ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/328908453

Toward Data-Driven Weather and Climate Forecasting: Approximating a Simple General Circulation Model With Deep Learning

Article *in* Geophysical Research Letters · November 2018

DOI: 10.1029/2018GL080704

CITATIONS
43

READS
621

1 author:

Sebastian Scher
Stockholm University

17 PUBLICATIONS **132** CITATIONS

SEE PROFILE

All content following this page was uploaded by Sebastian Scher on 23 January 2019.

The user has requested enhancement of the downloaded file.

AGU100 ADVANCING EARTH AND SPACE SCIENCE

Geophysical Research Letters

RESEARCH LETTER
10.1029/2018GL080704

Key Points:
- A neural network can emulate the dynamics of a simple general circulation model
- The trained network can successfully forecast the model weather
- The network can produce a realistic representation of the model climate

Supporting Information:
- Supporting Information S1
- Figure S1
- Figure S2
- Video S1
- Video S2
- Video S3

Correspondence to:
S. Scher,
sebastian.scher@misu.su.se

Citation:
Scher, S. (2018). Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, *45*, 12,616–12,622. https://doi.org/10.1029/2018GL080704

Received 30 SEP 2018
Accepted 7 NOV 2018
Accepted article online 12 NOV 2018
Published online 23 NOV 2018
Corrected 22 DEC 2018

This article was corrected on 22 DEC 2018. See the end of the full text for details.

©2018. The Authors.
This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

Toward Data-Driven Weather and Climate Forecasting: Approximating a Simple General Circulation Model With Deep Learning

S. Scher[1]

[1] Department of Meteorology and Bolin Centre for Climate Research, Stockholm University, Stockholm, Sweden

Abstract It is shown that it is possible to emulate the dynamics of a simple general circulation model with a deep neural network. After being trained on the model, the network can predict the complete model state several time steps ahead—which conceptually is making weather forecasts in the model world. Additionally, after being initialized with an arbitrary model state, the network can through repeatedly feeding back its predictions into its inputs create a climate run, which has similar climate statistics to the climate of the general circulation model. This network climate run shows no long-term drift, even though no conservation properties were explicitly designed into the network.

Plain Language Summary Numerical weather prediction and climate models are complex computer programs that represent the physics of the atmosphere. They are essential tools for predicting the weather and for studying the Earth's climate. Recently, a lot of progress has been made in machine learning methods. These are data-driven algorithms that learn from existing data. We show that it is possible that such an algorithm *learns* the dynamics of a simple climate model. After being presented with enough data from the climate model, the network can successfully predict the time evolution of the model's state, thus replacing the dynamics of the model. This finding is an important step toward purely *data-driven* weather forecasting—thus weather forecasting without the use of traditional numerical models and also opens up new possibilities for climate modeling.

1. Introduction

Numerical weather prediction (NWP) models and general circulation models (GCMs) are essential tools in weather prediction and climate science. They solve discretized physical equations of the physics of the atmosphere, in order to compute the evolution of atmospheric states over time. Recently, numerous applications using machine learning techniques in connection with GCMs and weather prediction models have been proposed, including learning relations between orbital parameters and climate fields from a climate model (Holden et al., 2015), learning from high-resolution simulations in order to improve predictions made with simpler models (Anderson & Lucas, 2018; Rasp et al., 2018), helping in decision making in extreme weather situations (McGovern et al., 2017), detecting extreme weather in climate data sets (Liu et al., 2016), and predicting the uncertainty of weather forecasts (Scher & Messori, 2018). All these proposed techniques are valuable techniques for climate science and meteorology. However, they all either aim to extract certain information from models, or to *add* information from models to other models.

Here we use deep learning not to extract information from a climate model, or to combine different models, but to directly emulate the complete physics and dynamics of a GCM, generating a neural network that takes as its input the complete model state of the GCM and then predicts the next model state. A lot of progress has been reported on this topic for very idealized models like the Lorenz63 and Lorenz96 models (Lorenz, 1963, 1996) and simple barotropic climate models (e.g., Karunasinghe & Liong, 2006; Krasnopolsky & Fox-Rabinovitz, 2006; Krasnopolsky et al., 2013; Vlachas et al., 2018), and recently also some progress for the prediction of geopotential height in a simplified weather forecasting setting (Dueben & Bauer, 2018). However, to the author's best knowledge, using machine learning methods to map the full state of a GCM to its full state several time steps later—thus learning the complete model dynamics—has not been done before. The aim of this study is to assess whether this is possible with *off-the-shelf* machine learning techniques (thus techniques that are already widely available), namely, with a convolutional neural network (CNN).

