Data Science for Engineering: Open thesis topics

Jun.-Prof. Dr. Sebastian Peitz

Paderborn University

October 29, 2021

This PDF file contains a list of open topics for Master (and possibly Bachelor) theses offered by the Data Science for Engineering group at Paderborn university (www.cs.upb.de/dse). If you are interested in one of the topics, then please read the description as well as the referenced literature carefully, and

prepare a short document ($\approx$ 2 pages) where you summarize the thesis content in your own words and briefly explain how you are going to approach the topic of your thesis, for instance in the form of a short work program.

You may use the template for thesis proposals: https://en.cs.upb.de/dse/teaching/templates. The resulting document can then serve as the basis for the full proposal, see https://cs.upb.de/en/studies/study-elements/general-guidelines/master-thesis (in particular point 3) for details. After sending me the document, we can schedule a meeting to further discuss the details.

Important note: all thesis topics offered by our group require a certain level of knowledge in linear algebra and calculus. A clean mathematical description of the used methods and concepts is absolutely necessary for a good thesis. You may take a look at the following very nice introductory paper on deep learning to get an idea what a good mathematical description should look like:
Higham & Higham (2019). Deep learning: An introduction for applied mathematicians. SIAM Review, 61(4), 860-891. URL: https://epubs.siam.org/doi/pdf/10.1137/18M1165748

Open Topics:

1 Multiobjective training of generative adversarial networks

Generative adversarial networks (GANs) [3] are an extremely powerful tool to generate "fake" data (e.g., images) that are barely indistinguishable from reality. This is not only helpful for the generation of images, but it can also be applied in many physical situations, for instance to reconstruct a complex physical measurement (such as the temperature distribution within a romm) from only a few measurements.

GANs consist of two neural networks, a *generator* – which generates the artificial data – and a *discriminator* which has to decide whether the presented data is real (i.e., from a training data set) or fake (i.e., created by the generator). Training is usually performed alternatingly, meaning that we perform an optimization step for the generator, then for the discriminator. We then go back to the generator and repeat this procedure until our training algorithm converges.

The topic of this thesis is to consider the training of the generator and the discriminator as two separate objective functions, and to then optimize them simultaneously using methods from multiobjective optimization such as the *multiobjective gradient descent method* [2]. The solution is an *optimal trade-off* between generator and discriminator performance, and a suitable parametrization of the procedure has to be identified in order to find a suitable trade-off solution. The resulting GAN then needs to be compared against a GAN trained by the standard procedure.

Literature

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[2] J. Fliege and B. F. Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

2 Flow field reconstruction using physics-informed neural network architectures

Generative adversarial networks (GANs) [3] are an extremely powerful tool to generate "fake" data (e.g., images) that are barely indistinguishable from reality. This is not only helpful for the generation of images, but it can also be applied in many physical situations, for instance to reconstruct a complex physical measurement (such as the temperature distribution within a romm) from only a few measurements.

In recent years, *physics-informed* neural network [4, 5] have become very popular. Here, the task is to train a neural network with the additional constraint that a known physical law (for instance, a *partial differential equation* describing the physics behind the data) has to be satisfied.

The task in this thesis is to give a detailed overview of physics-informed neural network architectures, and to compare several on a standard data set from fluid dynamics. A particular focus will be set on the task to reconstruct complex physical data from a small number of realistically measurable data points.

Literature

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[4] 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. *Journal of Computational Physics*, 378:686–707, 2019.

[5] L. Yang, D. Zhang, and G. E. M. Karniadakis. Physics-informed Generative Adversarial Networks for stochastic differential equations. *SIAM Journal on Scientific Computing*, 42(1):292–317, 2020.

3 An extension of Neural ODEs to variable layer widths

Residual neural networks [6] are nowadays the mots popular approach to create deep neural networks with a very large number of layers. The concept of a residual network is to not consider each layer as a general function transformation, but as a function that learns a residual. It was soon noticed that this approach is related to the numerical discretization of *ordinary differential equations*, i.e., equations of the form

$$\frac{dx}{dt} = f(x),$$

to which a numerical solution can be approximated on a discrete grid $\{x_0, x_1, \ldots, x_n\}$

$$f(x_i) = \frac{dx_i}{dt} \approx \frac{x_{i+1} - x_i}{\Delta t} \quad \Leftrightarrow \quad x_{i+1} \approx x_i + \Delta t f(x_i).$$

In terms of residual networks, this results in so-called *neural ordinary differential equations* (or *Neural ODEs*) [7].

In this context, one challenge is that the state of an ODE does not change in size, which means that all layers of a neural ODE have the same width. As this may not always be necessary or even advantageous, the task of this thesis is to extend the Neural ODE approach from [7] to variable layer width by including a sparsity constraint in the training phase. A sparsity constraint is realized by a penalty term (cf., e.g., [8]) to the objective function of minimizing the training error. This way, as many connections as possible are set to zero and consequently – even though the original network has the same width in each layer – a large number of neurons are "deactivated", which leads to variable layer sizes. The task is to implement this sparsity constraint and to validate the resulting training method on several benchmark examples.

Literature

[6] K. He, X. Zhang, R. Shaoqing, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural Ordinary Differential Equations. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[8] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

4 Study of the Fourier Neural Operator for a Flow Control Problem

Very recently, the *Fourier Neural Operator* was introduced in [9]. This is a very specific neural network architecture that is suited to provide the numerical solution of complex dynamical systems (for instance, of fluid flows or – more generally – *partial differential equations*) in a very short time. This makes it very interesting for control problems, where the task is to optimally steer a system towards a desired state in real time. To this end, a *surrogate model* needs to be constructed form data that can be solved in such a short time that real-time capability is achieved [10].

The task of this thesis is to study the Fourier Neural Operator in detail and to assess its potential for the control of fluid flows in real-time. Several examples of increasing complexity shall be studied in order to reveal the potential as well as the shortcomings.

Literature

[9] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv:2010.08895*, 2020.

[10] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and Computational Fluid Dynamics*, 34:577–591, 2020.

5 Development of a reinforcement learning interface for the finite element solver FEniCS

FEniCS (https://fenicsproject.org, [11]) is a highly scalable and easy to use python library to solve partial differential equations using the method of *finite elements*. This way, complex physics simulations can be performed very efficiently and with little implementation effort. FEniCS possesses a very large and continuously growing community, and it is today one of the standard tools for complex physics simulations.

The task of this thesis is to develop and implement an interface to use FEniCS within a reinforcement learning environment. *Reinforcement learning* (RL) [12] is a machine learning technique that learns good action policies through "trial and error". Over time, the RL agent learns which actions to take in order to achive a desired goal. The efficacy of RL for physics simulations has been demonstrated numerous times, see, for instance, [13]. In order to easily apply RL to various systems, a standardized interface for data-collection and control nees to be developed and validated. To this end, several examples of increasing complexity will be studied.

Literature

[11] Anders Logg, Kent-Andre Mardal, and Garth N. Wells, editors. *Automated Solutions of Differential Equations by the Finite Element Method - The FEniCS Book*. Springer Berlin Heidelberg, 2012.

[12] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[13] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, 865:281–302, 2019.