Diving Into Deep Learning: Part 2 – Discussions of Opportunities in Deep Learning

Elizabeth Newman
Emory University

North Carolina State University
April 21, 2023

Disclaimer

- I chose topics that...
  - are interesting and fun!
  - will (hopefully) generate discussion
  - bring a variety of mathematical and practical perspectives
  - were or are potential game-changers in deep learning
  - have easy-to-use code/tutorials to explore
  - ...
- This is a very small sample of what has been done and is ongoing with deep learning.
- I am not advocating that these techniques are "best"
- I am not an expert, just eager to learn from and brainstorm with you!

A Selection of Topics

DL + Dynamics

- Eldad Haber and Lars Ruthotto. "Stable architectures for deep neural networks". In: *Inverse Problems* 34.1 (Dec. 2017), p. 014004
- Ricky T. Q. Chen et al. "Neural Ordinary Differential Equations". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 6572–6583

DL + Physics

- M. Raissi, P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707
- Jiequn Han, Arnulf Jentzen, and Weinan E. "Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510

DL + Generative Modeling

- Lars Ruthotto and Eldad Haber. "An introduction to deep generative modeling". In: *GAMM-Mitteilungen* 44.2 (2021), e202100008

DL + UQ

- Laurent Jospin et al. "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users". In: *IEEE Computational Intelligence Magazine* 17 (May 2022), pp. 29–48

Introductory Repositories

DL + Dynamics

- UvA Deep Learning Tutorials

DL + Physics

- `https://github.com/elizabethnewman/dnn101` [to be added]

DL + Generative Modeling

- `https://github.com/EmoryMLIP/DeepGenerativeModelingIntro`

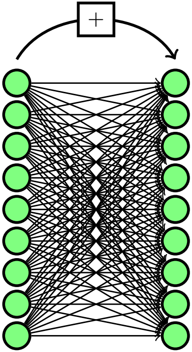
DL + UQ

- Bayesian Neural Networks
- CUQIpy

Things to Think About

- How can we apply mathematical tools to make deep learning easier/more effective/more understandable?
- How can we bring prior knowledge and models into deep learning?
- How can deep learning create new samples?
- How can uncertainty quantification help deep learning and vice versa?

Residual Neural Networks, Continuously



Discrete: Given $\mathbf{u}_0$,

$$\mathbf{u}_j = \mathbf{u}_{j-1} + h\sigma(\mathbf{K}_j\mathbf{u}_{j-1} + \mathbf{b}_j)$$

for layers $j = 1, \ldots, d$.

Continuous:

$$\dot{\mathbf{u}}(t) = \sigma(\mathbf{K}(t)\mathbf{u}(t) + \mathbf{b}(t)) \quad \text{with} \quad \mathbf{u}(0) = \mathbf{u}_0$$

over time interval $[0, T]$.

Stable Deep Neural Networks

$\dot{\mathbf{u}}(t) = \sigma(\mathbf{K}(t)\mathbf{u}(t) + \mathbf{b}(t))$ with $\mathbf{u}(0) = \mathbf{u}_0$ over time interval $[0, T]$

.

Stability

Assume weights evolve slowly in time and σ is monotone increasing. Then forward propagation is stable if

$$\min_{i=1,\ldots,n} \mathrm{Re}(\lambda_i(\mathbf{J}(t))) \leqslant 0 \quad \text{for all } t \in [0, T]$$

where

$$\mathbf{J}(t) \equiv \nabla_{\mathbf{u}} \left[\sigma(\mathbf{K}(t)\mathbf{u} + \mathbf{b}(t))\right] = \mathrm{diag}(\sigma'(\mathbf{K}(t)\mathbf{u}(t) + \mathbf{b}(t)))\mathbf{K}(t)$$

Haber and Ruthotto, "Stable architectures for deep neural networks"

Stable Deep Neural Networks

$$\dot{\mathbf{u}}(t) = \sigma(\mathbf{K}(t)\mathbf{u}(t) + \mathbf{b}(t)) \quad \text{with} \quad \mathbf{u}(0) = \mathbf{u}_0 \quad \text{over time interval } [0, T]$$

.

