

Data driven global weather predictions at high resolutions

John Taylor (✉ john.taylor@data61.csiro.au)

CSIRO

Pablo Larraonndo

ANU

Bronis de Supinski

LLNL

Article

Keywords: Numerical Weather Prediction, Data Driven Models, Precipitation Fields, Operational Weather Forecasting

Posted Date: March 9th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-310930/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at The International Journal of High Performance Computing Applications on August 18th, 2021. See the published version at <https://doi.org/10.1177/10943420211039818>.

Data driven global weather predictions at high resolutions

John A. Taylor^{1,2,3*}, Pablo Larraondo⁴ and Bronis R. de Supinski⁵

March 2021

¹CSIRO Data61, Canberra, Australian Capital Territory, Australia

²College of Engineering and Computer Science, The Australian National University, Canberra, Australia

³Defence Science and Technology Group, Department of Defence, Canberra Australia

⁴Fenner School of Environment and Society, The Australian National University, Canberra, Australia

⁵Livermore Computing, Lawrence Livermore National Laboratory, Livermore, California, USA

*Corresponding Author Address: John Ashley Taylor, CSIRO Data61, CSIRO Synergy Building, Clunies Ross Street, Black Mountain, Canberra ACT 2601.
E-mail: John.Taylor@data61.csiro.au

Abstract

Society has benefited enormously from the continuous advancement in numerical weather prediction that has occurred over many decades driven by a combination of outstanding scientific, computational and technological breakthroughs. Here we demonstrate that data driven methods are now positioned to contribute to the next wave of major advances in atmospheric science. We show that data driven models can predict important meteorological quantities of interest to society such as global high resolution precipitation fields (0.25°) and can deliver accurate forecasts of the future state of the atmosphere without prior knowledge of the laws of physics and chemistry. We also show how these data driven methods can be scaled to run on super-computers with up to 1024 modern graphics processing units (GPU) and beyond resulting in rapid training of data driven models, thus supporting a cycle of rapid research and innovation. Taken together, these two results illustrate the significant potential of data driven methods to advance atmospheric science and operational weather forecasting.

Main

Improvements in our understanding of the physical and chemical processes that govern the state of the atmosphere have combined with rapid advances in computational science to pave the way to dramatically improved understanding of weather and climate over the past few decades (Bauer, Thorpe, and Brunet 2015). Predicting the evolution of the weather using the laws of physics and chemistry, a deterministic approach, is achieved by integrating a known set of partial and ordinary differential equations using a set of initial conditions that are based on observations. Significant improvements in weather forecasting using a deterministic approach were achieved not only through better data assimilation and modelling systems, but also from massive increases in the amount of observational data, particularly meteorological data derived from satellites (Simmons and Hollingsworth 2002).

Not all significant physical processes, particularly those operating at small scales, can be explicitly resolved by weather and climate models and instead are parameterised (Bauer, Thorpe, and Brunet 2015). The chaotic non-linear nature of weather forecasting also presents fundamental limits to its predictive skill. To better address these uncertainties, ensemble weather forecasting approaches have been developed that produce probabilistic forecasts, with the computational cost proportional to the number of members in the ensemble (Zhang and Pu 2010).

More recently, machine-learning technology driven by the availability of massive data sets and graphics processing units (GPUs) has produced substantial

breakthroughs in our ability to analyse images, video and audio streams, text and speech that have enjoyed widespread adoption in both commercial and research settings (Lecun, Bengio, and Hinton 2015). Deep learning, a sub-category of machine learning, has played a leading role in these developments. Deep learning constructs a representation of input data through a series of connected non-linear layers. These networks are able to produce the desired output by optimising a set of adjustable parameters, or weights, that minimise a desired loss function. The optimisation, or training, process adjusts the weights using a form of gradient descent (Lecun, Bengio, and Hinton 2015). This article explores the application of deep learning methods to weather and climate problems.

An important feature of multi-layer deep neural networks (DNN) is their ability to represent arbitrarily complex functions or a combination of functions that map from one finite dimensional space to another, referred to as the universal approximation theorem (Hornik, Stinchcombe, and White 1989). The sweeping range of successful applications of deep learning models provides further confirmation that DNNs are a class of universal approximators (Lecun, Bengio, and Hinton 2015). Where a deterministic relationship exists between the input and the target, DNNs can potentially complement our understanding of any physical system. Particularly, where the complexity of the physical system is high, parameters are poorly constrained, and/or the required set of initial conditions are not complete.

While the universal approximation theorem provides confidence that we can apply DNNs to complex physical systems, it does not provide guidance as to the number of layers in the DNN and the number of parameters or weights required to approximate an unknown function to an acceptable degree of accuracy (Hornik, Stinchcombe, and White 1989). A DNN model may be under-fit, where we have a DNN of insufficient complexity, or, in the more challenging problem, over-fit, where we have a DNN with too many parameters so it learns just the features included in the training dataset without generalising to the underlying problem. To avoid over-fitting, practitioners commonly match the increasing capacity of models with larger datasets, providing a regularisation effect that leads to better generalisation. This requirement for large quantities of data leads to the characterisation of many machine learning approaches as big-data driven methods. The results presented below address this need for large quantities of high quality training data.

The development of convolutional neural networks (CNN) produced a dramatic drop in error rates for image classification tasks and resulted in their rapid adoption (Krizhevsky, Sutskever, and Hinton 2012). An important property of CNNs is their efficiency when compared with fully-connected neural networks, so they can be applied to multi-dimensional data and can scale to large image sizes. They have seen widespread application to 2-D image and video data in many scientific domains (Liu, Ouyang, Wang, Fieguth, Chen, Liu, and Pietikäinen 2020). In

order to address the problem of modelling data sets that include a time dimension, recurrent neural networks (RNNs) map the current time step from past time steps (Lecun, Bengio, and Hinton 2015). A crucial advance in the practical application of RNNs was the development of long short-term memory (LSTM) RNNs (Hochreiter and Schmidhuber 1997), which can capture the long-range dependencies in the time series data while avoiding the problem of numerically unstable backpropagated gradients. The combination of CNNs and with an LSTM allows us to model complex multi-dimensional time series data and is likely to enjoy widespread application to scientific problems, including weather forecasting and climate modelling (Shi, Chen, Wang, Yeung, Wong, and Woo 2015).

