# Review of “Generalization properties of neural networks trained on Lorenz systems” by Scher and Messori

Note I wrote these comments before viewing those from the first set of review comments.

This paper addresses the questions of whether neural networks can:

1. learn to simulate dynamical systems if the training data does not include the full range of possible system states and
2. learn to project the effect of a changing forcing on a dynamical system into the future, as the forcing increases beyond the range seen in training.

This is done with experiments using two fairly simple dynamical systems. The machine learning algorithms do not perform very well at either task overall and the authors argue that this illustrates challenges for machine-learning applications. It is suggested that the results have relevance for applying machine learning to simulate the climate system.

My overall opinion is that whilst the questions the paper addresses are interesting and important, and the results are well-presented on the whole, the experiments performed are not very close to how neural networks would be applied in reality, so it's not clear if they have real-world applications (even notwithstanding the simplicity of the systems being studied). In particular, the performance of the neural networks on reproducing the training data often looks so poor that they would not be used in an application, or the training performance is not presented in enough detail, so it's not clear that the results would apply to real-world applications that would require good validation performance. Also, the changes in forcings applied in the second part seem a lot larger than for applications that neural networks might be considered for. In addition, the authors' experiments on the Lorenz '63 system have used a particular neural network design, where the full state is predicted at every time step (it's not clear if this is the case for the Lorenz '95 system as well) – this may be expected to work worse than other designs where only the change in the state is predicted at each step, or where bias-correction of an approximate dynamical model is performed, and the results here are not clearly generalisable to those set ups.

I have given more detailed comments below. I think the work could eventually be publishable, if the comments are adequately addressed. Since my comments are quite substantial, it could be acceptable to just include the L63 experiments on training on part of the attractor and the L95 experiments on response to forcing – the L63 experiments on responding to a parameter change seem less applicable to real-world cases like predicting climate change. I do encourage the authors to continue with this line of investigation, which I think is potentially very valuable.

**Most significant comments:**

1. Training performance of neural networks:
    a. For the Lorenz '63 experiments, the ability of the trained neural network to reproduce the attractor appears quite poor (fig.1), and much worse than in the results presented by Zhang (2017), whose work the authors say they are following. A model with such performance would not be used in any real-world application I can think of, and the later results may be much worse than for a well-trained system. It should probably be checked that all of the important steps in the prior work were followed, and if that does not resolve the problem, different architectures tried (e.g. using more neurons) until a good simulation of the attractor is produced.

b. More diagnostics for the performance of the Lorenz '95 networks on the training data and the equivalent portion of the test data should be presented to indicate the system's performance and the degree of overfitting e.g. MAE of single-timestep predictions relative to the variance of the system's tendencies.
c. In the experiments testing how well the networks capture the response to a changing forcing, it needs to be shown how well the networks reproduce the trend in the training data. For the results to be applicable to predicting the path of global warming, for example, there should be a discernible trend in the training data and the neural networks should reproduce it with an accuracy similar to what climate models achieve, else they would be deemed to be unsuitable for use in prediction.