Stability

Assume weights evolve slowly in time and σ is monotone increasing. Then forward propagation is stable if

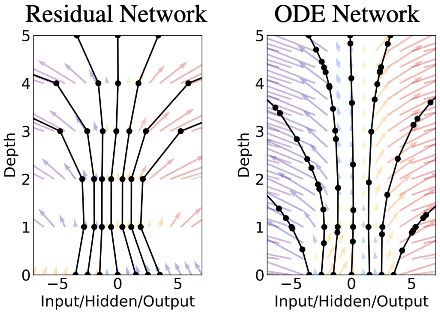
$$\min_{i=1,\ldots,n} \mathrm{Re}(\lambda_i(\mathbf{K}(t))) \leqslant 0 \quad \text{for all } t \in [0, T]$$

where

$$\mathbf{J}(t) \equiv \nabla_{\mathbf{u}} \left[\sigma(\mathbf{K}(t)\mathbf{u} + \mathbf{b}(t))\right] = \mathrm{diag}(\sigma'(\mathbf{K}(t)\mathbf{u}(t) + \mathbf{b}(t)))\mathbf{K}(t)$$

Haber and Ruthotto, "Stable architectures for deep neural networks"

Neural ODEs



$$\dot{\mathbf{u}}(t) = f(\mathbf{u}(t), t, \boldsymbol{\theta}(t)) \implies \mathbf{u}(t_{j+1}) = \mathbf{u}(t_j) + \int_{t_j}^{t_{j+1}} f(\mathbf{u}(s), s, \boldsymbol{\theta}(s))ds$$

Chen et al., "Neural Ordinary Differential Equations"

Discussion: Bringing Mathematical Tools to Deep Learning

- Are there any numerical analysis tools that can change our perspective on DNNs?

  Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. "Deep Equilibrium Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019

  Samy Wu Fung et al. "JFB: Jacobian-Free Backpropagation for Implicit Models". In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2022)

- Are there symmetries and structures we can exploit?

  Michael M. Bronstein et al. "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges". In: *ArXiv* abs/2104.13478 (2021)

- ...

Further Reading

Some Classical References:

- Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778
- Weinan E. "A Proposal on Machine Learning via Dynamical Systems". In: *Communications in Mathematics and Statistics* 5.1 (2017), pp. 1–11
- Yiping Lu et al. "Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 3276–3285

Some Modern References

- Lars Ruthotto and Eldad Haber. "Deep Neural Networks Motivated by Partial Differential Equations". In: *Journal of Mathematical Imaging and Vision* 62.3 (2020), pp. 352–364
- Chris Finlay et al. "How to Train Your Neural ODE: The World of Jacobian and Kinetic Regularization". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020

Physics-Informed Neural Networks

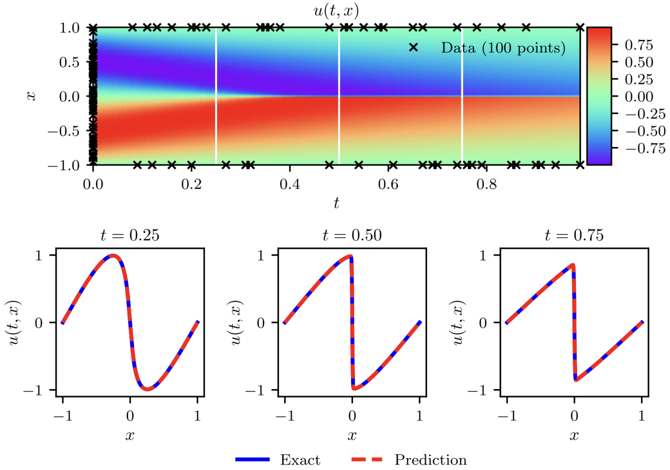
Goal: Learn a network that *approximates* the **solution** to a PDE

$$\mathcal{A}(u) = q \qquad + \text{boundary/initial conditions}$$

Training Problem: Let $\mathcal{B}$ be a set of points on the boundary and let $\mathcal{I}$ be a set of sample points on the interior.

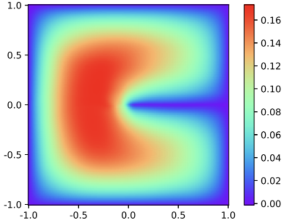
$$\min_{\boldsymbol{\theta}} \Phi^{\text{pinn}}(\boldsymbol{\theta}) \equiv \underbrace{\frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x},u)\in\mathcal{B}} L(f(\mathbf{x},\boldsymbol{\theta}), u)}_{\text{fitting term}} + \underbrace{\frac{1}{|\mathcal{I}|} \sum_{(\mathbf{x},q)\in\mathcal{I}} L(\mathcal{A}(f(\mathbf{x},\boldsymbol{\theta})), q)}_{\text{physics-informed regularization}}$$