10.1029/2018GL080704

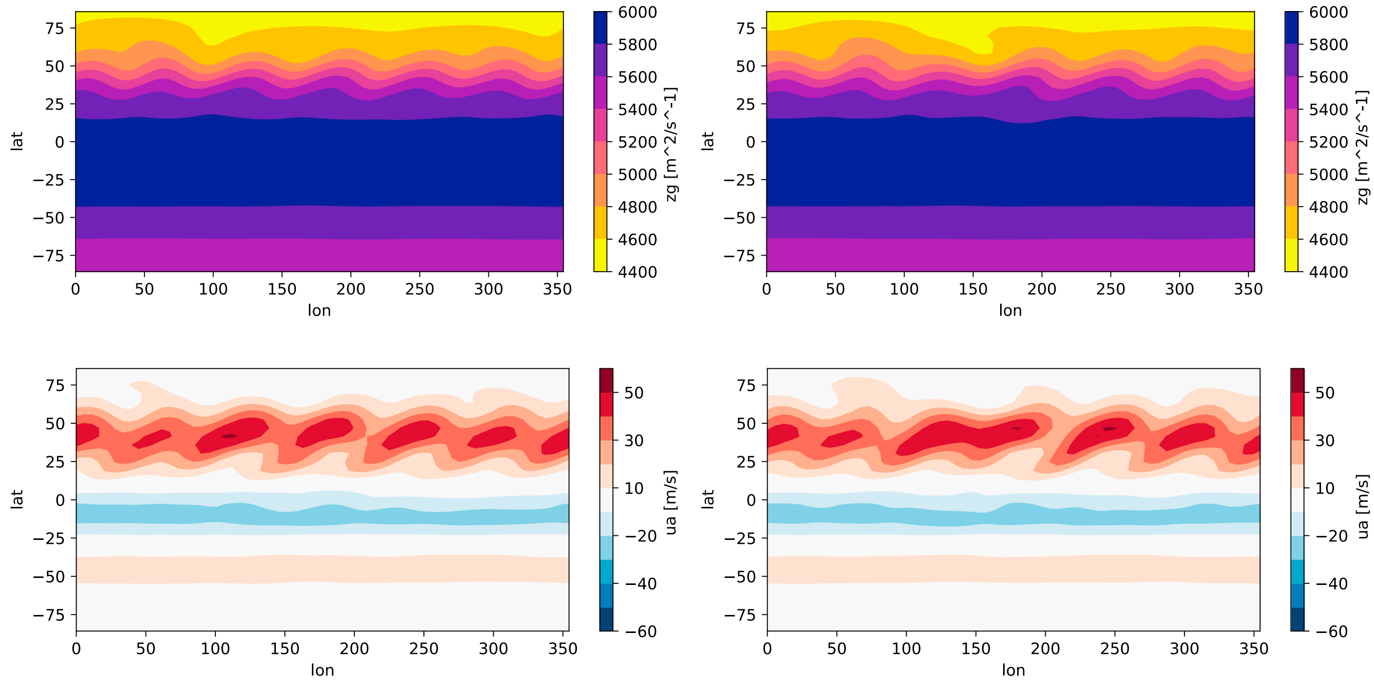

Figure 1. Two model states of the PUMA model, separated by 5 days (from left to right). The upper row shows geopotential at 500 hPa (zg), and the lower row shows zonal wind at 300 hPa (ua).

Naming conventions in this paper: in the machine learning literature, machine learning algorithms (e.g., neural networks) are often referred to as *model*. To avoid confusion with the GCM used in this paper, in the context of this paper, model always refers to climate and weather prediction models, and the machine learning model will only be referred to as *network*.