Machine learning techniques have been successfully applied to solve problems such as weather forecasting (Zhang and Pu 2010), (Wang, Luo, Lu, Li, Zhang, Yan, and Zheng 2019), estimating weather forecasting uncertainty (Scher and Messori 2018) post-processing ensemble weather forecasts to surface stations (Rasp and Lerch 2018), climate analytics on massive climate modelling data sets (Kurth, Treichler, Romero, Mudigonda, Luehr, Phillips, Mahesh, Matheson, Deslippe, Fatica, Prabhat, and Houston 2019) and identifying synoptic-scale fronts (Lagerquist, McGovern, and Gagne 2019). Applying machine learning to precipitation estimation has attracted significant interest including its application to downscaling (Pan, Hsu, AghaKouchak, and Sorooshian 2019), predicting precipitation from weather radar data at high spatial and temporal resolutions (Aditya Sai Srinivas, Somula, Govinda, Saxena, and Pramod Reddy 2020), and combining continuous and categorical binary indices for precipitation forecasting (Larraondo, Renzullo, Van Dijk, Inza, and Lozano 2020).

With the early successes of applying machine learning techniques to aspects of climate modelling and weather forecasting, attempts are now being made to develop elementary weather prediction models using DNNs (Scher and Messori 2019). Recent results deliver predictions that outperform simple persistence, climatology and early deterministic weather forecast models (Weyn, Durran, and Caruana 2019). DNNs have also been found to outperform numerical weather prediction (NWP) models under specific circumstances (Sønderby, Espeholt, Heek, Dehghani, Oliver, Salimans, Agrawal, Hickey, and Kalchbrenner 2020). DNNs have been applied to build low resolution global scale models that predict the 500-hPa geopotential height using about 7 years of atmospheric data as the training data set in order to better understand the challenges that we will face in achieving the goal of building weather forecasting models using DNNs of comparable complexity to existing deterministic models (Dueben and Bauer 2018). In order to accelerate the development of data-driven methods for weather forecasting, benchmark data sets and recommended evaluation metrics have recently been released that are intended to facilitate rapid and consistent model inter-comparisons (Rasp, Dueben,

Scher, Weyn, Mouatadid, and Thuerey 2020) and proposals for the development of a novel scalable infrastructure for weather and climate modelling have been proposed (Bauer, Dueben, Hoefler, Quintino, Schulthess, and Wedi 2021).

Due to the large size and complexity of weather and climate datasets, exploration of new deep learning methodologies remains a challenge. In this work, we present a framework for scaling up DNN models and for accelerating the process of discovering new DNN based approaches to model the weather and climate.

Results

In this paper we focus on the following five key features of developing a data-driven approach to weather forecasting and climate modelling. (1) Our focus is at the global scale. We use global high resolution (0.25°) weather fields extending over multiple decades at an hourly temporal resolution as our training data. (2) As a first step, we demonstrate that a DNN is capable of learning the complex physical relationship between geopotential height and precipitation at high spatial and temporal resolutions comparable to modern global weather forecasting systems. (3) We investigate scaling of the DNN model by training the model using thousands of GPU devices, a necessary requirement when working with terabytes of training data. (4) We then extend the DNN model to the time dimension and make predictions of geopotential height. We chose geopotential height as it has long been recognised as a key meteorological forecast field. (5) We suggest a pathway for developing data driven models of comparable complexity to modern numerical weather prediction models.

Predicting global high resolution precipitation using a DNN

Our first experiment focusses on using a DNN to successfully perform a 2-dimensional spatial regression using geopotential height data to predict surface precipitation. While the dataset includes a temporal domain, in these initial experiments we consider each sample to be independent, so we refer to this forecast problem as a nowcast. We test the limits of what is possible by using meteorological data with global coverage and high spatial and temporal resolution in order to determine if data driven models can operate at comparable scales to modern weather forecasting and climate models. In this case by predicting the complex surface total precipitation field using geopotential height fields at 500, 800 and 1000 hPa levels. Figure 1a presents a schematic representation of the data set that we have utilised for model training and validation. Figure 1b provides a summary of the DNN we have trained to predict surface total precipitation fields. The model shown in Figure 1b was implemented in TensorFlow 2.2 (Abadi, Agarwal, Barham, Brevdo,

Chen, Citro, Corrado, Davis, Dean, Devin, Ghemawat, Goodfellow, Harp, Irving, Isard, Jozefowicz, Jia, Kaiser, Kudlur, Levenberg, Mané, Schuster, Monga, Moore, Murray, Olah, Shlens, Steiner, Sutskever, Talwar, Tucker, Vanhoucke, Vasudevan, Viégas, Vinyals, Warden, Wattenberg, Wicke, Yu, and Zheng 2015).

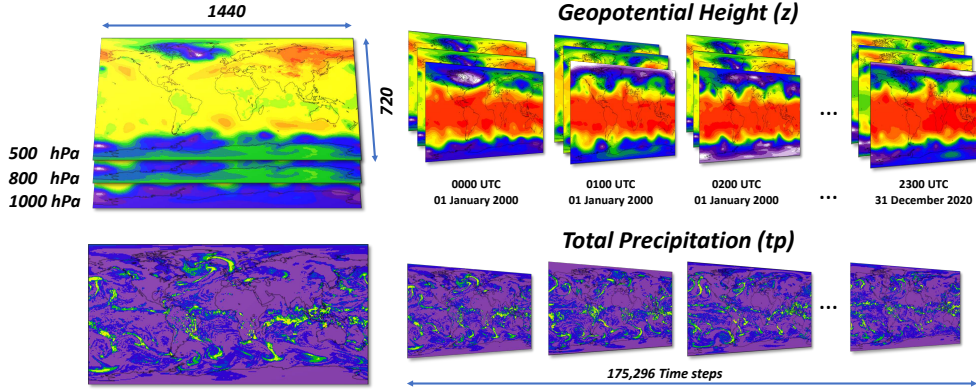


Figure 1a: Presents a schematic representation of the data set that we have utilised for model training and validation. The data for model training is selected from the ERA5 data set and provides hourly estimates of variables with global coverage at a spatial resolution of 0.25 degrees \approx 30km.