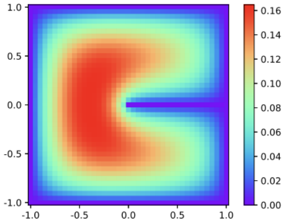
PINNs Example: Burgers' Equation



Raissi, Perdikaris, and Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations"

Deep Ritz Method





Goal: Solve the variational problem for boundary valued problems

$$\min_{u \in \mathcal{H}} \int_{\Omega} \frac{1}{2} |\nabla u(x)| + f(x)u(x)dx$$

Approximate $u(x) \approx F(x, \boldsymbol{\theta})$.

Han, Jentzen, and E, "Solving high-dimensional partial differential equations using deep learning"

Discussion: Combining Models and Deep Learning

- Are there other ways to bring domain knowledge into DNNs?
- With what kinds of models do you work? Can these models be used to regularize DNNs?
- Can we use deep learning for model correction or discovery?
- ...

Further Reading

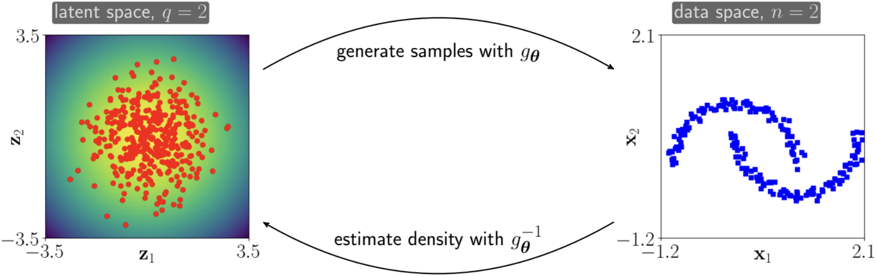
Some Classical References:

- Raissi, Perdikaris, and Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations"
- Han, Jentzen, and E, "Solving high-dimensional partial differential equations using deep learning"
- Jiequn Han, Arnulf Jentzen, and Weinan E. "Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510

Some Modern References:

- Nils Thuerey et al. *Physics-based Deep Learning*. WWW, 2021
- Pi-Yueh Chuang and Lorena Barba. *Experience report of physics-informed neural networks in fluid simulations: pitfalls and frustration*. May 2022
- Lars Ruthotto et al. "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 (2020), pp. 9183–9193

Deep Generative Modeling

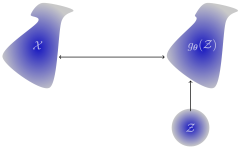


Shout Out! An excellent tutorial on generative modeling by Lars Ruthotto

https://www.math.emory.edu/~lruthot/workshops/dgm/

Ruthotto and Haber, "An introduction to deep generative modeling"

Maximum Likelihood Training



Setup:

- simple distribution $\mathcal{Z}$
- complicated, data distribution $\mathcal{X}$
- parameterized map $g_{\boldsymbol{\theta}}(\mathcal{Z}) \approx \mathcal{X}$

Change of Variables: Approximate the likelihood $p_{\mathcal{X}}(\mathbf{x})$ via

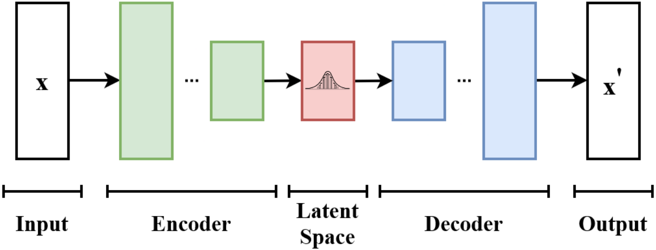
$$\begin{aligned} p_{\mathcal{X}}(\mathbf{x}) \approx p_{\boldsymbol{\theta}}(\mathbf{x}) &= p_{\mathcal{Z}}(g_{\boldsymbol{\theta}}^{-1}(\mathbf{x}))\,\det \nabla g_{\boldsymbol{\theta}}^{-1}(\mathbf{x}) \\ &= (2\pi\sigma)^{-n/2} \exp\left(-\frac{1}{2\sigma}\|g_{\boldsymbol{\theta}}(\mathbf{z})\|_2^2\right) \end{aligned}$$