2. Methods

2.1. PUMA Model

The Portable University Model of the Atmosphere (PUMA) is a simple model of the global atmosphere that is the dynamical core of the PLASIM (Planet Simulator) model (Fraedrich et al., 2005). Based on the primitive equations, it is conceptually very similar to state-of-the-art global NWP models, which are the main tool for medium-range weather forecasts, and to global circulation models which are one of the main tools in climate science. This study is intended as a proof of concept; therefore, PUMA was chosen due to its low computational requirements, which makes it easy to generate long time series of its model climate. Additionally, the relatively low resolution keeps the amount of data in a range that is easy to handle. For the same reasons, the simplest possible model setup was chosen, with no seasonal cycle (eternal Northern Hemispheric winter), no orography, a horizontal resolution of T21($\sim$625 km, 32 $\times$ 64 grid points when projected on a regular latlon grid), 10 vertical levels, no diurnal cycle, no ocean, and a time step of 45 min. Still, the model is more complex and closer to the real atmosphere than, for example, the barotropic model which Vlachas et al. (2018) approximated with a neural network. For the final data, the model's four state variables (horizontal and meridional wind, temperature, and geopotential height) are interpolated from the 10 model levels to 10 pressure levels. Representing model data on pressure level is a standard approach and allows analyzing standard variables like 500-hPa geopotential height, which eases the interpretation of the results. The model state is saved daily. Neural networks are not scale independent; therefore, all variables are normalized to zero mean and unit variance for each layer separately. Figure 1 shows two snapshots of the model integration separated by 5 days. Despite the simplicity of the model, the typical eastward propagating waves are clearly visible in the winter hemisphere (see also Video S1 in the supporting information). Such waves are also observed in the real atmosphere (Rossby, 1939) and are one of the main features of midlatitude weather variability on timescales of several of days.

For machine learning tasks it is standard practice to split up the available data samples into three parts: a training set used for actually training the network, a development set (sometimes also called validation set) used for tuning, and a test set used for final validation. This is necessary because machine learning algorithms

