Review of "Generalization properties of neural networks trained on Lorenz systems"

June 24, 2019

1 Review

The paper presents a study of the overfitting properties of feed-forward and 1-d convolutional neural networks and whether they can capture the dynamics along with the influence of an external forcing parameter, in a 40-dimensional (grid-points of the discretized spatial state) Lorenz-95/96 and a three-dimensional Lorenz-63 system. In Lorenz-63, they show that neural networks struggle to extrapolate the dynamics on attractor regions under-represented in the training data, even in short-term predictions. Long-term predictions starting from these regions do not resemble the ones from the system equations. Moreover, the authors train a small neural network on the whole data and identify neurons of the neural network that are responsible for capturing the dynamics of specific parts in the training data. Thus, they argue that there the neural network learns subnetworks responsible for the dynamics locally and does not learn a global model for the dynamics, which would probably generalize better. In Lorenz-95, they find that including the information of the external forcing variation might degrade performance and argue that the hypothesis and quest to identify models that work on past climate data, expecting them to work well on future data might be erroneous.

Quality

- The quality of the paper is good.
- The authors do not reference related work on generalization properties of neural networks to unseen data, or other machine learning models designed for non-stationary time series.
- They do raise two important open problems in data-driven prediction of dynamical systems. Namely how well neural networks generalize and if they can learn from non-stationarity data.
- The code and data used to generate the results are publicly accessible which enables reproduction of the research.
- The argumentation in Section 2, about whether the network learns only one or many mappings for different regions is inconsistent. The two representations are mathematically equivalent. The impression the reader gets about what the authors are trying to express, is whether different parts of the network are responsible for different (local dynamics) parts of the training data.
- In the Lorenz-63 system, the authors try to answer the aforementioned question (identify specific parts of the neural network that are responsible for capturing the dynamics locally), by analyzing the activation levels of the neurons of the neural network, freezing neurons that are mostly active on specific regions, to check the deterioration of performance on other regions (for a model trained on the whole training data). This argument, that parts of the neural network are responsible for local models of the training data, is very interesting. Especially for large (relative to the application) overparametrized models, this argument makes sense. However, it has to be tested systematically in larger models (maybe a large model applied to the Lorenz-96), and in a more structured way to be accepted as a general attribute of neural networks, as the small network used in this study might be misleading.

- The authors do not explain the training procedure and how they cope against overfitting in the CNN applied to Lorenz-95. Especially in the low data regime, the absence of measures against overfitting can have a detrimental influence on the performance on the test dataset.

- Since the neural network is forecasting a deterministic system with full state information, the prediction accuracy reported in the Appendix on page 17, seems quite low. In the provided plots, the networks seems to be forecasting inaccurately, as the difference in the plots even at early timesteps is obvious.

- The generalization of the study is problematic. The study is limited to feedforward neural networks, with a one to one training scheme. By changing the loss to include some stability metric, or long-term performance, trying out different architectures, regularization techniques, etc. the generalization properties might be improved. The first problem of extrapolation is a general pitfall of data-driven approaches, however, the second problem of non-stationarity might be alleviated with more sophisticated architectures. As reported in Section 3.4, many trained CNN networks are not stable, because they were trained for single step forecasts. This is expected, as the neural networks are not trained for long-term forecasting. RNNs can be used in these low-dimensional systems, backpropagating the gradient many timesteps in the past to ensure stability. The authors use a posteriori analysis of the networks to identify the stable ones. Moreover, the models are applied to non-stationary timeseries with external forcing, which is a really challenging application. The selected models and the training procedure used is not adequate to extract general conclusions. For example, complex RNN architectures that try to capture multiple time scales, or Reservoir Computing approaches might work better. The conclusions of the paper should be specific only to feedforward neural networks. The arsenal of machine learning tools to counter these open problems is much wider.

- The second question the paper poses, is a very interesting one. Real time-series data are most of the time non-stationary. Even though many neural architectures have been successfully used in seasonal or non-stationary data, it is not clear if the networks can actually learn varying dynamics, or how efficient they are in that. One solution could be to train networks on the fly as new data come in. There is available literature on applying machine learning approaches for non-stationary data. Even though the model used in this study appears to be incapable of generalizing, this might not hold for other models. For example, the forcing was provided as an additional input to the network. However, we do know that this external forcing is not the same type of input as the rest. This information could be provided in a different way to the network.

- The statement "... the trajectories of the network forecast simply point back towards the region included in the training." regarding the behavior of the neural network in regions of the phase space not included in the training data, seems rather arbitrary. Since the neural network is not trained in these regions the behavior can be anything.

Clarity

- Good. The results of the paper and the conclusions are clearly explained.

Originality

- The first problem of generalizing to unseen data is a well-known one. As a data-driven approach, neural networks have a hard time to extrapolate to unseen regions in the dataspace. This is addressed in many previous studies, not only related with dynamical systems. Regularization, coupling neural networks with equations, adding constraints etc. are known measures to cope with this deficiency. Most data-driven methods suffer from this problem. It is not surprising to see that a small neural network trained on the left wing of the Lorenz-63 attractor cannot generalize to the vastly different dynamics (in terms of data, not equations) of the right wing. The situation is expected to worsen as the models grows bigger (overfitting easier).

- Implicitly, the authors state a very interesting question, whether the neural networks learn sub-networks that are responsible for modeling the dynamics locally in parts of the training data. In the Lorenz-63 system, they manage to demonstrate this in terms of identifying the neuron that seems to be responsible for a specific part of the training data (right or left wing). Whether this argument holds for large models or other more complicated dynamics and is not specific to this study, remains open. However, it is an original and interesting finding that needs to be tested for more general settings (large networks, more applications).

- The second issue raised, is whether NN can learn dynamics that evolve based on external forcing. This is connected with the known open problem of neural networks learning from non-stationary data/dynamics. The architectures proposed in the study are not compared with other state-of-the art approaches, like reservoir computers, RNNs, ARIMA models, etc. and long-term results are not presented (from iterative forecasting) so it is not straightforward to judge their efficiency.

Confidence

- The reviewer is confident but not absolutely certain.

Recommendation

- Accept subject to major revisions.
- Accept after the revision of the issues raised above, or at least referencing them in the text. Especially for the argument about the sub-networks having learned local dynamics, a bigger model needs to be tested. I doubt there is any model applied in practice with only 8 neurons. ML models applied in practice have thousands to millions of parameters. In order to support this claim, it has to be tested on large models in a systematic way, which is however, challenging to achieve.

2 Typos

1. Page 1, line 18, typo "... the widely studied ..."
2. Page 2, line 5, typo "... the widely In in this paper ..."
3. Page 4, line 24, typo "The neural networks are trained..."
4. Page 6, line 19, In order to avoid misconception, the following reformulation would help the reader: "Figure 2 shows the training data and the forecast error for a network ..."
5. Page 15 line 3, typo "... of how to test test the reconstruction at training time ..."