Goal: Learn a generator $g_{\boldsymbol{\theta}}^{-1} : \mathcal{Z} \to \mathcal{X}$ using maximum likelihood training by minimizing

$$J_{\mathrm{ML}}(\boldsymbol{\theta}) \equiv \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}[-\log p_{\boldsymbol{\theta}}(\mathbf{x})] \approx \frac{1}{s}\sum_{i=1}^{s}\left(\frac{1}{2}\|g_{\boldsymbol{\theta}}^{-1}(\mathbf{x}_i)\|_2^2 - \log\,\det \nabla g_{\boldsymbol{\theta}}^{-1}(\mathbf{x}_i)\right)$$

Ruthotto and Haber, "An introduction to deep generative modeling"

Variational Autoencoders



Goal: Construct encoder $e_{\boldsymbol{\psi}} : \mathcal{X} \to (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\mathcal{Z} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and decoder $g_{\boldsymbol{\theta}} : \mathcal{Z} \to \mathcal{X}$ by minimizing

$$J_{\text{ELBO}}(\boldsymbol{\psi}, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\mathbb{E}_{\mathbf{z}\sim e_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x})}[-\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - \log p_{\mathcal{Z}}(\mathbf{z}) + \log e_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x})]$$

Kingma and Welling, "Auto-Encoding Variational Bayes"; Ruthotto and Haber, "An introduction to deep generative modeling", Image from Wikipedia

Discussion: Learning Maps from Distributions to Distributions

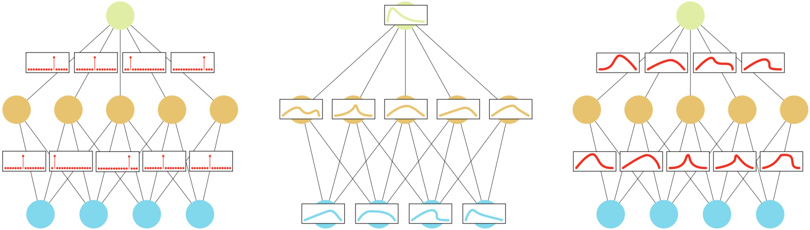
- How should we compare complicated probability distributions?
- How should we choose the latent distribution?
- How can we use generators in other deep learning tasks?
- ...

Further Reading

Some References:

- Laurent Dinh, David Krueger, and Yoshua Bengio. *NICE: Non-linear Independent Components Estimation*. 2015
- Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. 2017
- Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014
- Chin-Wei Huang et al. "Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization". In: *International Conference on Learning Representations*. 2021
- ...

Bayesian Neural Networks



(a) (b) (c)

Bayesian Neural Networks

Setup: prior distribution over weights $p(\boldsymbol{\theta})$ (e.g., $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ per layer)

Forward Propagate: For $i = 1, \ldots, N$,

$$\begin{aligned} \boldsymbol{\theta}_i &\sim p(\boldsymbol{\theta}) \\ \mathbf{c}_i &= F(\mathbf{y}, \boldsymbol{\theta}_i) \end{aligned}$$

Evaluate: Ensemble!

$$\hat{\mathbf{c}} = \frac{1}{N} \sum_{i=1}^{N} F(\mathbf{y}, \boldsymbol{\theta}_i)$$

Backward Propagate: AD using $\hat{\mathbf{c}}$

Update: SG variant on learnable weights $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$

Active Learning

Goal: Query next, best training sample to reduce uncertainty
Given:

- a trained model $F(\cdot, \boldsymbol{\theta})$ trained on $\mathcal{T}$
- a set of potential next queries $\mathcal{U} = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$
- covariance matrix of uncertainty

$$\boldsymbol{\Sigma}_{\mathbf{c}|\mathbf{y}} = \frac{1}{N-1} \sum_{i=1}^{N} (F(\mathbf{y}, \boldsymbol{\theta}_i) - \hat{\mathbf{c}})(F(\mathbf{y}, \boldsymbol{\theta}_i) - \hat{\mathbf{c}})^\top$$

Uncertainty Score: $s(\mathbf{y}_i, \boldsymbol{\theta})$ for $i = 1, \ldots, k$; e.g.,

$$s(\mathbf{y}_i, \boldsymbol{\theta}) = \max_{i=1,\ldots,k} g(\boldsymbol{\Sigma}_{\mathbf{c}_i|\mathbf{y}_i})$$

where

$$i^* \in \arg\max_{j=1,\ldots,k} s(\mathbf{y}_i, \boldsymbol{\theta})$$

Choose input $\mathbf{y}_{i*}$ that yields highest uncertainty score and update

$$\mathcal{T} \leftarrow \mathcal{T} \cup \{(\mathbf{y}_{i*}, \texttt{oracle}(\mathbf{y}_{i*}))\}$$
$$\mathcal{U} \leftarrow \mathcal{U} \backslash \{\mathbf{y}_{i*}\}$$

Retrain!