10.1029/2018GL080704

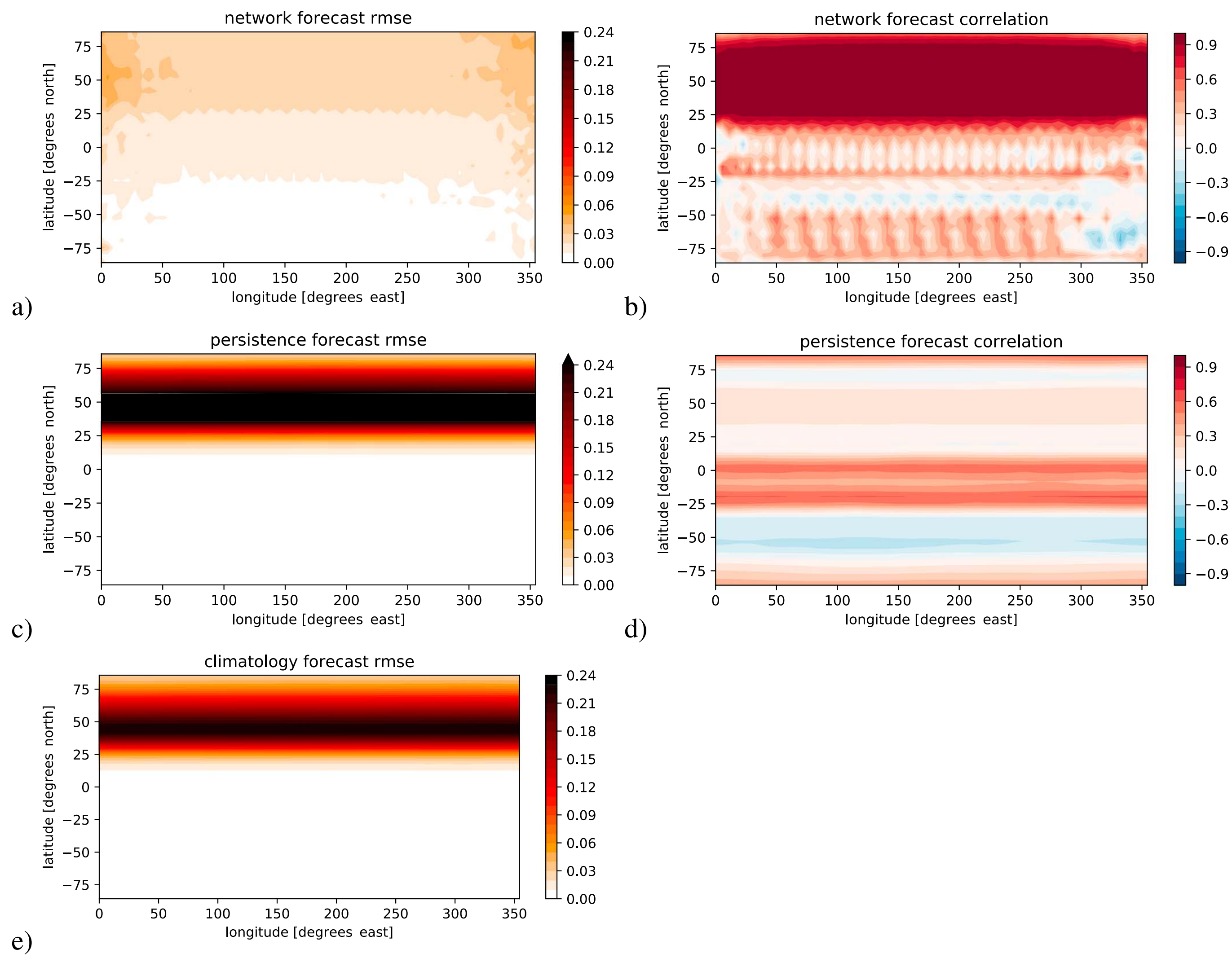

Figure 2. (a) Root-mean-square error (rmse) and (b) correlation for 6-day 500-hPa geopotential height forecasts of the network. (c, d) Same as a and b but for persistence forecasts. (e) Same as a but for climatology forecasts.

can overfit, which means that they can predict samples from the training set very well but fail to predict on unseen data (data that were not present during the training). The necessity for separate development and test sets stems from the fact that even though the development set is not seen in the training, one chooses the architecture that works best on the development set and therefore might end up with a configuration that only works for this particular development set. The PUMA run is split up the following way: The first 30 years are discarded as spin up years, leaving 150 years of daily data. Then the data are split up into three blocks: 20 years for development, 30 years for test, and 100 years for training. For the tuning of the network (see Text S1), the training data were reduced to 30 years for computational efficiency. All results in this study will be reported for the test data.

2.2. Neural Network Architecture

In general, artificial neural networks are able to approximate any nonlinear function (Nielsen, 2015). CNNs are a special type of artificial neural networks that are widely used in image recognition (e.g., Krizhevsky et al., 2012). They use convolutional kernels to process information from one layer to the next, and after each convolution layer there is a nonlinear transformation. For details the reader is referred to the literature (e.g., Nielsen, 2015). It is common to represent atmospheric data on regular grids. Therefore, a single time slice of a weather or climate model can be seen as an image, with individual atmospheric variables on a specific height or pressure level corresponding to individual image channels. This makes it interesting to use CNNs in connection with atmospheric models. A big advantage of CNNs compared to other machine learning methods is that there is

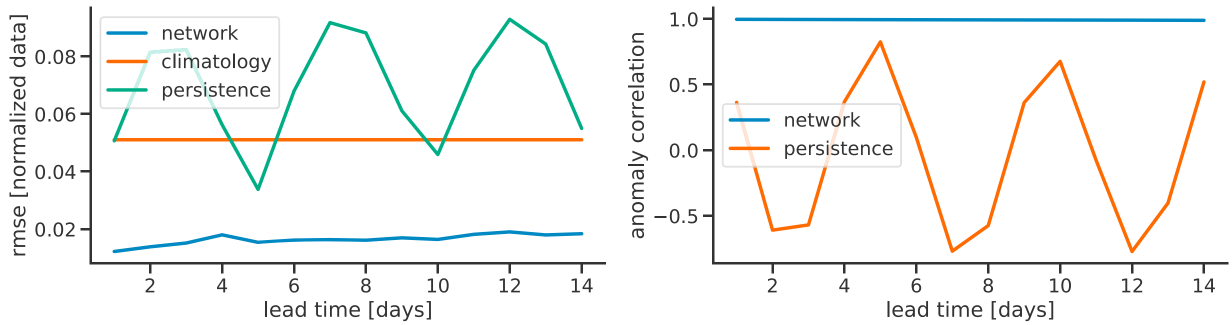

Figure 3. Root-mean-square error and anomaly correlation of the network forecasts and of the baseline methods for different lead times.

no need for prior dimensionality reduction, as in, for example, Vlachas et al. (2018). CNNs have already been used in postprocessing of GCM output in order to detect extreme events (Liu et al., 2016; Racah et al., 2016).

