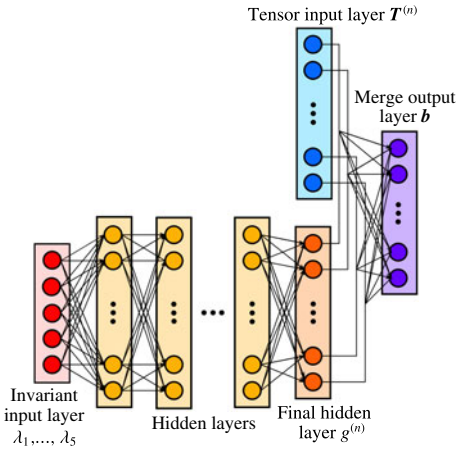
Downloaded from https://www.cambridge.org/core. IP address: 27.125.189.31, on 23 Apr 2021 at 05:20:41, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/jfm.2016.803

Focus on **Fluids** journals.cambridge.org/focus

Deep learning in fluid dynamics

J. Nathan Kutz†

Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA



It was only a matter of time before deep neural networks (DNNs) – deep learning – made their mark in turbulence modelling, or more broadly, in the general area of high-dimensional, complex dynamical systems. In the last decade, DNNs have become a dominant data mining tool for big data applications. Although neural networks have been applied previously to complex fluid flows, the article featured here (Ling *et al.*, *J. Fluid Mech.*, vol. 807, 2016, pp. 155–166) is the first to apply a true DNN architecture, specifically to Reynolds averaged Navier Stokes turbulence models. As one often expects with modern DNNs, performance gains are achieved over competing state-of-the-art methods, suggesting that DNNs may play a critically enabling role in the future of modelling complex flows.

Key words: computational methods, low-dimensional models, turbulence modelling

1. Introduction

Neural networks were inspired by the Nobel prize winning work of Hubel and Wiesel on the primary visual cortex of cats (Hubel & Wiesel 1962). Their seminal experiments showed that neuronal networks where organized in hierarchical layers of cells for processing visual stimulus. The first mathematical model of a neural network, termed the Neocognitron in 1980 (Fukushima 1980), had many of the characteristic features of today's deep neural networks (DNNs), which are typically between 7–10 layers, but more recently have been scaled to hundreds of layers for certain applications. The recent success of DNNs has been enabled by two critical components: (i) the continued growth of computational power and (ii) exceptionally large labelled data sets which take advantage of the power of a multi-layer (deep) architecture. Indeed, although the theoretical inception of DNNs has an almost

† Email address for correspondence: kutz@uw.edu

doi:10.1017/jfm.2016.803

Downloaded from https://www.cambridge.org/core. IP address: 27.125.189.31, on 23 Apr 2021 at 05:20:41, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/jfm.2016.803

four-decade history, the analysis and training of a DNN using the ImageNet data set in 2012 (Krizhevsky *et al.* 2012) provided a watershed moment for deep learning (LeCun *et al.* 2015). DNNs have since transformed the field of computer vision by dominating the performance metrics in almost every meaningful computer vision task intended for classification and identification. Remarkably, DNNs were not even listed as one of the top 10 algorithms of data mining in 2008 (Wu *et al.* 2008). But in 2016, its growing list of successes on challenge data sets make it perhaps the most important data mining tool for our emerging generation of scientists and engineers.

Data methods are certainly not new in the fluids community. Computational fluid dynamics has capitalized on machine learning efforts with dimensionality-reduction techniques such as proper orthogonal decomposition or dynamic mode decomposition, which compute interpretable low-rank modes and subspaces that characterize spatio-temporal flow data (Holmes *et al.* 1998; Kutz *et al.* 2016). POD and DMD are based on the singular value decomposition which is ubiquitous in the dimensionality reduction of physical systems. When coupled with Galerkin projection, POD reduction forms the mathematical basis of reduced-order modelling, which provides an enabling strategy for computing high-dimensional discretizations of complex flows (Benner *et al.* 2015).

