Deep Learning ⇌ Partial Differential Equations

SIAM/CAIMS Annual Meeting, 2020

slides:

mathcs.emory.edu/~lruthot/talks/2020-SIAMCAIMS.pdf

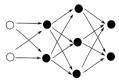
Lars Ruthotto

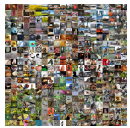
Departments of Mathematics and Computer Science, Emory University

lruthotto@emory.edu

@lruthotto

Core of Science: Understanding the World Through Models and Data





"komondor"

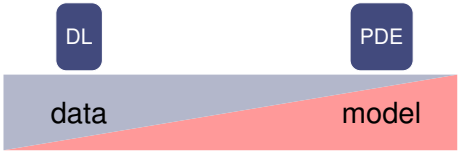
(Deng et al. 2009; Krizhevsky et al. 2012; He et al. 2015)

Deep Learning (DL)

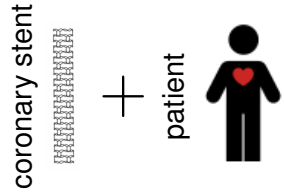
1. model: deep neural network
2. data: ImageNet $\geq$ 14M images
3. key: generalize beyond data

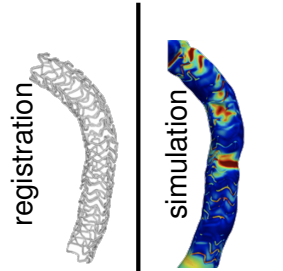
Partial Differential Equations (PDE)

1. models: elasticity, Navier stokes, ...
2. data: 1-3 images of patient
3. key: analyze, discretize, solve PDE



Today: Connecting DL and PDEs

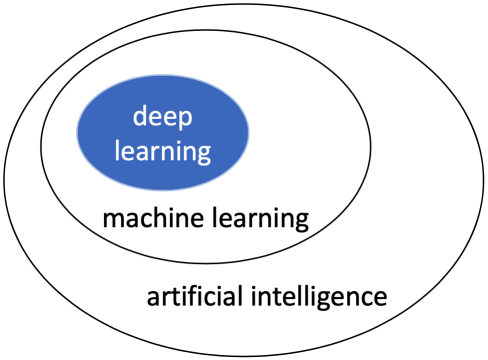


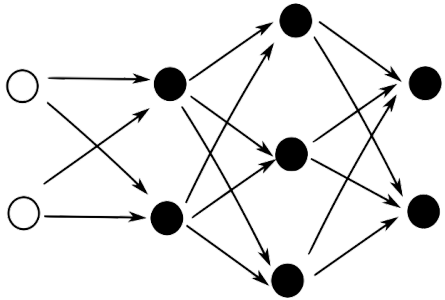


success rate >90%

(Lin et al. 2017; Lefieux et al. 2020; Viguerie and Veneziani 2019)

Deep Learning in a Nutshell





from (Higham and Higham 2018)

- DL := machine learning (ML) with deep neural networks (DNN)
- DNN := neural network with many layers
- AI research activity follows waves, starting ≈ 1950s
- new surge due to massive datasets and computing power
- excellent references: Goodfellow et al. 2016; Higham and Higham 2018

Computational and Applied Mathematicians' Role in DL

An (almost perfectly) true statement

$$\text{lots of data} + \text{back propagation} + \text{GPU} + \left\{ \begin{array}{c} \text{TensorFlow} \\ \text{Caffe} \\ \text{Torch} \\ \vdots \end{array} \right. \Rightarrow \text{success!}$$

So, why study the mathematics of deep learning?

Fundamental Questions and Recent Mathematical Advances

Expressibility, Approximation Properties

- why does the DNN succeed (or fail) to approximate a function/operator?
- recent works: Poggio et al. 2017; Bölcskei et al. 2019; Lu et al. 2019a

Learning, Optimization, Generalization

- why do some optimization algorithms lead to better generalization than others?
- recent works: Bottou et al. 2018; Zhang et al. 2018; Chizat and Bach 2020; Osher et al. 2018
- Tue, 5PM MT1: Generalization Theory in Machine Learning (A. Oberman)

Explainability, Interpretability

- why does the neural network choose a certain prediction?
- recent works: Samek et al. 2017; Montavon et al. 2018; Adebayo et al. 2018

Robustness, Adversarial Attacks, Stability

- why can DNNs be easily fooled by perturbing input features or training data?
- recent works: Madry et al. 2017; Shafahi et al. 2018; Wang et al. 2018; Etmann et al. 2019

Fairness

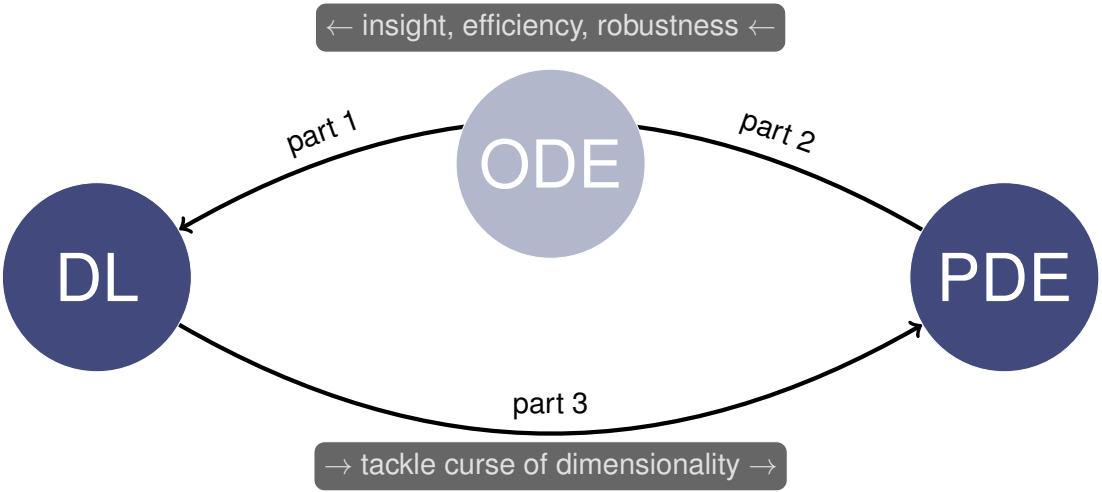
- does my DNN discriminate based on sensitive features (e.g., gender, ethnicity)?
- recent works: Friedler et al. 2016; Kleinberg et al. 2016; Bellamy et al. 2018