We trained the model shown in Figure 1a and 1b using 30 years of data over 50 epochs using a RELU activation function and the root mean square error (RMSE) as the loss function. The DNN model was implemented in TensorFlow 2.2 (Abadi, Agarwal, Barham, Brevdo, Chen, Citro, Corrado, Davis, Dean, Devin, Ghemawat, Goodfellow, Harp, Irving, Isard, Jozefowicz, Jia, Kaiser, Kudlur, Levenberg, Mané, Schuster, Monga, Moore, Murray, Olah, Shlens, Steiner, Sutskever, Talwar, Tucker, Vanhoucke, Vasudevan, Viégas, Vinyals, Warden, Wattenberg, Wicke, Yu, and Zheng 2015), a programming interface specifically designed for implementing machine learning algorithms. Importantly, TensorFlow natively runs on GPU computing devices. We also use the Horovod deep learning framework (Sergeev and Balso 2018), which takes advantage of decades of research on parallel computing by the HPC community to facilitate distributed deep learning across multiple GPU devices on multiple nodes.

The model was run on 80 Nvidia V100 processors each with 32 Gb memory on the Gadi Supercomputer at the National Computational Infrastructure. On 80 GPUs a single epoch consisting of 261,800 time steps was completed in 287s and

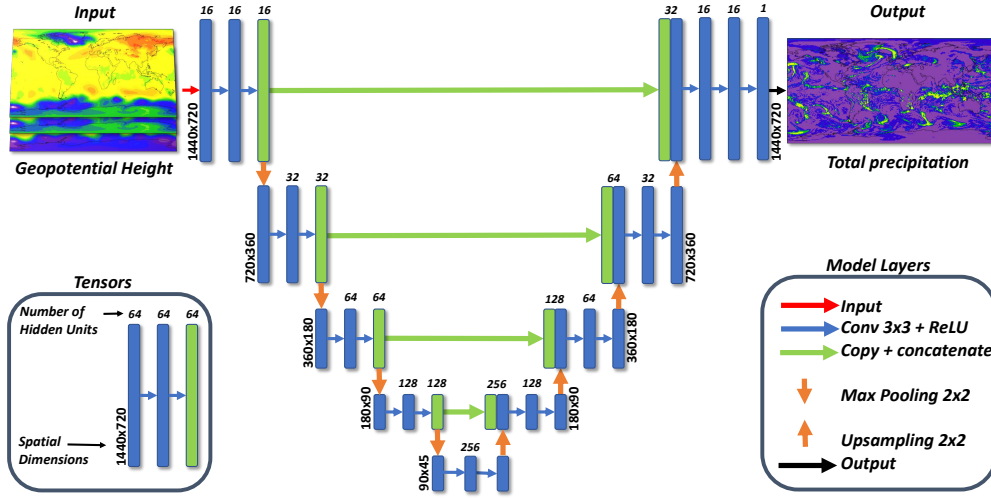


Figure 1b: Provides a summary of the architecture of the DNN we have trained to predict surface total precipitation fields. The model is based on the U-net DNN model. The model shown was implemented in python using the TensorFlow 2.2 API.

produced a MSE value of 0.0791 on the validation data set. Figure 2 presents an example set of results obtained from training the DNN. Figure 2 shows a plot of the ERA5 total precipitation field which we wish to estimate, the input geopotential height field at two of the three pressure levels used to predict the surface precipitation and the resulting estimates of surface total precipitation. While the model training requires significant compute resources the inference step, where we predict the global high resolution surface precipitation, requires only a fraction of a second of compute time. The DNN model is clearly able to model the spatial variability of key precipitation features both at mid-latitudes and in the tropics, though to a lesser extent in the tropics, indicating that geopotential height at multiple levels provides important information that determines the spatial distribution of surface total precipitation. Further improvements, such as to the prediction of precipitation in the tropics, could be achieved by employing additional data fields in addition to the geopotential height data used in this study.

Scaling a DNN model on GPU devices

The combination of TensorFlow and Horovod allows DNN models to be scaled to thousands of GPU devices providing access to a level of scalability and a concomitant compute capability likely greater than what is currently available to

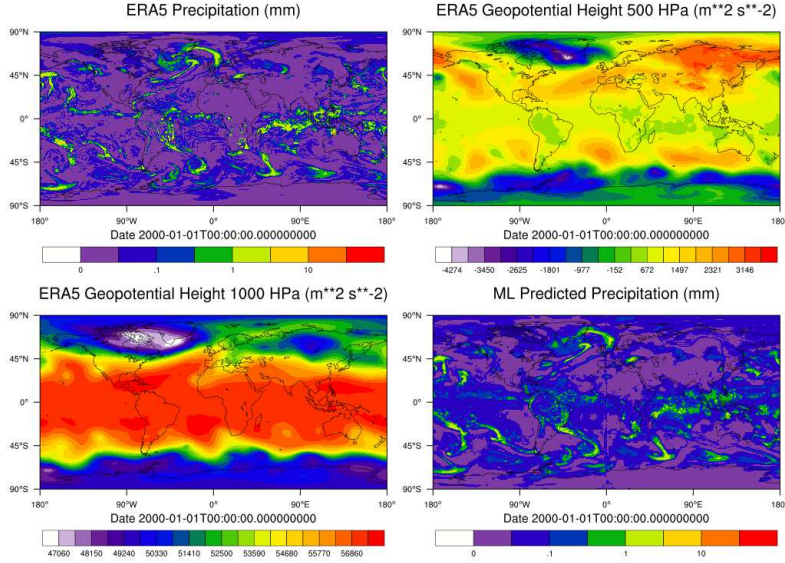


Figure 2: Prediction of surface total precipitation using the trained DNN model. a. The ERA5 total precipitation field which we wish to estimate (mm); the plot scale is non-linear in order to reveal the variation in precipitation over several orders of magnitude. b. The input geopotential height field at 500 hPa pressure level ($\text{m}^2 \text{s}^{-2}$) c. The input geopotential height field at 1000 hPa pressure level ($\text{m}^2 \text{s}^{-2}$) d. DNN estimates of surface total precipitation (mm) corresponding to 2a; as in 2a., the plot scale is non-linear in order to reveal the variation in predicted precipitation over several orders of magnitude.