The success of dimensionality reduction in fluids is enabled by (i) significant performance gains in computational speed and memory and (ii) the generation of physically interpretable spatial and/or spatio-temporal modes that dominate the physics. Thus computations are enabled and critical physical intuition gained. Such success is tempered by two well-known failings of POD/DMD based reductions: (i) their inability to capture transient, intermittent and/or multi-scale phenomenon without significant tuning and (ii) their inability to capture invariances due to translation, rotation and/or scaling. DNNs are almost diametrically opposed in their pros and cons. Specifically, DNNs are well suited for extracting multi-scale features as the DNN decomposition shares many similarities with wavelet decompositions, which are the computational work horse of multi-resolution analysis. Moreover, translations, rotations and other invariances are known to be easily handled in the DNN architecture. These performance gains are tempered by the tremendous computational cost of building a DNN from a large training set and the inability of DNN to produce easily interpretable physical modes and/or features.

2. Overview of DNNs in turbulence applications

Turbulent flows generally exhibit multi-scale (spatial and temporal) physics that are high dimensional with rotational and translational intermittent structures also present. Such data provide an opportunity for DNN to make an impact in the modelling and analysis of turbulent flow fields. Ling, Kurzawski & Templeton (2016) have proposed using DNNs for Reynolds averaged Navier Stokes (RANS) models which are widely used because of their computational tractability in modelling the rich set of dynamics induced by turbulent flows. In this highlighted body of work, the specific aim is to use DNNs to build an improved representation of the Reynolds stress anisotropy tensor from high-fidelity simulation data. Remarkably, despite the widespread success of DNNs at providing high-quality predictions in complex problems, there have been only limited attempts to apply deep learning techniques to turbulence. Thus far, these attempts have been limited to a couple hidden layers (Zhang & Duraisamy 2015). Ling *et al.* (2016) move to DNNs by constructing 8–10 hidden layers, making it truly a deep network. But this highlighted work does so much more than simply

Downloaded from https://www.cambridge.org/core. IP address: 27.125.189.31, on 23 Apr 2021 at 05:20:41, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/jfm.2016.803

build a DNN. Indeed, the authors construct a specialized neural network architecture which directly embeds Galilean invariance into the neural network predictions. This neural network is able to predict not only the anisotropy eigenvalues, but the full anisotropy tensor while preserving Galilean invariance. This invariance preserving DNN is critical for respecting physical properties in order to provide significant performance gains in prediction. The DNN is trained and evaluated on a database of flows for which both RANS and high-fidelity data are available.

The specific DNN architecture used by the authors is referred to as the tensor basis neural network. This network architecture is capable of embedding rotational invariance by enforcing that the predicted anisotropy tensor lies on a basis of isotropic tensors. Rotational invariance signifies that the physics of the fluid flow does not depend on the orientation of the coordinate frame of the observer. This is a fundamental physical principle, and it is important that any turbulence closure obeys it. Otherwise, the machine learning model evaluated on identical flows with the axes defined in different directions could yield different predictions. Enforcement of the rotational invariance in this DNN shows substantial improvement over a more generic feed forward multi-layer perceptron that does not embed Galilean invariance. Training of their DNN was performed on a database of flows where both high-fidelity as well as RANS results were available. The flow database represents various flow configurations, thus the DNN is not simply interpolating or matching to similar flows. Rather, the DNN extracts information about the underlying flow in a principled manner.

The authors demonstrated that their DNN configuration was significantly more accurate than either of two conventional RANS models on two different test cases. Moreover, on a wavy wall test case which had a different geometry than any of the training cases, their DNN was able to provide improved predictions, suggesting the method was doing more than simply interpolating the training data. Additionally, on a duct flow test, the DNN was capable of improving predictions despite the test set being at a significantly different Reynolds number. Ultimately, the results suggest that the physics respecting DNN trained with embedded Galilean invariance can outperform, often significantly, other RANS turbulence models.