Scientific Use of Machine Learning

- recent works: Lusch et al. 2018; Raissi et al. 2019; Han et al. 2018; Arridge et al. 2019
- July 20 – July 24: Mathematical and Scientific Machine Learning (MSML 2020)

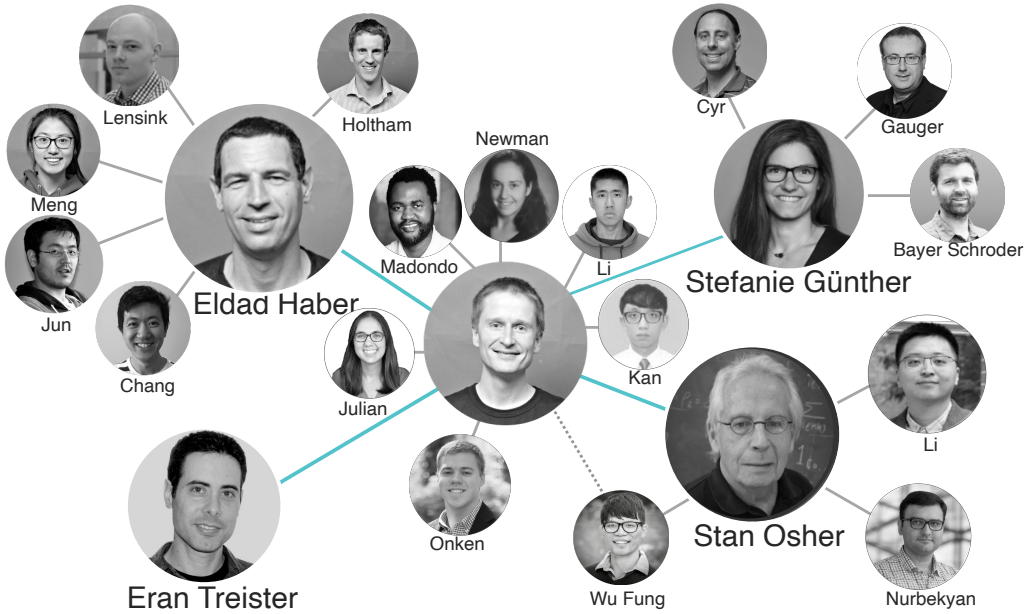
Roadmap: Deep Learning ⇌ Partial Differential Equations

Main goal: highlight connections between PDEs, ODEs, and DL.

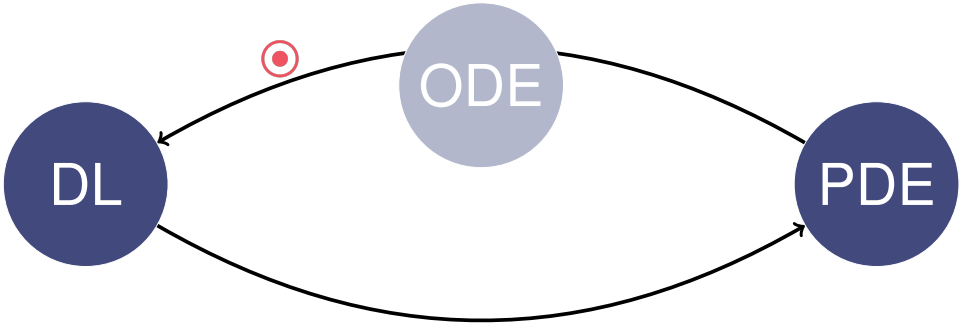


DL: Deep Learning PDE: Partial Differential Equations ODE: Ordinary Differential Equations

Collaborators and Funding

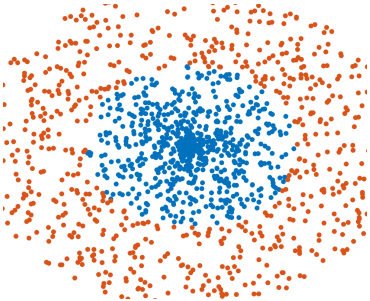


Funding: DMS 1751636, 1522599 BSF 2018209 MLP 2019

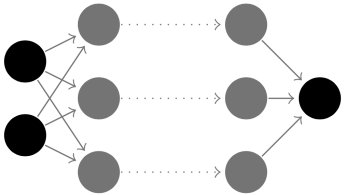


Up next: ODE → DL

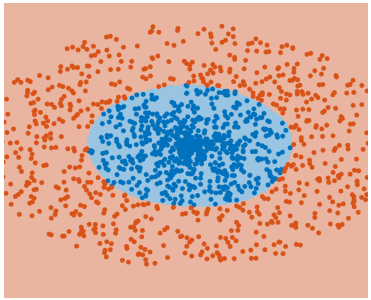
Example: Supervised Classification with a DNN



training data



DNN sketch



classification result

ResNet: Residual Neural Networks (He et al. 2016)

Training data $\{(\mathbf{y}^{(1)}, c^{(1)}), (\mathbf{y}^{(2)}, c^{(2)}), \dots\} \subset \mathbb{R}^2 \times \{0, 1\}$.
Forward propagation of input $\mathbf{y}$ through simple ResNet

$$
\begin{aligned}
\mathbf{u}_0 &= \mathbf{K}_{\text{in}}\mathbf{y} \\
\mathbf{u}_1 &= \mathbf{u}_0 + h\,\sigma(\mathbf{K}_0\mathbf{u}_0 + \mathbf{b}_0) \\
\vdots &= \quad \vdots \\
\mathbf{u}_N &= \mathbf{u}_{N-1} + h\,\sigma(\mathbf{K}_{N-1}\mathbf{u}_{N-1} + \mathbf{b}_{N-1})) \\
z &= s(\mathbf{K}_{\text{out}}\mathbf{u}_N + \mathbf{b}_{\text{out}}),
\end{aligned}
$$

with $h > 0$, $\quad \boldsymbol{\theta}_i^{\text{Res}} := (\mathbf{K}_i, \mathbf{b}_i)$.