Here we use an architecture that is inspired by so-called autoencoders (Baldi, 2012). An autoencoder is a type of neural network that consists of two parts. In the first part, there is a dimensionality reduction after each layer. In the second part, there is *mirrored* upsampling (increase in dimensionality) after each layer. Most neural network architectures either have low-dimensional input and low-dimensional output, or high-dimensional input and low-dimensional output. The special property of our autoencoder-like architecture is that it has both high-dimensional input and high-dimensional output, or more precisely, the output has exactly the same dimension as the input (40 channels $\times$ 2,048 grid points $= 81,920$). This allows the network to take as input the complete set of atmospheric fields of the GCM at one time step and predict a new complete set of fields at a certain later time. For a weather prediction setting this is not necessarily needed. One could, for example, also aim to predict the weather only at a certain location and thus need only one output variable, for example, the temperature at one specific location. However, aiming to predict the complete model field is more appealing due to two reasons: First, it allows a global weather prediction in one go and thus to emulate the complete model. Second, predicting the complete fields allows the new fields to be fed back into the network again. This can be extended to actually use the network to create a *climate* run, which is started from one initial condition and then allowed to evolve over time. Training of the network takes around 90 min on 20 CPUs. The choice and tuning of the network architecture is described in the supporting information (Text S1), the final architecture is shown in Figure S1.

2.3. Baseline Methods

In order to judge the skill of our network, we compare its predictions to persistence and climatology, which are important baselines in weather forecasting. Persistence entails using the current atmospheric state as forecast, whereas a climatological forecast uses the climatological mean as forecast. The network climate run is compared to the climatology of the climate model.

3. Results

In this section, first, the results of letting the network forecast the climate model several days ahead ("weather" forecasting) are shown. All results are averaged over the whole test set. In the second part, the results from using the network to create a climate run are presented. The analysis focuses on 500-hPa geopotential height, which is a standard validation metric in NWP.

3.1. Weather Forecasting Mode

Here we assess the capability of the trained network in forecasting the model weather (thus forecasting the evolution of the model state up to 14 days ahead). The network was trained for each lead time separately, thus emulating the relationship of the model state between $t = 0$ and $t = 1$ day, $t = 0$ and $t = 2$ days, and so forth. Figures 2a and 2b show maps of the root-mean-square error (rmse) compared to the true state of the GCM for 6-day forecasts of 500-hPa geopotential height. The forecasts have low errors (<0.06, remember that the variables are all normalized to a standard deviation of 1) and very high correlation in the Northern (eternal winter) Hemisphere. The highest error occurs at the edges of the domain. This might be due to the fact that the network is not designed to wrap around the edges, whereas the climate model has a continuous (circular) domain. The low correlation in the Southern (eternal summer) Hemisphere is caused by the very low