run current weather and climate models (Kurth, Treichler, Romero, Mudigonda, Luehr, Phillips, Mahesh, Matheson, Deslippe, Fatica, Prabhat, and Houston 2019). We investigate the scalability of our DNN model on GPU devices on the Lassen supercomputer at Lawrence Livermore National Laboratory. Figure 3a presents the results of a scaling DNN training using 4-256 Nvidia V100 GPUs with 16GB memory using 2 years of training data. Figure 3b presents the results of a scaling DNN training using 128-1024 Nvidia V100 GPUs with 16GB memory. We use 10 years of input data, a total of 87602 time steps, to train the DNN model, noting that the performance gains (scalability) shown in Figure 3 are not dependent on the number of years of input data used for training. We find that training times increase linearly with increasing years of input data, for example using 128 GPUs with 2 years of training data we observed a training time per epoch of 11s as shown in Figure 3a and using 10 years of data we obtain a per epoch training time of 55s as shown in Figure 3b. The results in Figure 3 illustrate that a high degree of scalability can be achieved up to 1024 GPUs when training our DNN, with the potential that good scalability beyond 1024 GPUs was possible. We also investigate scalability on the Gadi supercomputer using 16, 32 and 64 V100 GPUs with 32 GB memory using 1 year of input data for training. We observed a per-epoch training times of 52, 26, and 13 seconds respectively, again consistent with a very high degree of scalability of training for our DNN using a combination of TensorFlow and Horovod.

Forecasting global high resolution 500 hPa geopotential height using a DNN

Having successfully applied our DNN model, as described in Figure 1, to predict the spatial distribution of total precipitation, we extended our DNN model to include the temporal dimension. We achieved this by adding an LSTM (Hochreiter and Schmidhuber 1997), in combination with the 2-D convolutional capabilities, making it possible to forecast the time series evolution of 2-D meteorological variables. In Figure 1b we replaced all the layers labelled 'Conv' with a 'Conv2DLSTM' layer, otherwise the DNN architecture remains the same. This change, however, produced a more than ten fold increase in the number of parameters in the DNN model as a result of including additional parameters required by the LSTM to capture the time series evolution of complex global high resolution 2-D meteorological fields.

With our forecast DNN model we focus on the prediction of 500 hPa geopotential height using only past observations of 500 hPa geopotential height as input. We chose the 500 hPa geopotential height as prior studies using DNNs at much lower spatial resolutions has successfully forecast 500 hPa geopotential height (Dueben

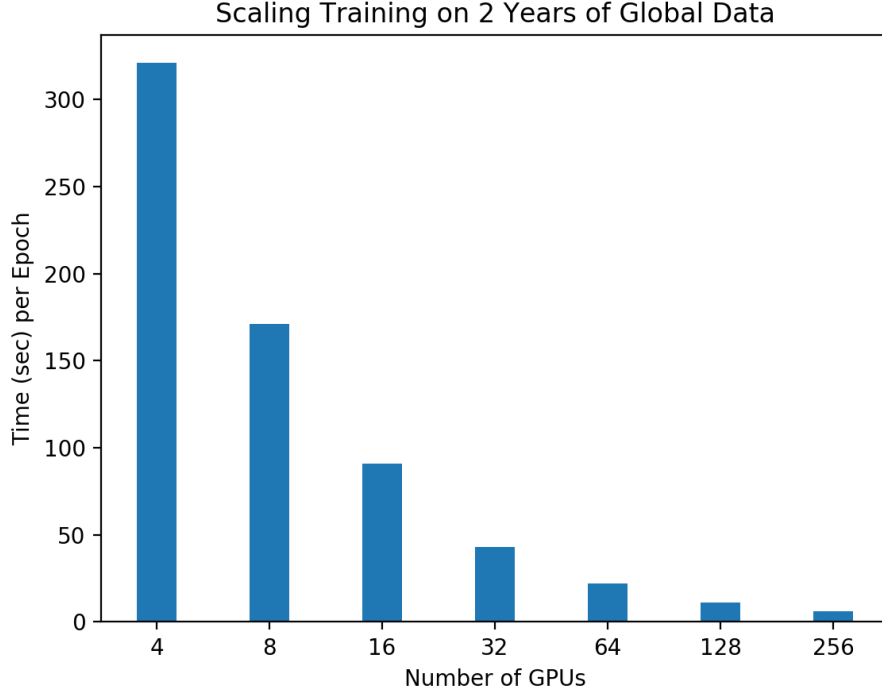


Figure 3a: Scaling DNN training on multiple GPU devices. We scale DNN training on 4-256 Nvidia V100 GPUs with 16GB memory running on the Lassen supercomputer at the Lawrence Livermore National Laboratory, USA. We used 2 years of input data 1980-81, consisting of 17521 time steps, to train the DNN model at each GPU count. We report the time to complete training on one epoch (s) as reported by TensorFlow, representing a full pass of the model parameter optimisation process over the entire 2 year training data set.

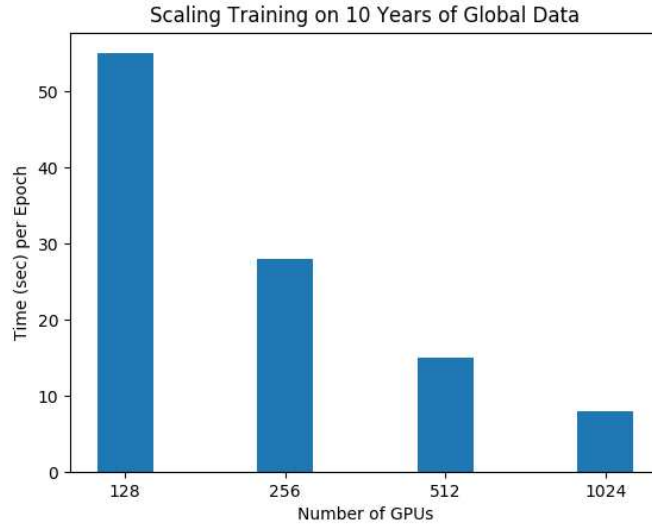


Figure 3b: Scaling DNN training on multiple GPU devices. We scale DNN training on 128-1024 Nvidia V100 GPUs with 16GB memory running on the Lassen supercomputer at the Lawrence Livermore National Laboratory, USA. We used 10 years of input data 1980-89, consisting of 87602 time steps, to train the DNN model at each GPU count. We report the time to complete training on one epoch (s) as reported by TensorFlow, representing a full pass of the model parameter optimisation process over the entire 10 year training data set.

and Bauer 2018), (Weyn, Durran, and Caruana 2019). We use input data for the forecast DNN consisting of 10 prior hourly time steps and output a forecast consisting of 2 hourly time steps into the future. We then use a recursive approach to forecast up to 24 hours ahead in 1 hour increments. The selection of the number of input and output time steps was limited by the availability of GPU memory. Both the increase in the number of model parameters required by the LSTM and in the number of time steps to 12 (10 past + 2 future), results in a large increase in GPU memory requirements.

