLANDS

Thong Le
Institut für Geometrie und Praktische Mathematik
RWTH Aachen
Templergraben 55
52062 Aachen
GERMANY

Charles Miranda
Weierstraß-Institut für
Angewandte Analysis und Stochastik
Mohrenstraße 39
10117 Berlin
GERMANY

Prof. Dr. Anthony Nouy
Centrale Nantes, Nantes Université
1, rue de la Noe
P.O. Box 92101
44321 Nantes Cedex 3
FRANCE

Prof. Dr. Ivan Oseledets
Fachbereich Mathematik
T.U. Kaiserslautern
Postfach 3049
67618 Kaiserslautern
GERMANY

Mathias Oster
Institut für Mathematik
RWTH Aachen
Templergraben 55
52062 Aachen
GERMANY

Prof. Dr. Holger Rauhut
Department of Mathematics
LMU München
Theresienstr. 39
80333 München
GERMANY

Prof. Dr. Luca Saluzzi
Scuola Normale Superiore
Piazza dei Cavalieri, 3
Pisa 56126
ITALY

Janina Schütte
Weierstraß-Institut für
Angewandte Analysis und Stochastik
Mohrenstraße 39
10117 Berlin
GERMANY

Dr. Philipp Trunschke
Laboratoire de Mathématiques
Jean Leray UMR 6629
Université de Nantes, B.P. 92208
2 Rue de la Houssiniere
44322 Nantes Cedex 03
FRANCE