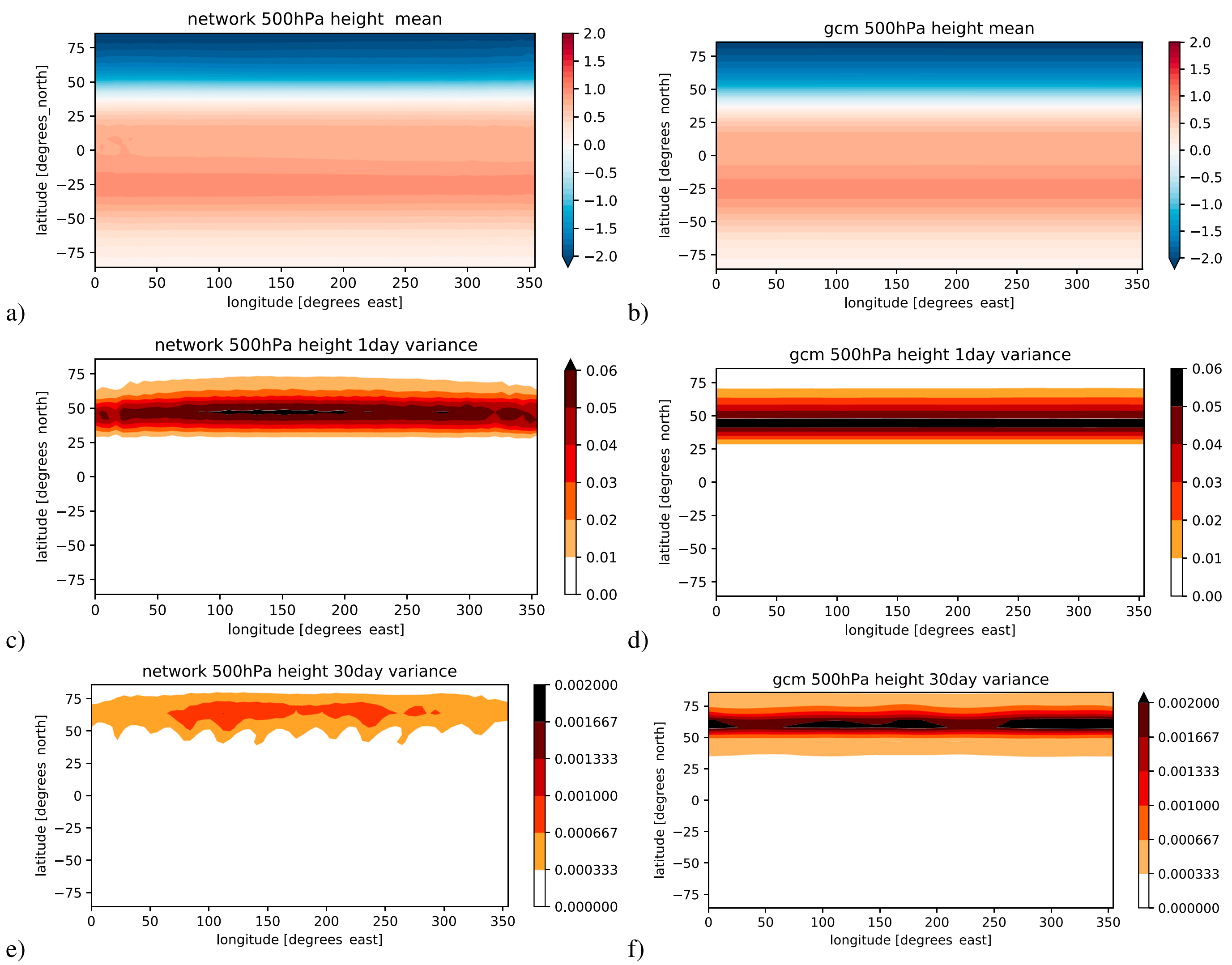

Figure 4. (a, b) Climatology of normalized 500-hPa geopotential height of (a) the network and (b) the general circulation model (GCM). (c, d) Variance of daily mean 500-hPa geopotential height of (c) the network and (d) the GCM. (e, f) Same as c and d but for variance of 30-day means.

variance of the model weather in summer, which renders correlation as metric less useful. The rmse, which is also meaningful in a low-variance weather, is even lower in the summer hemisphere than in the winter hemisphere. The network forecasts are much better than both the persistence forecasts (assuming that the model state stays the same) and the climatological forecasts (using the climatological mean as forecasts). For the persistence forecasts this is the case both for correlation and rmse (Figures 2c and 2d). For climatological forecasts a correlation cannot be computed (because the climatological forecast is one fixed number for each grid point and correlation therefore not defined), the rmse is comparable to the persistence forecasts (panel e).

Figure 3 shows rmse and the anomaly correlation coefficient of network forecasts for lead times 1–14 days (blue line), persistence forecasts (green line), and climatological forecasts (orange line). The network forecasts are far better than the baselines for all lead times. The persistence forecasts are better than the climatology for most lead times. The very high anomaly correlation (∼0.99) and the low rmse show that the network seems to capture the model dynamics very well. The results are similar for other variables and pressure levels (not shown). We can thus conclude that the network is able to correctly forecast the "weather" of the climate model.

3.2. Climate Mode

Now, we use the trained network to create a *climate run*. For this, a 1-day prediction is made from a randomly chosen model state out of the (unseen) test data. Then this prediction is used as starting field for the network, and a new prediction is made, and so on. With this a timeline spanning 30 years of daily fields is created.

10.1029/2018GL080704

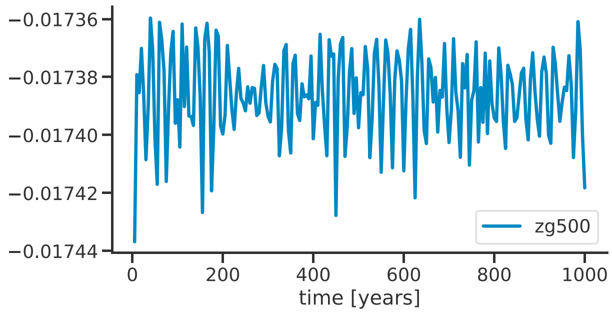

Figure 5. Five-year mean values of 500-hPa height of a 1,000-year network climate run.