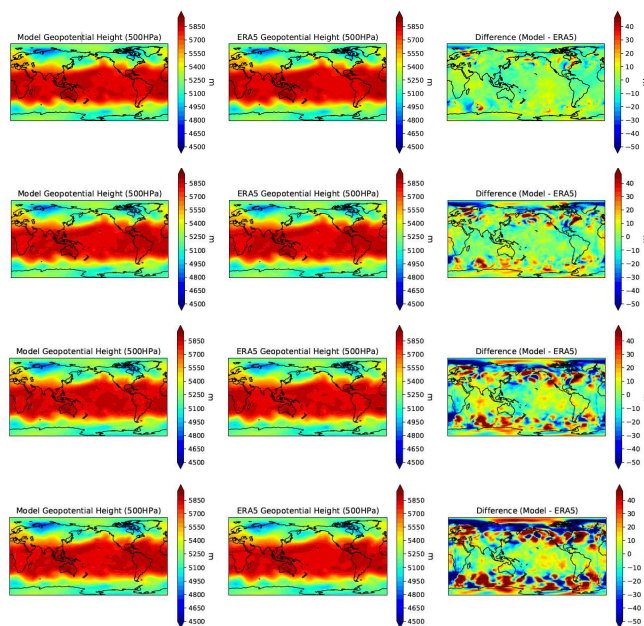


Figure 4a: An example time series prediction of 500 hPa geopotential height at lead times $t=1, 6, 12, 18$ hours (m). Each panel includes the actual geopotential height, the corresponding forecast DNN model estimates of geopotential height and the difference between these two plots. At mid-latitudes, the location of the greatest spatial and temporal variation in geopotential height, we observe the greatest prediction errors.

Figure 4a provides example time series predictions of 500 hPa geopotential height at $t=1, 6, 12, 18$ hours into the future. Figure 4b provides a comparison of the forecast error against persistence and a weekly climatology using a range of 2-10 input time steps. We find that the forecast DNN model is able to deliver results that are significantly better than these simple forecast metrics, indicating the potential of this approach to provide predictions of global scale high resolution meteorological parameters comparable to modern NWP models. Figure 4b

demonstrates the importance of the number of past time steps to the successful prediction of future 500 hPa geopotential height with the predictions improving as we increase in the number of input time steps. Using 4 input time steps results in the transition from an exponential increase in RMSE to a linear increase in RSME with increasing forecast time. As we increase the number of input data points we observe that the slope of the RMSE curve decreases and the forecast improves particularly at the longer forecast times. Figure 4b also indicates that increasing the number of prior time steps beyond 10 is likely to continue to improve the forecast accuracy.

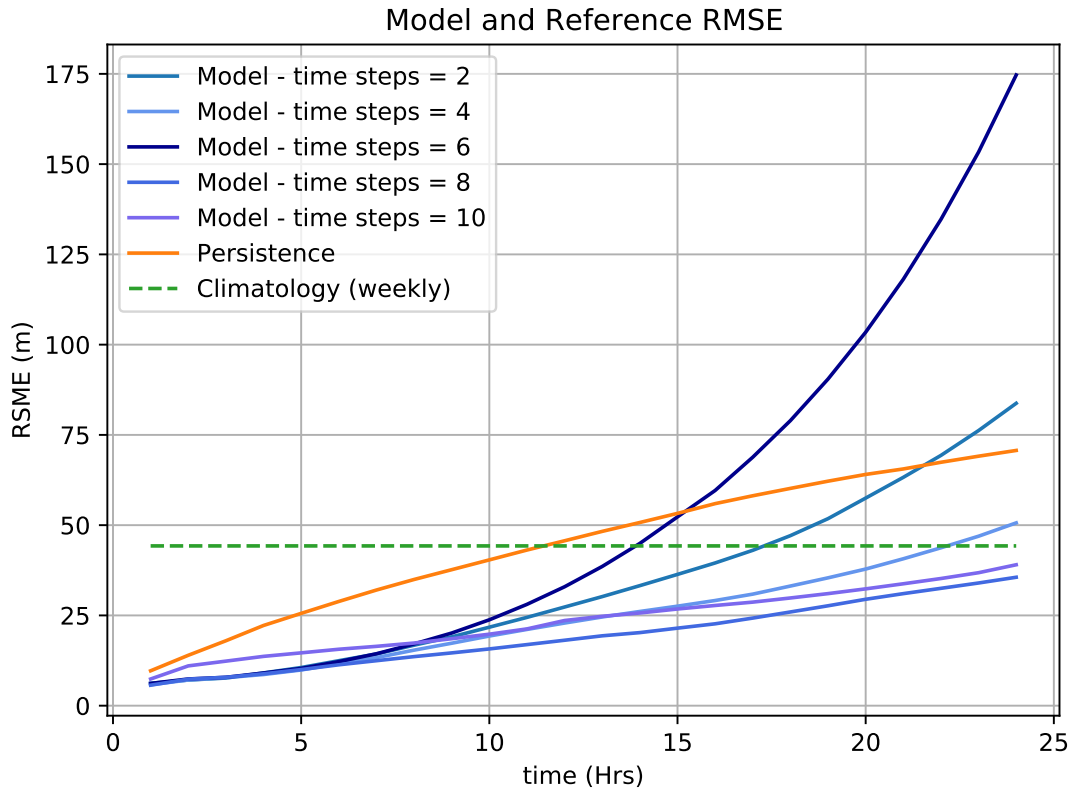


Figure 4b: A comparison of the forecast error as RMSE (m) using a forecast DNN models trained with a range of input time steps (2-10) compared with persistence and a weekly climatology.