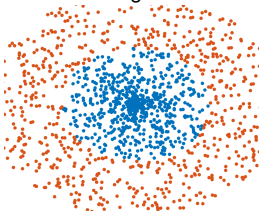
Let $F(\mathbf{y}, \boldsymbol{\theta}) := z$, weights $\boldsymbol{\theta} := (\boldsymbol{\theta}_0^{\text{Res}}, \dots, \boldsymbol{\theta}_{N-1}^{\text{Res}}, \mathbf{K}_{\text{out}}, \mathbf{b}_{\text{out}})$.

Train weights by solving (Bottou et al. 2018)

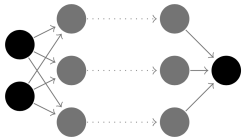
$$
\min_{\boldsymbol{\theta}} \mathbb{E}\left[\ell(F(\mathbf{y}, \boldsymbol{\theta}), c)\right] + \frac{\alpha}{2}\|\boldsymbol{\theta}\|_2^2,
$$

with cross entropy loss $\ell(z, c) = -c\log(z) - (1 - c)\log(1 - z)$.

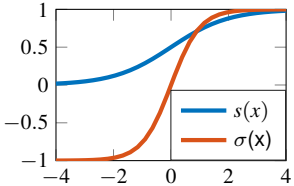
training data:



DNN sketch:



activation σ and sigmoid s:



ResNet: Discussion

In ResNet, $\mathbf{u}_N$ is forward Euler approximation of $\mathbf{u}(T)$,

$$\partial_t \mathbf{u}(t) = f(\mathbf{u}(t), \boldsymbol{\theta}^{\text{ODE}}(t)), \quad t \in (0, T], \quad \mathbf{u}(0) = \mathbf{u}_0;$$

see (E 2017; Haber and Ruthotto 2017) ($\approx$ same time).

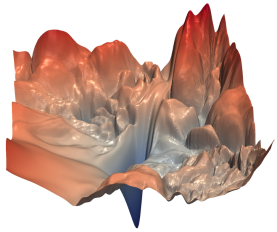
Advantages over other Architectures

1. ResNets often improve with depth
2. state-of-the-art results for many tasks
3. easy to train and easy to add depth

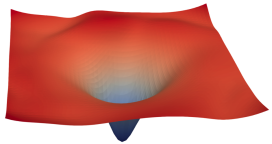
Remarks

1. $f(\mathbf{u}, \boldsymbol{\theta}^{\text{ODE}}(t)) = \sigma(\mathbf{K}(t)\mathbf{u} + \mathbf{b}(t))$ gives ResNet on previous slide
2. in practice: more complicated layer f, concatenate ResNets to change width or image resolution
3. similar continuous networks, extensions to PDEs, and implicit time integrators in (Rico-Martínez et al. 1992; González-García et al. 1998).

impact on loss function (Li et al. 2018)



56-layer network (no ResNet)



56-layer ResNet

Stable Architectures for DNNs (Haber and Ruthotto 2017)

When is forward propagation stable? That is, when $\exists M > 0$ such that

$$\|F(\mathbf{y} + \boldsymbol{\epsilon}, \boldsymbol{\theta}) - F(\mathbf{y}, \boldsymbol{\theta})\| \leq M\|\boldsymbol{\epsilon}\| \quad (\boldsymbol{\epsilon} \text{ input perturbation})$$

Motivation: well-posed training problem, adversarial attacks, efficient optimization,...

Main Findings and Contributions

1. $\partial_t \mathbf{u}(t) = \sigma(\mathbf{K}(t)\mathbf{u}(t) + \mathbf{b}(t))$ not stable for all $\mathbf{K}(\cdot), \mathbf{b}(\cdot)$
2. alternative f: antisymmetric $\mathbf{K}$, Hamiltonian-inspired networks
3. symplectic integrators to obtain stable architecture ($\neq$ ResNet)
4. stable DNNs perform competitively (on simple tasks)

Improvements and Related Works

1. more expressive architectures (Chang et al. 2018), multilevel training (Chang et al. 2017)
2. improved stability results (Ruthotto and Haber 2020; Celledoni et al. 2020)
3. analysis: convergence (Thorpe and Gennip 2018), opt. conditions (Benning et al. 2019)
4. multi-step and other time integrators (Lu et al. 2017)
5. discrete weights (Li and Hao 2018), maximum principles (Li et al. 2017)

Neural ODEs: Neural Ordinary Differential Equations (Chen et al. 2018)

Main Novelties and Contributions

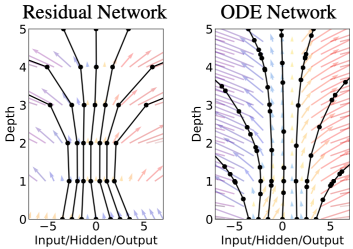
1. apply adaptive time integrator to continuous ResNet
2. compute gradients of loss function using adjoint equation

$$-\partial_t \mathbf{p}(t) = \nabla f(\mathbf{u}(t), \boldsymbol{\theta}(t))^\top \mathbf{p}(t), \quad \mathbf{p}(T) = \nabla_{\mathbf{u}_N} \ell(F(\mathbf{y}), c)$$

3. save memory by re-computing $\mathbf{u}(t)$ backward in time with $\mathbf{p}$.
4. popularized continuous models in ML community

Improvements and Related Works

1. example for numerical instability of item 3 above (Gholami et al. 2019) and alternative using checkpointing
2. invertible ResNet (Behrmann et al. 2019), generative modeling (Grathwohl et al. 2018; Chen et al. 2019)
3. augmented, more expressive models (Dupont et al. 2019)



from (Chen et al. 2018)

Artificial intelligence / Machine learning

A radical new neural network design could overcome big challenges in AI

Researchers borrowed equations from calculus to redesign the core machinery of deep learning so it can model continuous processes like changes in health.

by **Karen Hao** December 12, 2018

MIT Tech Review, 2018

Optimize-Discretize vs. Discretize-Optimize (Gholami et al. 2019)

