[5] Adrien Vacher and Boris Muzellec and Alessandro Rudi, and Francis Bach and François-Xavier Vialard. A Dimension-free Computational Upper-bound for Smooth Optimal Transport Estimation. *arXiv e-prints*, page arXiv:2101.05380, January 2021.
[6] Boris Muzellec and Adrien Vacher and Francis Bach and François-Xavier Vialard and Alessandro Rudi. A Dimension-free Computational Upper-bound for Smooth Optimal Transport Estimation. *arXiv e-prints*, page arXiv:2101.05380, December 2021.
[7] Holger Wendland and Christian Rieger. Approximate interpolation with applications to selecting smoothing parameters. *Numer. Math.*, 101(4):729–748, October 2005.

Towards Molecular Computational Anatomy?

Alain Trouvé
(joint work with Michael Miller, Daniel Tward, Laurent Younes)

Current descriptions of brain diseases usually need to put together several orders of magnitude ranging from the millimeter scale for tissues in standard imaging devices to the micron or even nano scale for neural cells and molecules.

Organizing these representations within a given patient or between a population for statistical modelling and understanding could be quite helpful but is still very challenging from a mathematical and computational perspective. In this talk, I will present our recent attempt to make a step in that direction in the context of computational anatomy based on two key assets: 1- Layered coarse-to-fine diffeomorphic transport based on idea coming from optimal control and riemannian geometry 2- Varifold based representations of information and reduction.

References

[1] M.I. Miller, D. Tward, A. Trouvé *Molecular Computational Anatomy: Unifying the Particle to Tissue Continuum via Measure Representations of the Brain*, BME Frontiers (To appear).
[2] M.I. Miller, A. Trouvé, L. Younes, *Image Varifolds on Meshes for Mapping Spatial Transcriptomics*, arXiv:2208.08376 (2022)

Analysis of 1-Lipschitz Neural Networks

Sebastian Neumayer
(joint work with Pakshal Bohra, Stanislas Ducotterd, Alexis Goujon, Dimitris Perdios, and Michael Unser)

The topics covered in this talk are related to the recent preprint [1]. Lipschitz constrained neural networks have several advantages compared to unconstrained ones and can be applied to various different problems. Consequently, they have recently attracted considerable attention in the deep learning community. Since designing and training expressive Lipschitz-constrained networks is very challenging, there is a need for improved methods and a better theoretical understanding. As the general case is very demanding, we restrict our attention to feed-forward neural networks with 1-Lipschitz component-wise activation functions and weight matrices with p-norm less or equal than one. This indeed leads to 1-Lipschitz

neural networks, for which naturally the question of expressiveness arises. Unfortunately, it turns out that networks with ReLU activation functions have provable disadvantages in this setting. Firstly, they cannot represent even simple piece-wise linear functions such as the hat function. Secondly, there exists a whole class of relatively simple functions that cannot be approximated in terms of the uniform norm on bounded boxes. To show this fact, we can make use of the second-order total variation and the fact that ReLU networks can only produce functions with bounded second-order total variation.

Due to these observations, we propose to use learnable spline activation functions with at least 3 linear regions instead. Clearly, this more complicated architecture should be motivated by theoretical findings. To this end, we prove that our architecture is optimal among all component-wise 1-Lipschitz activation functions in the sense that no other weight constrained architecture can approximate a larger class of functions. However, it remains an open question whether such NNs are universal approximators of 1-Lipschitz functions and our result can be seen as a first step towards its solution. Further, we prove that our proposed networks are in principal able to reproduce functions with arbitrary high second-order variation. Note that our architecture is also at least as expressive as the recently introduced non component-wise Groupsort activation function [2] for 2-norm-constrained weights. A more thorough comparison of linear splines to non component-wise activation functions is subtle, and it is so far unclear which choice leads to more expressive NNs in the remaining settings. Concerning the question of universality, the talk focused mainly on the approximation of scalar-valued functions $f\colon \mathbb{R}^d \to \mathbb{R}$. This also reflects the current state of research, where most results are only formulated for scalar-valued NNs. Extending these results to vector-valued functions appears highly non-trivial and should be addressed in future research. Finally, I would like to mention that little is known about the optimal structure for deep spline and Groupsort NNs, i.e., if it is more preferable to go deep or wide in architecture design.