Discussion

The results above demonstrate that we can make high resolution global scale predictions of meteorological variables, such as total precipitation, at a scale and time frequency comparable to current generation NWP models. While we have been successful with our current DNN model, further gains in the DNN model performance are possible. For example, the DNN model can be improved by adding additional input variables or modifying how the variables are presented to the DNN, a process referred to as feature selection by the machine learning community. Additional input variables would improve the model fit only if they provide additional information about the processes producing total precipitation. By selectively adding and removing additional input variables we are able to explore the importance of each variable and the role that it plays in the prediction of the desired output. Feature selection can also include data transformations that improve the process of numerical optimisation of DNN parameters, such as data normalisation, and can be just as critical to successfully constructing a DNN model as selecting the best set of input parameters.

The DNN model can also be modified by changing the architecture of the model and by modifying the many hyper-parameters related to the model fitting process. We have based our DNN model on the Unet model architecture and have clearly demonstrated that this model architecture can deliver reliable predictions of meteorological quantities with high spatial variability. We were then able to modify the UNET model to enable weather prediction at high spatial and temporal resolutions by combining the Unet model with the LSTM, Unet-LSTM. The combined Unet-LSTM approach has delivered excellent results in our weather forecasting application and this architecture is likely to be relevant to many other applications that seek to predict the spatial-temporal evolution of physical quantities. Our current results provide a useful benchmark against which other high resolution global DNN models can be compared. Increasing the number of past time steps used as input to the LSTM well beyond the 10 used in this study will likely lead to improved model predictions, and this is currently only limited by available GPU memory.

While the results presented in this study are very promising, the DNN model developed in this study has a very limited focus compared to major weather forecasting systems, which incorporate dozens of high resolution input and output fields and forecast models with millions of lines of code. However, the ongoing development of modern programming interfaces written in Python for developing DNN models, such as TensorFlow (Abadi, Agarwal, Barham, Brevdo, Chen, Citro, Corrado, Davis, Dean, Devin, Ghemawat, Goodfellow, Harp, Irving, Isard, Jozefowicz, Jia, Kaiser, Kudlur, Levenberg, Mané, Schuster, Monga, Moore, Murray, Olah, Shlens, Steiner, Sutskever, Talwar, Tucker, Vanhoucke, Vasudevan, Viégas,

Vinyals, Warden, Wattenberg, Wicke, Yu, and Zheng 2015), which now incorporates the high level Keras API (Chollet 2015) that runs on top of TensorFlow, will allow the rapid development of data driven models of similar complexity to NWP models. When we construct a DNN model we define a set of inputs, model layers and outputs, which limits our ability to build large complex models as the resulting model would then be monolithic. Fortunately, the Keras interface includes a functional API that can handle multiple inputs and outputs and can treat any model as yet another layer by invoking it as an input or an output of another layer, which allows both the model architecture and the weights to be reused. Models can be nested and ensembles of models can easily be defined using this approach. The Keras functional API therefore provides an example of a pathway for efficiently developing DNN models with complex graph topologies of similar complexity to modern NWP models.

In addition to the challenges posed by developing complex DNN models, we also need to be able to train these models in a reasonable timeframe. In our study we have demonstrated how we can scale our DNN models using Horovod (Sergeev and Balso 2018) to run on up to 1024 GPUs with significant potential for scaling beyond this number of GPUs, which has allowed us to take full advantage of the more than billion fold improvement in computing power achieved on HPC systems over the past three decades. Again, while these results are very promising we have encountered limitations in the available GPU technology. As we increase model complexity and the number of input and output variables, particularly at high resolutions demanded by weather forecasting models, we overwhelm the available GPU memory. GPU device memory is steadily increasing however developing complex DNN models will likely require GPU devices with hundreds of GBs to TBs of memory, similar to what is available in current CPU servers. Other developments addressing the issue of GPU memory limitations include combining both data-parallel approaches with model-parallel approaches (Awan, Jain, Anthony, Subramoni, and Panda 2020), (Van Essen, Kim, Pearce, Boakye, and Chen 2015).

Building complex DNN models will also pose a significant challenge related to the amounts of data required for training. A DNN model similar in complexity to a NWP model will probably use PBs of data during training. Moving this data from storage through to the GPU efficiently will be essential, as model training requires that all training data is read during multiple training cycles or epochs. In our study we took advantage of the large amounts of CPU memory available on the GPU nodes (256 GB) by staging the data to CPU memory. The total CPU memory available on Lassen when using 1024 GPUs on 256 GPU nodes was 65TB. Given our experience in training DNN models, training complex DNN models will clearly benefit from co-design of future HPC systems to optimise that training.

Based on the many previous studies and the results presented here, data driven

methods are clearly now positioned to contribute to the next wave of major breakthroughs in atmospheric science. However, it is important to note that achieving this goal will require a substantial effort, comparable to the investment currently being made in NWP development, and that data-driven weather prediction (DWP) is intended to be complimentary to existing numerical weather forecasting approaches.

Methods

Dataset

The ERA5 data set (Neumann, Düben, Adamidis, Bauer, Brück, Kornbluh, Klocke, Stevens, Wedi, and Biercamp 2019) provides hourly estimates, currently commencing in 1979, of many atmospheric, land and oceanic variables at global scale with a spatial resolution of 0.5 degrees, ~ 30 km. Atmospheric variables are computed on 137 levels to a height of 80 km. ERA5 dataset was created by combining a comprehensive set of historical meteorological observations with a sophisticated data assimilation and modelling workflow developed by ECMWF. ERA5 data from ECMWF can be obtained on request from ECMWF’s meteorological data archive and retrieval system (MARS).

Our experiments use geopotential height (z) at the pressure levels [1000,800,500] hPa and total precipitation (tp) variables. We use the full global data set with latitude and longitude dimensions of [720,1440]. The temporal domain data span from the Year 2000 up to the end of 2019, with a temporal resolution of 1 hr. The resulting geopotential height data are represented as a four-dimensional numerical array with shape [175296, 720, 1440 and 3] corresponding to dimensions [time, latitude, longitude, and height], a total of approximately 2.18 TBytes. Similarly, the total precipitation is represented by a three-dimensional numerical array with shape [175296, 720, 1440] representing the [time, latitude, and longitude] dimensions, an additional 727 GBytes.