Compare optimal control approaches for learning problems like

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\ell(F(\mathbf{y},\boldsymbol{\theta}),c)\right] + \frac{\alpha}{2}\|\boldsymbol{\theta}\|_2^2,$$

where $\mathbf{u}_N$ in F is approximately equal to $\mathbf{u}(T)$ given by

$$\partial_t \mathbf{u}(t) = f(\mathbf{u}(t), \boldsymbol{\theta}^{\mathrm{ODE}}(t)), \quad t \in (0,T], \quad \mathbf{u}(0) = \mathbf{u}_0.$$

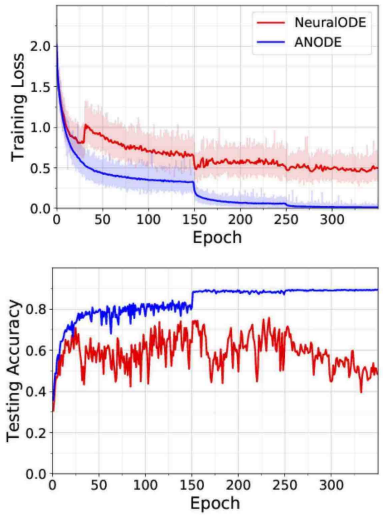
$O \to D$: Optimize-Discretize (Neural ODE)

1. keep $\boldsymbol{\theta}^{\mathrm{ODE}}, \mathbf{u}$ continuous in time
2. Euler-Lagrange-Equations $\rightsquigarrow$ adjoint equation
3. use adaptive time integrators in optimization

$D \to O$: Discretize-Optimize (ANODE)

1. discretize $\boldsymbol{\theta}^{\mathrm{ODE}}, \mathbf{u}$ in time (could use different grids)
2. differentiate discrete problem $\rightsquigarrow$ backpropagation
3. keep discretization fixed during optimization

Example (image classification)



more examples in (Gholami et al. 2019; Onken and Ruthotto 2020)

My advice: use $D \to O$ (accurate gradients, fixed costs, convergence, generalization)

Layer-Parallel Training of Deep ResNets (Günther et al. 2020)

Idea: Train ResNet with parallel-in-time methods from time-dependent control (Falgout et al. 2014)

1. replace (sequential) forward and backward propagation with (parallel) nonlinear multigrid iteration
2. simultaneously iterate on $\boldsymbol{\theta}$ and $\mathbf{u}, \mathbf{p}$.

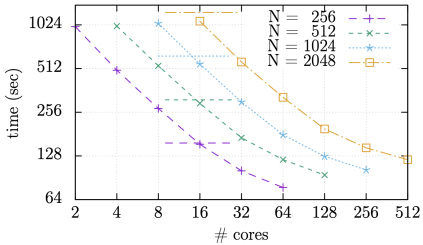
Findings and Contributions (Indian Pines Example):

1. parallel multigrid faster when using $\geq$ 16 cores
2. simultaneous optimization speeds up training by 4.4$\times$ with 128 cores ($\approx$3.5% efficiency)
3. layer-parallelism is new way to parallelize and distribute ResNet training (in addition to data parallelism)

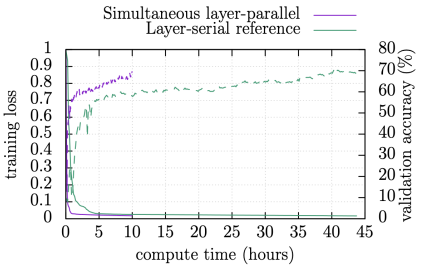
Improvements and Related Works:

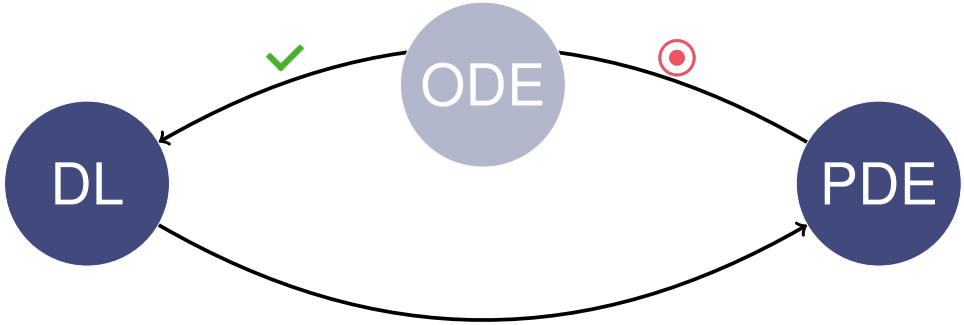
1. nested iterations increase performance (Cyr et al. 2019)
2. more efficient parallel-in-time (Parpas and Muir 2019)

strong scaling (fwd + gradient)



simultaneous optimization

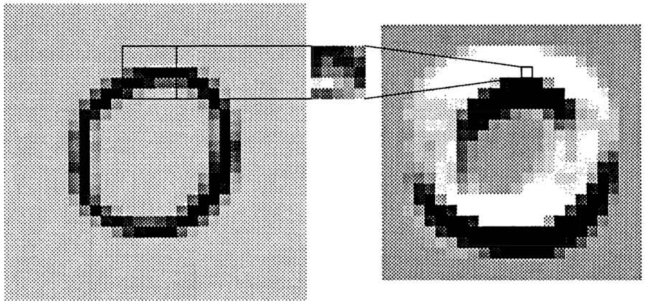




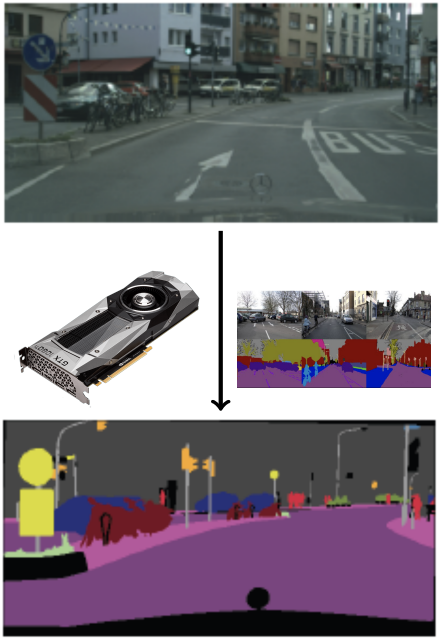
Up next: PDE → DL

Convolutional Neural Networks (CNNs) for Speech, Image, Video Data

Example: digit recognition (LeCun et al. 1990)