On the numerical side, we are currently preparing a preprint with extensive experiments and details for an efficient implementation. For the implementation, we basically rely on a B-spline representation, which was already used before in [3]. However, we also need to take care of the additional Lipschitz constrained. This can be done in several ways, but naive approaches can lead to inferior training performance. To circumvent this issue, we instead propose to directly optimize over the set of 1-Lipschitz linear splines based on a method called *Differentiable Slope Clipping*. Our preliminary numerical results for one-dimensional function fitting, Wasserstein distance estimation and image reconstruction within the Plug-and-Play framework confirm that our architecture is at least competitive (often even better) with other recently proposed activation functions such as GroupSort, Householder activations and parametric ReLU, which were also all designed with the goal of increasing expressivity in mind. One additional advantage of our implementation over the other methods is that it can be applied to any already trained network by just initializing the linear splines accordingly. This possibly avoids

an expensive retraining. A Github repository with the implementation will be available soon.

References

[1] S. Neumayer, A. Goujon, P. Bohra, and M. Unser, *Approximation of Lipschitz Functions using Deep Spline Neural Networks*, arXiv:2204.06233 (2022).

[2] Q. Li, S. Haque, C. Anil, J. Lucas, R. Grosse, and J. Jacobsen, *Preventing Gradient Attenuation in Lipschitz Constrained Convolutional Networks*, Advances in Neural Information Processing Systems **32**, (2019), 15364–15376.

[3] P. Bohra, J. Campos, H. Gupta, S. Aziznejad, and M. Unser, *Learning activation functions in deep (spline) neural networks*, IEEE Open Journal of Signal Processing **1** (2020), 295–309.

Discrete geodesic calculus in the manifold of Sobolev curves

Benedikt Wirth

(joint work with Martin Rumpf)

The manifold of (closed) Sobolev curves is a well-known example of an infinite-dimensional shape space. It consists of immersions of the circle $\mathbb{S}^1$ into $\mathbb{R}^d$ with Sobolev regularity,

$$\mathrm{Imm}^m = \{c \in W^{m,2}(\mathbb{S}^1;\mathbb{R}^d) \,|\, c'(\theta) \neq 0 \text{ for all } \theta \in \mathbb{S}^1\},$$

where $m \geq 2$ and c' denotes the derivative with respect to the parametrization variable θ (which for $m \geq 2$ is everywhere defined). This manifold can be equipped with a Riemannian metric of Sobolev type,

$$g_c(\xi,\zeta) = \int_{\mathbb{S}^1} \sum_{i=1}^{m} \partial_s^i \xi \cdot \partial_s^i \zeta \,\mathrm{d}s$$

for any curve $c \in \mathrm{Imm}^m$ and tangent vectors $\xi, \zeta : \mathbb{S}^1 \to \mathbb{R}^d$. Above, $s = \int_0^\theta |c'(\tilde\theta)| \,\mathrm{d}\tilde\theta$ denotes arclength along the curve c so that

$$\mathrm{d}s = |c'(\theta)|\mathrm{d}\theta, \qquad \partial_s = \tfrac{\partial_\theta}{|c'(\theta)|}.$$

The induced Riemannian distance between two curves $c_0, c_1 \in \mathrm{Imm}^m$ can then be computed by minimizing the path energy $\mathcal{E}$ among all paths $(c_t)_{t\in[0,1]}$ in Imm^m with fixed endpoints c_0, c_1,

$$d^2(c_0,c_1) = \inf \mathcal{E}[(c_t)_{t\in[0,1]}] \qquad \text{with } \mathcal{E}[(c_t)_{t\in[0,1]}] = \int_0^1 g_{c_t}(\dot c_t, \dot c_t)\,\mathrm{d}t$$

(where $\dot c_t$ denotes the derivative of the path with respect to the time variable t). Furthermore, geodesics in the manifold of Sobolev curves can be defined as minimizers of this path energy for fixed endpoints.

Bruveris, Michor and Mumford have shown [1, 2] that this manifold of Sobolev curves is metrically and geodesically complete and that shortest geodesics between any two curves exist (as long as they lie in the same connected component, thus, if they have the same winding number when $d = 2$). To show this one exploits that the path energy $\mathcal{E}$ actually just behaves like the squared Sobolev norm of