Discussion: UQ for DL and DL for UQ

- Why UQ for DL?
- What are the major challenges using UQ for DL?
- Can DL make UQ easier?
- ...

Further Reading

Some References

- Hao Wang and Dit-Yan Yeung. "A Survey on Bayesian Deep Learning". In: *ACM Comput. Surv.* 53.5 (Sept. 2020)
- Charles Blundell et al. *Weight Uncertainty in Neural Networks*. 2015
- Rohit K. Tripathy and Ilias Bilionis. "Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification". In: *Journal of Computational Physics* 375 (2018), pp. 565–588
- Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297
- Vu-Linh Nguyen, Sébastien Destercke, and Eyke Hüllermeier. "Epistemic Uncertainty Sampling". In: *Discovery Science*. Ed. by Petra Kralj Novak, Tomislav Šmuc, and Sašo Džeroski. Cham: Springer International Publishing, 2019, pp. 72–86
- `https://www.cs.toronto.edu/~duvenaud/distill_bayes_net/public/`
- ...

Thanks for a great week!

References I

Abdar, Moloud et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information Fusion* 76 (2021), pp. 243–297.

Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "Deep Equilibrium Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

Blundell, Charles et al. *Weight Uncertainty in Neural Networks*. 2015.

Bronstein, Michael M. et al. "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges". In: *ArXiv* abs/2104.13478 (2021).

Chen, Ricky T. Q. et al. "Neural Ordinary Differential Equations". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 6572–6583.

Chuang, Pi-Yueh and Lorena Barba. *Experience report of physics-informed neural networks in fluid simulations: pitfalls and frustration*. May 2022.

Dinh, Laurent, David Krueger, and Yoshua Bengio. *NICE: Non-linear Independent Components Estimation*. 2015.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. 2017.

References II

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

Finlay, Chris et al. "How to Train Your Neural ODE: The World of Jacobian and Kinetic Regularization". In: *Proceedings of the 37th International Conference on Machine Learning*. ICML'20. JMLR.org, 2020.

Fung, Samy Wu et al. "JFB: Jacobian-Free Backpropagation for Implicit Models". In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2022).

Haber, Eldad and Lars Ruthotto. "Stable architectures for deep neural networks". In: *Inverse Problems* 34.1 (Dec. 2017), p. 014004.

Han, Jiequn, Arnulf Jentzen, and Weinan E. "Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510.

— ."Solving high-dimensional partial differential equations using deep learning". In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510.

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

References III

Huang, Chin-Wei et al. "Convex Potential Flows: Universal Probability Distributions with Optimal Transport and Convex Optimization". In: *International Conference on Learning Representations*. 2021.

Jospin, Laurent et al. "Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users". In: *IEEE Computational Intelligence Magazine* 17 (May 2022), pp. 29–48.

Kingma, Diederik P. and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014.

Lu, Yiping et al. "Beyond Finite Layer Neural Networks: Bridging Deep Architectures and Numerical Differential Equations". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 3276–3285.

Nguyen, Vu-Linh, Sébastien Destercke, and Eyke Hüllermeier. "Epistemic Uncertainty Sampling". In: *Discovery Science*. Ed. by Petra Kralj Novak, Tomislav Šmuc, and Sašo Džeroski. Cham: Springer International Publishing, 2019, pp. 72–86.

Raissi, M., P. Perdikaris, and G.E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707.

References IV

Ruthotto, Lars and Eldad Haber. "An introduction to deep generative modeling". In: *GAMM-Mitteilungen* 44.2 (2021), e202100008.

— ."Deep Neural Networks Motivated by Partial Differential Equations". In: *Journal of Mathematical Imaging and Vision* 62.3 (2020), pp. 352–364.

Ruthotto, Lars et al. "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 (2020), pp. 9183–9193.

Thuerey, Nils et al. *Physics-based Deep Learning*. WWW, 2021.

Tripathy, Rohit K. and Ilias Bilionis. "Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification". In: *Journal of Computational Physics* 375 (2018), pp. 565–588.

Wang, Hao and Dit-Yan Yeung. "A Survey on Bayesian Deep Learning". In: *ACM Comput. Surv.* 53.5 (Sept. 2020).