Example: segmentation (Brostow et al. 2009)



Key Challenges (Segmentation Example):

1. input: images consist of $\approx$ 700K pixels
2. output: label each pixel into one of 32 classes
3. convolutions act locally $\rightsquigarrow$ need many layers
4. need efficient and robust prediction

$\Rightarrow$ **need computing power, storage, new ideas**

Akiyama et al. 2017

Lessons from PDE-Based Image Processing

A few seminal works

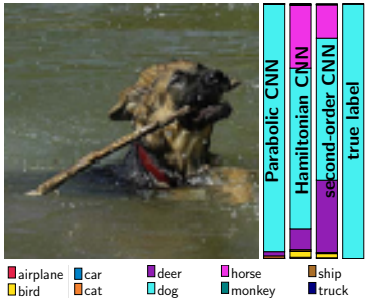
- optical flow (Horn and Schunck 1981)
- elasticity for image registration (Broit 1981)
- variational methods for image segmentation (Mumford and Shah 1989)
- **total variation for edge-preserving denoising (Rudin et al. 1992)**
- nonlinear diffusion (Perona et al. 1994; Scherzer and Weickert 2000; Weickert 2009)

Common thread: Replace discrete images and operators with functions and PDE $\rightsquigarrow$ better understanding, improved robustness, higher efficiency.

see: SIAM Bookstore, SIAM IS20 (e.g., IP2 by Thomas Pock, MS 53, MS 70)

Deep Neural Networks Motivated by PDEs (Ruthotto and Haber 2020)

Idea: design CNNs that inherit properties of PDEs.

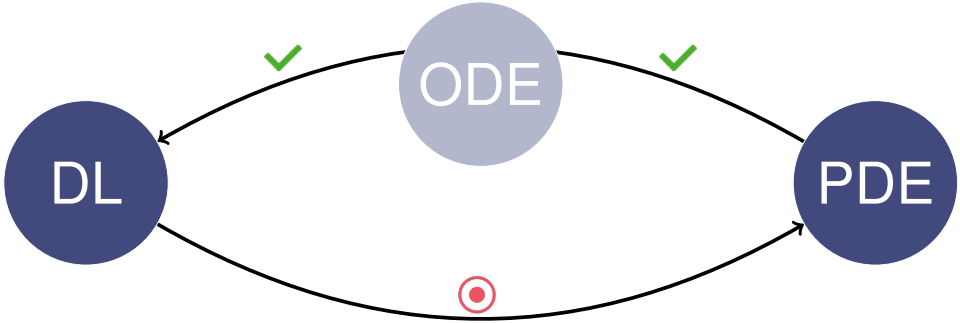


Main Findings and Contributions

1. stability result for non-autonomous **parabolic CNNs**
2. first-order and second-order **hyperbolic CNNs**
3. hyperbolic: symplectic integrators $\rightsquigarrow \downarrow$ memory
4. different PDEs lead to competitive performance

Improvements and Related Works

1. reversibility first used in (Chang et al. 2018) to train 1202-layer ResNet on one GPU
2. overcome field-of-view problem with semi-implicit time-stepping (Haber et al. 2019)
3. reduce massive number of CNN weights ($\mathcal{O}(10^6)$) using lean operators (Ephrath et al. 2020a) and multigrid-in-channel (Ephrath et al. 2020b)
4. (Lensink et al. 2019) added wavelets to alleviate all memory requirements
5. related: parameter estimation (González-García et al. 1998), reaction diffusion for denoising (Chen and Pock 2017)



Up next: DL → PDE

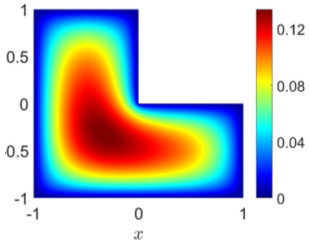
Example: Deep Learning for High-Dimensional PDEs

Consider this PDE problem

$$-\Delta\Phi(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \quad \Phi(\mathbf{s}) = 0, \quad \mathbf{s} \in \partial\Omega.$$

Idea: Parameterize $\Phi(\cdot)$ with a neural network $F(\cdot, \boldsymbol{\theta})$ and solve

$$\min_{\boldsymbol{\theta}} \int_\Omega \frac{1}{2}\|\nabla F(\mathbf{x}, \boldsymbol{\theta})\|^2 - F(\mathbf{x}, \boldsymbol{\theta})g(\mathbf{x})d\mathbf{x} + \frac{\lambda}{2}\int_{\partial\Omega} F(\mathbf{s}, \boldsymbol{\theta})^2 d\mathbf{s}.$$



from (Lu et al. 2019b)

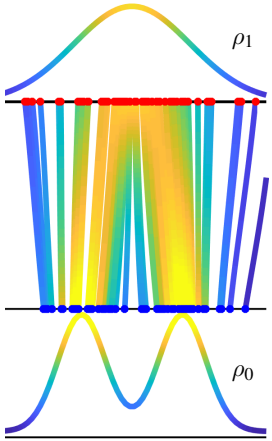
Short discussion

- DNNs are mesh-free and scale to high-dimensions
- use PDE (no training data needed)
- linear PDE ⇝ non-convex optimization problem

A few (of many) recent works in this area

1. Deep Galerkin method (Sirignano et al. 2018), Deep Ritz method (Weinan and Yu 2018)
2. Nonlinear Black-Scholes, Hamilton-Jacobi Bellman, Allen-Cahn (Han et al. 2018)
3. Physics-informed NN: forward/inverse problems, fractional, stochastic (Raissi et al. 2019)
4. theoretical results (Kutyniok et al. 2019; Shin et al. 2020)

ML for High-Dimensional Mean Field Games (Ruthotto et al. 2020)



Idea: Use DNNs for optimal transport and mean field games.