Video S1 shows the evolution of the network state over time. As can be seen, even though the network climate seems to show a little less variability than the climate model simulation and some noise, all fields look realistic. Figure 4 shows the climatology of 500-hPa geopotential height for the network climate (panel a). The climatology is both in spatial distribution as well as in magnitude very similar to the climatology of the climate model (panel b). The network also captures the variance of daily and 30-day means quite well (panel c−f), even though it underestimates the 30-day variance. Also, there is too little variance at the edges of the domain. As with the RMSE in the weather forecasting mode, this might be due to the fact that the network is not designed to wrap around the edges, whereas the climate model has a continuous (circular) domain. All results were tested with different randomly chosen initial states of the climate model and are insensitive to the initial state (not shown).

Finally, we make a much longer climate run with the network to assess whether the network shows any drift or spurious long-term variations. This cannot be excluded a priori, as there is no explicit energy conservation or similar prescribed in our network, and in principle nothing prevents it from drifting to completely unrealistic states. Figure 5 shows the evolution of 5-year means of 500-hPa geopotential height over a 1,000-year network climate run (which is significantly longer than most climate simulations). Five-year means were chosen because they still can reasonably be plotted. As can be seen, there is no long-term drift in any of the variables nor in any of the other variables or levels (not shown). These results are insensitive to the choice of initial state (not shown). As with the 30-day variance, the variance of 5-year means shown in Figure 5 is lower in the network than in the model (not shown).

4. Discussion and Conclusion

We have used a relatively standard machine learning technique, namely, a deep CNN, to completely emulate a simple GCM. The network is trained on the GCM and learns to emulate the dynamics of the model. After training, the network can successfully predict the model several time steps ahead—which conceptually is making weather forecasts in the model world. Additionally, we initialized the network with a random initial state from the climate model and then repeatedly fed back the prediction the input of the network. With this, the network can produce a complete climate run, with similar climate statistics as the climate model. This network climate is stable even for very long runs (1,000 years). This is especially encouraging since we have in no explicit way prescribed any form of energy or other conservation properties, and still the network produces a stable climate without drifting away. This is different than what Dueben and Bauer (2018) found for their neural network, which turned out to be unstable after 2 weeks. This might be related to the fact that we trained our network on a simple GCM, whereas Dueben and Bauer (2018) trained on reanalysis data, which is much more complex than our model. Additionally, their network was trained with an hourly time step, compared to the daily time step we used. Finally, another possible cause could be that we used a different neural network architecture (a convolutional autoencoder opposed to dense layers in Dueben & Bauer, 2018).

This study is to be seen as a proof of concept, in which we have shown that in principle it is possible to let a neural network learn the time evolution—and thus the complete dynamics—of a simple GCM. As the very high skill of the neural network in predicting the model weather might partly be caused by the relative simplicity of the model, the next natural step for future studies would be to proceed to more complex and realistic models, including models with external forcing. Correctly representing external forcing is essential when using neural networks for climate studies. The above mentioned will present new challenges due to the more complex dynamics in those models and due to the higher amount of data. The latter is caused both by higher resolution of more complex models and by the fact that probably more training years are needed for a more complex climate/weather system. Still, the principle opens up the possibility of a new type of data-driven weather forecasting. One could train a neural network on long sets of high-resolution climate model simulations (whose resolutions start to approach the resolution of NWP models), for example, the high-resolution (<50 km) simulations from the HighResMip initiative (Haarsma et al., 2016). Then one can feed the analysis of a weather forecast model as predictor into the network and finally use the network to predict the actual weather several days ahead. Additionally, one could try to train the network not on model data but on observations/reanalysis and then run it in climate mode to create a climate simulation.

In terms of neural network development it would also be interesting to attempt developing a special convolutional network architecture that operates on the native model grid (not projected on a regular grid), for example, following the work of Monti et al. (2017). Additionally, one could assess in detail how the dimensionality reduction in the autoencoder-like network used in this study affects the results and also whether adding recurrent elements to the network can lead to improved predictions.