2. Forcing experiments:
   a.
      - The changes in the forcing terms are rather large compared to the effects expected from anthropogenic climate forcing, for example, and I'm not aware of another real-world case where neural networks would be considered for modelling the effects of such large changes in forcing, so it's not clear to me that these results have real-world applicability. For context, anthropogenic radiative forcing of the climate system is projected to be up to a few percent of solar radiative forcing, and in the scenarios with the largest climate changes, total global warming is around 5x what has been seen in the 20$^{th}$ century, and comparable in size (though not rate) to changes in between ice ages (so we arguably have some data for testing whether models can simulate such large changes well). In the Lorenz '63 experiments here, the change in the sigma parameter (meant to be analogous to radiative forcing of climate?) changes by a factor of 2. In the Lorenz '95 experiments, the forcing change is enough to change the system from being periodic to highly turbulent, which is a much larger qualitative change than expected from climate change. I wouldn't have expected neural networks to perform well at the tasks set, namely simulating systems that are very different from what they've been trained on, so these results don't seem to provide much new information.
      - It seems reasonable to think that neural networks could perform better at simulating the effects of smaller forcing changes, that are more comparable to those in real situations. It would be interesting to test whether the neural networks can reproduce the effects of forcing at the level seen at the end of their training period (relevant for attributing observed weather events to climate change, for example e.g. National Academy of Sciences, 2016, “Attribution of Extreme Weather Events in the Context of Climate Change”) and if so, how far beyond the range of forcing they were trained on can they make good predictions for? (c.f. the Paris climate agreement global warming targets of 1.5C and 2C, which are ~1.5x and ~2x the observed warming – it would be interesting to know if neural networks could provide results that are at all useful for predicting the effects of forcing changes of that magnitude.)
      - As a further comment, it doesn't seem likely that neural networks could learn the effects of forcings outside the range of the training data without having additional information about the effects of larger forcings e.g. the radiative effects of CO2 in the climate change context. So it seems *a priori* likely that for the given setup the performance will deteriorate as the forcing becomes larger. Perhaps the experiments here could demonstrate this, but I don't think it would be that surprising.
   b. The finding that including information about the forcing as an input often worsens performance seems surprising. One reason could be that the network architecture was tuned to optimise performance without the forcing input, and a larger architecture may be needed to perform well with this information. To be a fair test, the network architecture search should be repeated for the networks using forcing as an input – this may be especially relevant for the L63 case, where the network used is quite small. The

statement in the discussion that “it may be better not to include the forcing variable as network input” does not seem well-justified due to this, and also because I do not see how in principle a neural network could predict the effect of a change in forcing if it is not given information about the forcing.

3. It should be made clear in the abstract and conclusions that the results apply for a particular choice of neural network design, namely feedforward networks predicting the system state at time t+1 given the state at time t (it's not clear if this is also the case for the L95 experiments, and this should be clearly stated). Also, the L63 experiments testing whether neural networks could represent the system in one wing of the attractor having been trained on the other wing only used sigmoid activation functions.
   - Predicting the whole state at every time step may be expected to work worse than other designs where only the change in the state is predicted at each step (e.g. Dueben and Bauer, 2018), or where bias-correction of an approximate dynamical model is performed (Watson, 2019, https://doi.org/10.1029/2018MS001597). This is because in these cases, a lot of the variance in the quantity being predicted is removed, so minimising the RMSE in training may work better to give a system that is capturing the important aspects of the variability. These different methods should also be discussed, and the abstract and conclusions should say that the results may not apply to methods like these.
   - The choice of sigmoid activation functions for the L63 network may be relevant for the result that the network will not make predictions outside of the range of its training data because sigmoids saturate, and may have trained to saturate at prediction values that are not far outside the boundaries of the training data region, making it difficult for the neural network to predict values outside this region. It would be good to check what happens when using an activation function that does not saturate e.g. ReLu. (Though I still wouldn't expect it to work well when so much data is left out from training – but I'd still find the result interesting, particularly if done with a system that predicted the tendency rather than the whole state).

**Other comments**

1. p.1 L19 It would seem relevant to include citations to other recent studies using neural networks to simulate the Lorenz 95/6 system (Chattopadhyay et al., 2019, https://doi.org/10.31223/osf.io/fbxns; Watson, 2019, https://doi.org/10.1029/2018MS001597).
2. p.2 L10 - Perhaps also mention Lorenz '95 is sometimes called Lorenz '96
3. p.2 L10-12 – Some context here may be useful. For paleoclimate variability and the oceans over multiple decades, yes, but it's less likely to be the case for the atmosphere that unforced variability would be far outside what we've observed.
4. p.3 L9-10 Training where no large regions of phase space is left out seems to be the most realistic case for atmospheric modelling, which is what is referred to. The experiments may be relevant for e.g. ocean modelling, where time scales are much longer. (I do think they are inherently interesting, as well.)
5. p.3 L25 I'm not sure if all readers would be familiar with the Lorenz butterfly - perhaps refer to a figure.
6. p.3 L26 A better description of the solver is needed e.g. what software package is this from? Reference?
7. p.4 L1 Lorenz95 seems to be more often used to describe the 2-level model Lorenz introduced in the same paper. Perhaps use a different name to make it clear you are considering the 1-level version. (As an aside, the 2-level model could be used to test how well neural networks perform compared to a “truncated” model of the system i.e. the 1-level model – this would address whether neural networks can improve upon other reasonable dynamical models, which is a key question.)