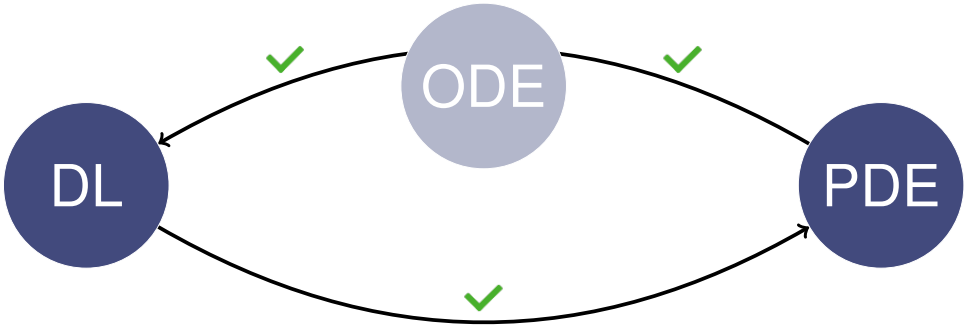
1. variational approach, Hamilton-Jacobi-Bellman (HJB) penalty
2. simulate population density with Lagrangian PDE solver
3. tailored neural network, fast gradient and Laplacian computations

Main Findings and Contributions

1. $d = 2$: tie with convex solver (Haber and Horesh 2015)
2. synthetic transport and crowd motion problems up to $d = 100$
3. HJB penalty: more efficiency and accuracy

Improvements and Related Work

1. ML applications: (Yang and Karniadakis 2019; Finlay et al. 2020; Onken et al. 2020)
2. stochastic mean field games (Lin et al. 2020)
3. CP 1: Derek Onken's poster
4. Wed 3:30 PM: Levon Nurbekyan's talk in MS 18



Up next: Summary

Deep Learning and PDEs: Related Activities

SIAM/CAIMS Annual Meeting

- **CP1 / MS34: Contributed and Poster**
- MS 8: Reinforcement Learning and Behavioral Modeling
- **MS 9: Tutorial on Emerging Research Areas**
- MT1: Generalization Theory (Adam Oberman)
- JP1: Optimal Transport for Machine Learning (Gabriel Peyré)
- MS 16 (3 parts): Developments in Machine Learning
- MS 18 (3 parts): Intersection of Optimal Control and ML
- MS 68: Sparse Recovery and ML
- IP 13: Solving Eigenvalue Problems in High Dimension (Jianfeng Lu)

SIAM Imaging Sciences 2020

- IP 2: Variational Networks (Thomas Pock)
- IP 5: Deep Internal Learning (Michael Irani)
- IP 6: Deep Learning in Wave-based Imaging and Inverse Problems (Maarten V. de Hoop)
- **MT 1: Model-Versus Learning-Based Approaches to Image Reconstruction** (Moeller and Cremers)
- MS 8 (Learning and Processing of Geometric Structures), MS 12 (Semi-Supervised Learning), MS 15 (Learning Priors), MS 32 (Learning Imaging Operators), MS 47

Lectures and Seminars Available On-Demand

SIAM MDS 2020

- Eldad Haber, *Deep Neural Nets meet ODE/PDEs* (65 mins)
- Christoph Reisinger, *DL for Optimal Control and PDEs* (56 mins)
- Weinan E, *A Mathematical Perspective of ML* (60 mins)
- **MS119: *Advances in Optimal Control for and with ML* (3 talks)**
- **MS130: *Advances in Optimal Control for and with ML* (4 talks)**

IPAM

- Levon Nurbekyan, *Computational Methods for Mean Field Games* (2 parts)
- Lars Ruthotto, *Deep Neural Networks as ODE/PDE* (2 parts)
- Lars Ruthotto, *Numerical Analysis Perspective on DNNs* (56 mins)

Other Talks

- Samy Wu Fung, *APAC Net - DL for Stochastic Mean Field Games* (45 mins)
- Lars Ruthotto, *ML* $\leftrightarrow$ *Optimal Transport* (48 mins)

Σ: Deep Learning ⇌ Partial Differential Equations

Let us correct our statement from above

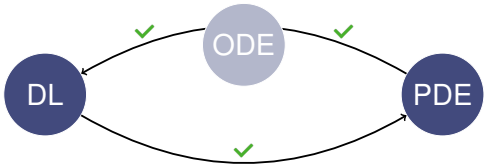
$$\text{data} + \text{back propagation} + \text{GPU} + \left\{ \begin{array}{c} \text{TensorFlow} \\ \text{Caffe} \\ \text{Torch} \\ \vdots \end{array} \right. + \text{mathematics} \Rightarrow \text{success}$$

What we covered:

- **PDE → DL**: insight, efficiency, robustness
- **DL → PDE**: tackle curse of dimensionality

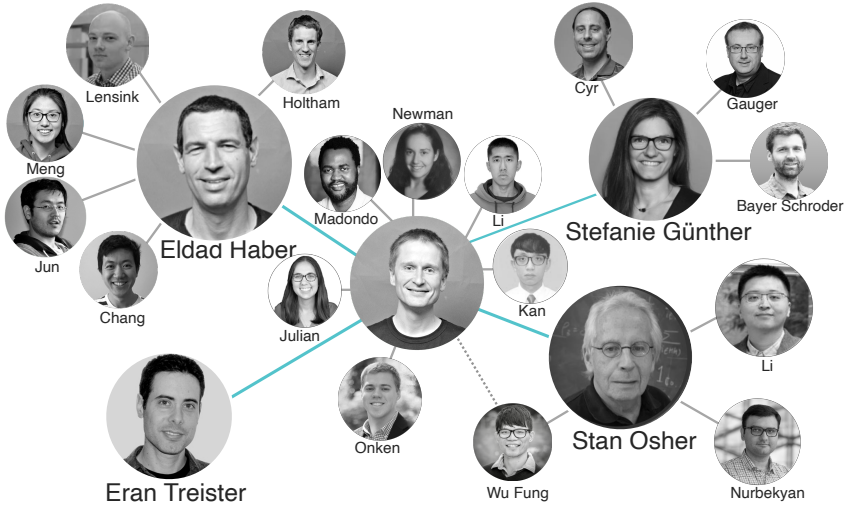
What we did not cover:

- unsupervised, semi-supervised, reinforcement, active learning
- **DL + PDE**: combine data and models



Short Q&A now! More at `emory.zoom.us/my/lruthotto`

Acknowledgements



Mentors, Colleagues, Collaborators:

Michele Benzi
Martin Burger
Chen Greif
Joseph Hart
James Herring
James Nagy
Jan Modersitzki
Alessandro Veneziani
Bart Van Bloemen Waanders
Yuanzhe Xi

Thanks to SIAM / CAIMS!