3. The future of DNNs for fluids modelling

DNNs will almost certainly have a transformative impact on modelling high-dimensional complex systems such as turbulent flows. The successes with many complex data sets will compel researchers to utilize this rapidly emerging data analysis tool for improving predictive capabilities. DNNs represent a paradigm shift for the community. Whereas many innovations have often been inspired from expert-in-the-loop intuition and physically interpretable models, DNNs have challenged these traditional notions by building prediction engines that simply outperform competing methods without providing clear evidence of why they are doing so. To some extent, the application of DNNs to turbulent flows will bring awareness to the fluids community of the two cultures of statistics and data science (Breiman 2001). These two outlooks are centred around the concepts of machine learning and statistical learning. The former focuses on prediction (DNNs) while the latter is concerned with inference of interpretable models from data (POD/DMD reductions). Although both methodologies have achieved significant success across many areas of big data analytics, the physical and engineering sciences have primarily focused on interpretable methods.

Downloaded from https://www.cambridge.org/core. IP address: 27.125.189.31, on 23 Apr 2021 at 05:20:41, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/jfm.2016.803

Despite its successes, significant challenges remain for DNNs. Simple questions remain wide open: (i) How many layers are necessary for a given data set? (ii) How many nodes at each layer are needed? (iii) How big must my data set be to properly train the network? (iv) What guarantees exist that the mathematical architecture can produce a good predictor of the data? (v) What is my uncertainty and/or statistical confidence in the DNN output? (vi) Can I actually predict data well outside of my training data? (vii) Can I guarantee that I am not overfitting my data with such a large network? The list goes on. These questions remain central to addressing the long-term viability of DNNs. The good news is that such topics are currently being intensely investigated by academic researchers and industry (Google, Facebook, etc.) alike. Undoubtedly, the next decade will witness significant progress in addressing these issues. From a practical standpoint, the work of Ling *et al.* (2016) determine the number of layers and nodes based upon prediction success, i.e. more layers and more nodes do not improve performance. Additionally, cross-validation is imperative to suppress overfitting. As a general rule, one should never trust results of a DNN unless rigorous cross-validation has been performed. Cross-validation plays the same critical role as a convergence study of a numerical scheme.

Given the computational maturity of DNNs and how readily available they are (see Google's open source software called TensorFlow: tensorflow.org), it is perhaps time for part of the turbulence modelling community to adopt what has become an important and highly successful part of the machine learning culture: challenge data sets. Donoho argues (Donoho 2015), and I am in complete agreement, that challenge data sets allow researchers a fair comparison of their DNN innovations on training data (publicly available to all) and test data (not publicly available, but accessible with your algorithm). Importantly, this would give the fluids community their own ImageNet data sets to help generate reproducible and validated performance gains on DNNs for applications on complex flows. Perhaps Ling, Kurzawski and Templeton can help push the community forward in this way.

References

BENNER, P., GUGERCIN, S. & WILLCOX, K. 2015 A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **57**, 483–531.

BREIMAN, L. 2001 Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Stat. Sci.* **16** (3), 199–231.

DONOHO, D. L. 2015 *50 Years of Data Science*. Tukey Centennial Workshop.

FUKUSHIMA, F. 1980 A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**, 193–202.

HOLMES, P., LUMLEY, J. & BERKOOZ, G. 1998 *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press.

HUBEL, D. H. & WIESEL, T. N. 1962 Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **160**, 106–154.

KRIZHEVSKY, A., SUTSKEVER, I. & HINTON, G. 2012 Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Proc. Syst.* 1097–1105.

KUTZ, J. N., BRUNTON, S., BRUNTON, B. & PROCTOR, J. 2016 *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM.

LECUN, Y., BENGIO, Y. & HINTON, G. 2015 Deep learning. *Nature* **521** (7553), 436–444.

LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech* **807**, 155–166.

WU, X., KUMAR, V., QUINLAN, J., GHOSH, J., YANG, Q., MOTODA, H., MCLACHLAN, G., NG, A., LIU, B., PHILIP, S. & ZHOU, Z. 2008 Top 10 algorithms in data mining. *Know. inf. sys.* **14** (1), 1–37.

ZHANG, Z. & DURAISAMY, K. 2015 Machine learning methods for data-driven turbulence modeling. *AIAA Aviation* 2015–2460.