8. p.4 L4 What is the behaviour of the system like with N=40? A plot of a time series or similar may be helpful.
9. p.4 L8-12 More detail seems necessary to make the results reproducible - e.g. Were input variables normalised? Regularisation? Minibatch size? Learning rate? Stopping rule?
10. p.4 L21-22 Why not give F as one value in the L95 system? (Analogous to using CO2 concentration as an input to a climate model.)
11. p.5 L8 – how big are the grid boxes being used?
12. p.5 L8-9 – what does “normalized data” mean here?
13. p.5 L22 For testing 1), the trajectory will tend to the fixed point and only reach it after infinite time, so it's not clear that it could be identified by looking for two points to be exactly equal - perhaps set a threshold on how small a tendency is acceptable and check if the tendency magnitude falls below this and does not rise above it again?
14. p.5 L23-24 What is the motivation behind 2? To identify periodic behaviour? But couldn't points on a periodic orbit fall at slightly different points on that orbit and so avoid detection?
15. p.6 L1-2 The information on training here could perhaps be merged with the earlier section about training.
16. p.6 L5 what does "consecutive forecasts" mean as opposed to just "forecasts"?
17. p.6 L5-6 See earlier comment about the simulated attractor not looking reasonable.
18. p.6 L9 How big are these errors compared to an average tendency magnitude? Also, could errors on the wings be larger because tendencies are typically larger? Perhaps showing a relative error would be more useful.
19. p.6 L14-15 I didn't get the meaning of “the points...included point”.
20. p.8 L11-12 The errors look quite a bit smaller to me, particularly for the case of fig.3f..
21. p.8 L12-14 I don't feel convinced by this analysis. Even if the neural network had learnt to fit the whole attractor well, there may be some neurons whose activations only vary in part of the phase space due to the values of the inputs, and fixing their values would of course be expected to degrade forecasts in areas of the phase space where they do matter.
22. p.11 L5-6 This experiment is not really like rising anthropogenic climate forcing because the sign of the “forcing” depends on whether y is larger than x. An additive forcing term like that used for the L95 experiments and by Palmer (1999, “A nonlinear dynamical perspective on climate prediction”, J. Clim.) would be a better analogy.
23. p.11 L6-7 Changing the number of time steps looks to change the rate of change in the forcing as well. This is something that should be made clear. The results will conflate the effects of changing the amount of training data and the rate of change of the forcing. Perhaps a test could be done for the case with a lower rate of change of forcing with the lower number of time steps used in training to see the effect of changing each aspect of the set up.
24. p.11 L17 It's not clear to me if different training and testing runs are produced for each experimental repeat.
25. p.13 L2 Given that the performance of the Lorenz systems with fixed F also vary between the situations with different training data lengths, could the different rates of forcing change in each case be affecting the system's behaviour in some important way?
26. p.15 L25 - “layers” should be “neurons”?
27. fig.2 - It should be pointed out clearly that the colour scales are different.
28. fig.3 - It's a bit unclear to compare the distributions across panels b and e to see which neurons change behaviour more - it might be better to put the values side by side on the same axes.
29. fig.3 – panel references are wrong in the caption. It also needs to be clearer about what the distributions in b and e are.
30. fig.5 - It needs to be made clear that single-timestep prediction errors are being shown. It would be more interesting to see metrics of longer-range forecast skill and how well the

attractor is simulated, since in the end single-timestep predictions are not the target and may not indicate that well the performance on longer time scales.

31. fig.5a - I suggest using a different colour for the vertical lines.
32. fig.5 - It would be better to write "Ntrain" in exponential notation.
33. fig.5 - The orders of the Lorenz systems could be reversed so they go start to end, the same as for the neural networks.
34. fig.5 - There are no bars on the simulations with the Lorenz system, so have you used the same Truth runs for each experiment, so you could only run the Lorenz systems once, or are the bars just not visible? It would be good to make small bars visible by giving their ends a width so they stick out.
35. References – some references have repeated DOIs.