Funding: DMS 1751636, 1522599 BSF 2018209 MLP 2019

References

Adebayo, Julius et al. (2018). “Sanity checks for saliency maps”. In: *Advances in Neural Information Processing Systems*, pp. 9505–9515.

Akiyama, Kazunori et al. (Feb. 2017). “Imaging the Schwarzschild-radius-scale Structure of M87 with the Event Horizon Telescope using Sparse Modeling”. In: *arXiv.org* 1, p. 1. arXiv: .

Arridge, Simon et al. (2019). “Solving inverse problems using data-driven models”. In: *Acta Numerica* 28, pp. 1–174.

Behrmann, Jens et al. (2019). “Invertible residual networks”. In: *International Conference on Machine Learning*, pp. 573–582.

Bellamy, Rachel KE et al. (2018). “AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias”. In: *arXiv preprint arXiv:1810.01943*.

Benning, Martin et al. (2019). “Deep learning as optimal control problems: models and numerical methods”. In: *arXiv preprint arXiv:1904.05657*.

Bölcskei, Helmut et al. (2019). “Optimal approximation with sparsely connected deep neural networks”. In: *SIAM Journal on Mathematics of Data Science* 1.1, pp. 8–45.

Bottou, L et al. (2018). “Optimization methods for large-scale machine learning”. In: *SIAM Journal on Mathematics of Data Science* 60.2, pp. 223–311.

Broit, Chaim (1981). “Optimal registration of deformed images”. In:

Brostow, Gabriel J. et al. (2009). “Semantic object classes in video: A high-definition ground truth database”. In: *Pattern Recognit. Lett.* 30, pp. 88–97.

Celledoni, Elena et al. (2020). *Structure preserving deep learning*. arXiv: .

Chang, Bo et al. (Oct. 2017). “Multi-level Residual Networks from Dynamical Systems View”. In: *arXiv.org*. arXiv: .

References (cont.)

Chang, Bo et al. (2018). “Reversible architectures for arbitrarily deep residual neural networks”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 1–8.

Chen, Ricky TQ et al. (2019). “Residual flows for invertible generative modeling”. In: *Advances in Neural Information Processing Systems*, pp. 9916–9926.

Chen, Tian Qi et al. (June 2018). “Neural Ordinary Differential Equations”. In: *NeurIPS*.

Chen, Yunjin and Thomas Pock (June 2017). “Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration.”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, pp. 1256–1272.

Chizat, Lenaic and Francis Bach (2020). “Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss”. In: *arXiv preprint arXiv:2002.04486*.

Cyr, Eric C. et al. (2019). *Multilevel Initialization for Layer-Parallel Deep Neural Network Training*. arXiv: .

Deng, J. et al. (2009). “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*.

Dupont, Emilien et al. (2019). “Augmented Neural ODEs”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 3140–3150. URL: .

E, Weinan (Mar. 2017). “A Proposal on Machine Learning via Dynamical Systems”. In: *Communications in Mathematics and Statistics* 5.1, pp. 1–11.

Ephrath, Jonathan et al. (Feb. 2020a). “LeanConvNets: Low-cost Yet Effective Convolutional Neural Networks”. In: *Selected Topics in Signal Processing, IEEE Journal of*, pp. 1–1.

Ephrath, Jonathan et al. (2020b). *Multigrid-in-Channels Architectures for Wide Convolutional Neural Networks*. arXiv: .

References (cont.)

Etmann, Christian et al. (2019). “On the connection between adversarial robustness and saliency map interpretability”. In: *arXiv preprint arXiv:1905.04172*.

Falgout, R D et al. (Jan. 2014). “Parallel Time Integration with Multigrid”. In: *SIAM Journal on Scientific Computing* 36.6, pp. C635–C661.

Finlay, Chris et al. (2020). *How to train your neural ODE: the world of Jacobian and kinetic regularization*. arXiv: .

Friedler, Sorelle A. et al. (2016). *On the (im)possibility of fairness*. arXiv: .

Gholami, Amir et al. (Feb. 2019). “ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs”. In: *arXiv.org*. arXiv: .

González-García, R et al. (Mar. 1998). “Identification of distributed parameter systems: A neural net based approach”. In: *Computers Chem Engn.* 22, S965–S968.

Goodfellow, Ian et al. (Nov. 2016). *Deep Learning*. MIT Press.

Grathwohl, Will et al. (2018). “Ffjord: Free-form continuous dynamics for scalable reversible generative models”. In: *arXiv preprint arXiv:1810.01367*.

Günther, Stefanie et al. (Feb. 2020). “Layer-Parallel Training of Deep Residual Neural Networks”. In: *SIAM Journal on Mathematics of Data Science* 2.1, pp. 1–23.

Haber, Eldad and Raya Horesh (Mar. 2015). “A Multilevel Method for the Solution of Time Dependent Optimal Transport”. In: *Numerical Mathematics: Theory, Methods and Applications* 8.01, pp. 97–111.

Haber, Eldad and Lars Ruthotto (2017). “Stable architectures for deep neural networks”. In: *Inverse Problems* 34.1, pp. 1–22.

References (cont.)

Haber, Eldad et al. (Mar. 2019). “IMEXnet: A Forward Stable Deep Neural Network”. In: *36th International Conference on Machine Learning*, pp. 1–10.

Han, Jiequn et al. (2018). “Solving high-dimensional partial differential equations using deep learning”. In: *Proceedings of the National Academy of Sciences* 115.34, pp. 8505–8510.

He, K et al. (2015). “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE ICCV*.

He, Kaiming et al. (2016). “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Higham, Catherine F and Desmond J Higham (Jan. 2018). “Deep Learning: An Introduction for Applied Mathematicians”. In: *arXiv.org*. arXiv: .

Horn, Berthold K.P. and Brian G Schunck (1981). “Determining Optical-Flow”. In: *Artificial Intelligence* 17.1-3, pp. 185–203.

Kleinberg, Jon et al. (2016). *Inherent Trade-Offs in the Fair Determination of Risk Scores*. arXiv: .

Krizhevsky, Alex et al. (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105.