Acknowledgments
I want to thank Gabriele Messori for helpful discussions. Furthermore, I want to thank Stephan Rasp and one anonymous reviewer for their helpful comments. I also want to thank Wilco Hazeleger for interesting duscussions that helped in forming the initial ideas of this study. S. Scher has been funded by the Department of Meteorology of Stockholm University and by Vetenskapsrådet grant 2016-03724. Part of the computations were done on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the High Performance Computing Center North (HPC2N) and National Supercomputer Centre (NSC). The neural network was built with the open-source python libraries Keras (Chollet, 2015) and Tensorflow (Abadi et al., 2015). The PUMA model is available from the University of Hamburg (https://www.mi.uni-hamburg.de/en/arbeitsgruppen/theoretische-meteorologie/modelle/plasim.html). Namelist files for the run used in this study are and the machine learning code is available in the accompanying repository on Zenodo (https://doi.org/10.5281/zenodo.1472023).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems.

Anderson, G. J., & Lucas, D. D. (2018). Machine learning predictions of a multiresolution climate model ensemble. *Geophysical Research Letters*, *45*, 4273–4280. https://doi.org/10.1029/2018GL077049

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning* (pp. 37–49).

Chollet, F. (2015). Keras, GitHub.

Dueben, P. D., & Bauer, P. (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, *11*(10), 3999–4009.

Fraedrich, K., Jansen, H., Kirk, E., & Luksch, U. (2005). Lunkeit, F. *Meteorologische Zeitschrift*, *14*, 299–304.

Haarsma, R. J., Roberts, M. J., Vidale, P. L., Senior, C. A., Bellucci, A., Bao, Q., et al. (2016). High Resolution Model Intercomparison Project (HighResMIP v1.0) for CMIP6. *Geoscientific Model Development*, *9*(11), 4185–4208. https://doi.org/10.5194/gmd-9-4185-2016

Holden, P. B., Edwards, N. R., Garthwaite, P. H., & Wilkinson, R. D. (2015). Emulation and interpretation of high-dimensional climate model outputs. *Journal of Applied Statistics*, *42*(9), 2038–2055. https://doi.org/10.1080/02664763.2015.1016412

Karunasinghe, D. S. K., & Liong, S.-Y. (2006). Chaotic time series prediction with a global model: Artificial neural network. *Journal of Hydrology*, *323*(1), 92–105. https://doi.org/10.1016/j.jhydrol.2005.07.048

Krasnopolsky, V. M., & Fox-Rabinovitz, M. S. (2006). Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, *19*(2), 122–134. https://doi.org/10.1016/j.neunet.2006.01.002

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2013). Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. *Advances in Artificial Neural Systems*, *2013*, 5.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105): Curran Associates Inc.

Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., et al. (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv*, *1605*, 01156.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, *20*(2), 130–141.

Lorenz, E. N. (1996). Predictability: A problem partly solved. Proc. seminar on predictability (Vol. 1).

McGovern, A., Elmore, K. L., Gagne, D. J., Haupt, S. E., Karstens, C. D., Lagerquist, R., et al. (2017). Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, *98*(10), 2073–2090. https://doi.org/10.1175/BAMS-D-16-0123.1

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., & Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model CNNs. Proc. CVPR (Vol. 1, 3 pp.).

Nielsen, M. A. (2015). *Neural networks and deep learning*. Determination Press.

Racah, E., Beckham, C., Maharaj, T., Kahou, S. E., Prabhat, & Pal, C. (2016). ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. arXiv:1612.02095 [cs, stat].

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39) 9684–9689. https://doi.org/10.1073/pnas.1810286115

Rossby, C.-G. (1939). Relation between variations in the intensity of the zonal circulation of the atmosphere and the displacements of the semi-permanent centers of action. *Journal of Marine Research*, *2*, 38–55.

Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*. https://doi.org/10.1002/qj.3410

Vlachas, P. R., Byeon, W., Wan, Z. Y., Sapsis, T. P., & Koumoutsakos, P. (2018). Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A*, *474*(2213), 20170844. https://doi.org/10.1098/rspa.2017.0844

Erratum

In the originally published version of this article, the right panel of Figure 3, as well as Figure S2 in the supporting information, erroneously showed correlation of the absolute fields instead of the anomaly correlation. These errors have since been corrected, and this version may be considered the authoritative version of record.

View publication stats