Kutyniok, Gitta et al. (2019). “A theoretical analysis of deep neural networks and parametric PDEs”. In: *arXiv preprint arXiv:1904.00377*.

LeCun, Y et al. (1990). “Handwritten digit recognition with a back-propagation network”. In: *Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop*, pp. 396–404.

Lefieux, Adrien et al. (2020). “Semi-automatic Reconstruction of Stented Coronaries: Data Assimilation and Computational Fluid Dynamics”. In: *submitted to IEEE Trans Medical Imaging*.

References (cont.)


Lensink, Keegan et al. (2019). *Fully Hyperbolic Convolutional Neural Networks*. arXiv: .


Li, H et al. (2018). "Visualizing the loss landscape of neural nets". In: *Advances in Neural Information Processing Systems*.


Li, Qianxiao and Shuji Hao (2018). "An optimal control approach to deep learning and applications to discrete-weight neural networks". In: *arXiv preprint arXiv:1803.01299*.


Li, Qianxiao et al. (2017). "Maximum principle based algorithms for deep learning". In: *The Journal of Machine Learning Research* 18.1, pp. 5998–6026.


Lin, Alex Tong et al. (2020). *APAC-Net: Alternating the Population and Agent Control via Two Neural Networks to Solve High-Dimensional Stochastic Mean Field Games*. arXiv: .


Lin, Claire Y et al. (2017). "Numerical Methods for Polyline-to-Point-Cloud Registration with Applications to Patient-Specific Stent Reconstruction". In: *Int J Numer Method Biomed Eng* 34.3, pp. 1–22.


Lu, Lu et al. (2019a). *DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators*. arXiv: .


Lu, Lu et al. (July 2019b). "DeepXDE: A deep learning library for solving differential equations". In: arXiv: .


Lu, Yiping et al. (Oct. 2017). "Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations". In: *arXiv.org*. arXiv: .


Lusch, Bethany et al. (2018). "Deep learning for universal linear embeddings of nonlinear dynamics". In: *Nature communications* 9.1, pp. 1–10.


Madry, Aleksander et al. (2017). *Towards Deep Learning Models Resistant to Adversarial Attacks*. arXiv: .

References (cont.)

Montavon, Grégoire et al. (2018). “Methods for interpreting and understanding deep neural networks”. In: *Digital Signal Processing* 73, pp. 1–15.

Mumford, D and J Shah (July 1989). “Optimal Approximations by Piecewise Smooth Functions and Associated Variational-Problems”. In: *Communications on Pure and Applied Mathematics* 42.5, pp. 577–685.

Onken, Derek and Lars Ruthotto (May 2020). “Discretize-Optimize vs. Optimize-Discretize for Time-Series Regression and Continuous Normalizing Flows”. In: *arXiv.org*. arXiv: .

Onken, Derek et al. (May 2020). “OT-Flow: Fast and Accurate Continuous Normalizing Flows via Optimal Transport”. In: *arXiv.org*. arXiv: .

Osher, Stanley et al. (2018). *Laplacian Smoothing Gradient Descent*. arXiv: .

Parpas, Panos and Corey Muir (2019). *Predict Globally, Correct Locally: Parallel-in-Time Optimal Control of Neural Networks*. arXiv: .

Perona, Pietro et al. (1994). “Anisotropic diffusion”. In: *Geometry-driven diffusion in computer vision*. Springer, pp. 73–92.

Poggio, Tomaso et al. (2017). “Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review”. In: *International Journal of Automation and Computing* 14.5, pp. 503–519.

Raissi, M et al. (Feb. 2019). “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Surveys in Operations Research and Management Science* 378, pp. 686–707.

Rico-Martínez, R et al. (1992). “Discrete- vs. Continuous-time Nonlinear Signal Processing of Cu Electrodissolution Data”. In: *Chemical Engineering Communications* 118.1, pp. 25–48.

Rudin, Leonid I et al. (Nov. 1992). “Nonlinear total variation based noise removal algorithms”. In: *Physica D: Nonlinear Phenomena* 60.1-4, pp. 259–268.

References (cont.)

Ruthotto, Lars and Eldad Haber (2020). “Deep neural networks motivated by partial differential equations”. In: *Journal of Mathematical Imaging and Vision* 62.3, pp. 352–364.

Ruthotto, Lars et al. (2020). “A machine learning framework for solving high-dimensional mean field game and mean field control problems”. In: *Proceedings of the National Academy of Sciences* 117.17, pp. 9183–9193.

Samek, Wojciech et al. (2017). “Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models”. In: *arXiv preprint arXiv:1708.08296*.

Scherzer, Otmar and Joachim Weickert (2000). “Relations between regularization and diffusion filtering”. In: *Journal of Mathematical Imaging and Vision* 12.1, pp. 43–63.

Shafahi, Ali et al. (2018). “Are adversarial examples inevitable?” In: *arXiv preprint arXiv:1809.02104*.

Shin, Yeonjong et al. (2020). *On the Convergence and generalization of Physics Informed Neural Networks*. arXiv: .

Sirignano, J et al. (Dec. 2018). “DGM: A deep learning algorithm for solving partial differential equations”. In: *Computers Chem Engn.* 375.375, pp. 1339–1364.

Thorpe, Matthew and Yves van Gennip (2018). “Deep limits of residual neural networks”. In: *arXiv preprint arXiv:1810.11741*.

Viguerie, Alex and Alessandro Veneziani (2019). “Deconvolution-based stabilization of the incompressible Navier-Stokes equations”. In: *JCP* 391, pp. 226–242.

Wang, Bao et al. (2018). *ResNets Ensemble via the Feynman-Kac Formalism to Improve Natural and Robust Accuracies*. arXiv: .

Weickert, Joachim (Apr. 2009). “Anisotropic Diffusion in Image Processing”. In: pp. 1–184.

References (cont.)


Weinan, E and Bing Yu (2018). "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems". In: *Communications in Mathematics and Statistics* 6.1, pp. 1–12.


Yang, Liu and George Em Karniadakis (Aug. 2019). "Potential Flow Generator with L_2 Optimal Transport Regularity for Generative Models". In: *arXiv.org*. arXiv: .


Zhang, Chiyuan et al. (Jan. 2018). "Theory of Deep Learning IIb: Optimization Properties of SGD". In: arXiv: .