The forecasting experiments use only geopotential height at 500 hPa arranged as a three-dimensional array with shape [175296, 720, 1440]. Input data is selected as a moving window using from 2-10 past time steps.

Models

We apply a convolutional encoder-decoder neural network that delivers pixel-wise semantic segmentation that enables us to generate quantitative estimates of meteorological variables of interest such as precipitation and geopotential height at each latitude-longitude grid-point [720,1440]. Leading examples of the application of

convolutional encoder-decoder neural networks approaches include SegNet (Badrinarayanan, Kendall, and Cipolla 2017), VGG16 (Simonyan and Zisserman 2015), and U-net (Ronneberger, Fischer, and Brox 2015). Previous work by (Larraondo, Renzullo, Inza, and Lozano 2019) investigated the application of SegNet, VGG16 and U-net to the prediction of precipitation fields and concluded that U-net delivered the best estimates of precipitation while employing significantly fewer model parameters. A smaller number of model parameters results in faster computation and lower memory consumption. The U-net model has been shown to work efficiently on large images running on modern GPUs as it was originally developed on 572x572 pixel images (Ronneberger, Fischer, and Brox 2015). Based on these advantages we have adopted the U-net model as the underlying model architecture for our study.

Methodology

The DNN model described in Figure 1 was written in Python using the TensorFlow and Keras APIs. We use Horovod to implement a data-parallel model. We chose a batch size that maximises the use of GPU memory. The total batch size is then the number of GPUs multiplied by the batch size on each GPU. The total batch size is therefore a function of the number of GPUs used in model training. We run the model training for 50 epochs, save the model and report the resulting MSE values. Using the saved model we then make model predictions (inference).

In order to load the 2.1 TBytes of model training data using a data-parallel approach we distribute the model data required by each GPU onto the CPU memory of the corresponding node. This approach facilitates the rapid loading of each batch of data to GPU memory and makes possible the highly scalable data-parallel training by preventing a filesystem IO bottleneck and delivering IO bandwidth at rates even greater than from local SSD drives. The approach is particularly important when running the forecast DNN model as the batch size includes multiple input time steps and we construct a batch using a rolling window from 2-10 time steps. In order to further reduce memory usage for the forecast DNN we define a data loader so we load from memory only the data that each batch requires at each time step.

When running the forecast DNN model we also normalised the input geopotential height data and trained the model using the differences, $t_{i+1} - t_i$. We found that using the normalised differences significantly improved the model training performance. We reverse the procedure when using the trained model to make predictions of geopotential height. We used the tanh activation function and ran the model training for 20 epochs, saving the best result. We also added an additional 16 wrap longitude grid points covering all latitudes at both the eastern and western extremities in order to maintain numerical consistency and stability

of the spatial convolution resulting in a grid of [1472, 720].

Acknowledgements

This research project was undertaken on the CSIRO Bracewell cluster, with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government and on the Lassen computer a U.S. Department of Energy facility operated by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344, Lawrence Livermore National Security, LLC.

Author Contributions

Competing Interests statement

The authors declare that they have no competing financial interests.

References

- Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng (2015). TensorFlow: Large-scale machine learning on heterogeneous systems.
- Aditya Sai Srinivas, T., R. Somula, K. Govinda, A. Saxena, and A. Pramod Reddy (2020). Estimating rainfall using machine learning strategies based on weather radar data. *International Journal of Communication Systems* 33(13).
- Arcomano, T., I. Szunyogh, J. Pathak, A. Wikner, B. R. Hunt, and E. Ott (2020). A Machine Learning-Based Global Atmospheric Forecast Model. *Geophysical Research Letters* 47(9).
- Awan, A. A., A. Jain, Q. Anthony, H. Subramoni, and D. K. Panda (2020). HyPar-Flow: Exploiting MPI and Keras for scalable hybrid-parallel DNN

- Training with TensorFlow. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Volume 12151 LNCS.
- Badrinarayanan, V., A. Kendall, and R. Cipolla (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(12), 2481–2495.
- Bauer, P., P. D. Dueben, T. Hoefler, T. Quintino, T. C. Schulthess, and N. P. Wedi (2021, feb). The digital revolution of Earth-system science. *Nature Computational Science* 1(2).
- Bauer, P., A. Thorpe, and G. Brunet (2015). The quiet revolution of numerical weather prediction. *Nature* 525(7567), 47–55.
- Chollet, F. (2015). Keras.
- Dueben, P. D. and P. Bauer (2018). Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development Discussions*, 1–17.
- Hernández, E., V. Sanchez-Anguix, V. Julian, J. Palanca, and N. Duque (2016). Rainfall prediction: A deep learning approach. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Volume 9648.
- Hersbach, H., B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. De Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. de Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J. N. Thépaut (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society* 146(730).
- Hochreiter, S. and J. Schmidhuber (1997). Long Short-Term Memory. *Neural Computation*.
- Hornik, K., M. Stinchcombe, and H. White (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5), 359–366.
- Ioffe, S. and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd International Conference on Machine Learning, ICML 2015*, Volume 1.

- Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, Volume 2, pp. 1097–1105.
- Kurth, T., S. Treichler, J. Romero, M. Mudigonda, N. Luehr, E. Phillips, A. Mahesh, M. Matheson, J. Deslippe, M. Fatica, P. Prabhat, and M. Houston (2019). Exascale deep learning for climate analytics. In *Proceedings - International Conference for High Performance Computing, Networking, Storage, and Analysis, SC 2018*.
- Lagerquist, R., A. M. McGovern, and D. J. Gagne (2019). Deep learning for spatially explicit prediction of synoptic-scale fronts. *Weather and Forecasting* 34(4).
- Larraondo, P., L. Renzullo, A. Van Dijk, I. Inza, and J. Lozano (2020). Optimization of Deep Learning Precipitation Models Using Categorical Binary Metrics. *Journal of Advances in Modeling Earth Systems* 12(5).
- Larraondo, P. R., L. J. Renzullo, I. Inza, and J. A. Lozano (2019). A data-driven approach to precipitation parameterizations using convolutional encoder-decoder neural networks.
- Lecun, Y., Y. Bengio, and G. Hinton (2015). Deep learning.
- Liu, L., W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen (2020). Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision* 128(2), 261–318.
- McGovern, A., R. Lagerquist, D. J. Gagne, G. E. Jergensen, K. L. Elmore, C. R. Homeyer, and T. Smith (2019). Making the black box more transparent: Understanding the physical implications of machine learning. *Bulletin of the American Meteorological Society* 100(11), 2175–2199.
- Neumann, P., P. Düben, P. Adamidis, P. Bauer, M. Brück, L. Kornblueh, D. Klocke, B. Stevens, N. Wedi, and J. Biercamp (2019, apr). Assessing the scales in numerical weather and climate predictions: Will exascale be the rescue? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 377(2142).
- Pan, B., K. Hsu, A. AghaKouchak, and S. Sorooshian (2019, mar). Improving Precipitation Estimation Using Convolutional Neural Network. *Water Resources Research* 55(3), 2301–2321.
- Rasp, S., P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey (2020). WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting. *Journal of Advances in Modeling Earth Systems* 12(11).

- Rasp, S. and S. Lerch (2018). Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review* 146(11), 3885–3900.
- Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat (2019). Deep learning and process understanding for data-driven Earth system science. *Nature* 566(7743), 195–204.
- Ronneberger, O., P. Fischer, and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Volume 9351, pp. 234–241. Springer Verlag.
- Scher, S. and G. Messori (2018). Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*.
- Scher, S. and G. Messori (2019). Weather and climate forecasting with neural networks: Using general circulation models (GCMs) with different complexity as a study ground. *Geoscientific Model Development* 12(7), 2797–2809.
- Sergeev, A. and M. D. Balso (2018). Horovod: Fast and easy distributed deep learning in tensorflow.
- Shi, X., Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*.
- Simmons, A. J. and A. Hollingsworth (2002). Some aspects of the improvement in skill of numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society* 128(580), 647–677.
- Simonyan, K. and A. Zisserman (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
- Sønderby, C. K., L. Espeholt, J. Heek, M. Dehghani, A. Oliver, T. Salimans, S. Agrawal, J. Hickey, and N. Kalchbrenner (2020, mar). Metnet: A neural weather model for precipitation forecasting.
- Van Essen, B., H. Kim, R. Pearce, K. Boakye, and B. Chen (2015, nov). LBANN: Livermore big artificial neural network HPC toolkit. In *Proceedings of MLHPC 2015: Machine Learning in High-Performance Computing Environments - Held in conjunction with SC 2015: The International Conference for High Performance Computing, Networking, Storage and Analysis*. Association for Computing Machinery, Inc.
- Wang, B., H. Luo, J. Lu, T. Li, G. Zhang, Z. Yan, and Y. Zheng (2019). Deep uncertainty quantification: A machine learning approach for weather

forecasting. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Weyn, J. A., D. R. Durran, and R. Caruana (2019). Can Machines Learn to Predict Weather? Using Deep Learning to Predict Gridded 500-hPa Geopotential Height From Historical Weather Data. *Journal of Advances in Modeling Earth Systems* 11(8), 2680–2693.

Wikner, A., J. Pathak, B. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, and E. Ott (2020). Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems. *Chaos* 30(5).

Zhang, H. and Z. Pu (2010). Beating the Uncertainties: Ensemble Forecasting and Ensemble-Based Data Assimilation in Modern Numerical Weather Prediction. *Advances in Meteorology* 2010, 1–10.

Figure Legends

Figure 1. Schematic representation of training data and the DNN model.

a. Presents a schematic representation of the data set that we have utilised for model training and validation. The data for model training is selected from the ERA5 data set (Neumann, Düben, Adamidis, Bauer, Brück, Kornbluh, Klocke, Stevens, Wedi, and Biercamp 2019) and provides hourly estimates of variables with global coverage at a spatial resolution of $0.25^\circ \times 30\text{km}$.

b. Provides a summary of the architecture of the DNN we have trained to predict surface total precipitation fields. The model is based on the U-net DNN model (Ronneberger, Fischer, and Brox 2015). The model shown was implemented in python using the TensorFlow 2.2 API.

Figure 2. Prediction of surface total precipitation using the trained DNN model.

a. The ERA5 total precipitation field which we wish to estimate (mm). Note that the plot scale is non-linear in order to reveal the variation in precipitation over several orders of magnitude. b. The input geopotential height field at 500 hPa pressure level (m^2s^{-2}) c. The input geopotential height field at 1000 hPa pressure level (m^2s^{-2}) d. DNN estimates of surface total precipitation (mm) corresponding to 2a. Note that as in 2a., the plot scale is non-linear in order to reveal the variation in predicted precipitation over several orders of magnitude.

Figure 3. Scaling DNN training on multiple GPU devices.

a. We scale DNN training on 4-256 Nvidia V100 GPUs with 16GB memory running on the Lassen supercomputer at the Lawrence Livermore National Laboratory, USA. We used 2 years of input data 1980-81, consisting of 17521 time

steps, to train the DNN model at each GPU count. We report the time to complete training on one epoch (s) as reported by TensorFlow, representing a full pass of the model parameter optimisation process over the entire 2 year training data set.

b. We scale DNN training on 128-1024 Nvidia V100 GPUs with 16GB memory running on the Lassen supercomputer at the Lawrence Livermore National Laboratory, USA. We used 10 years of input data 1980-89, consisting of 87602 time steps, to train the DNN model at each GPU count. We report the time to complete training on one epoch (s) as reported by TensorFlow, representing a full pass of the model parameter optimisation process over the entire 10 year training data set.

Figure 4. Forecast DNN model predictions.

a. An example time series prediction of 500 hPa geopotential height at lead times $t=1, 6, 12, 18$ hours (m). each panel includes the actual geopotential height, the corresponding forecast DNN model estimates of geopotential height and the difference between these two plots. At mid-latitudes, the location of the greatest variation in geopotential height, we observe the greatest prediction errors. b. A comparison of the forecast error as RMSE (m) using a forecast DNN models trained with a range of input time steps (2-10) compared with persistence and a weekly climatology.

Tables

no tables

Figures

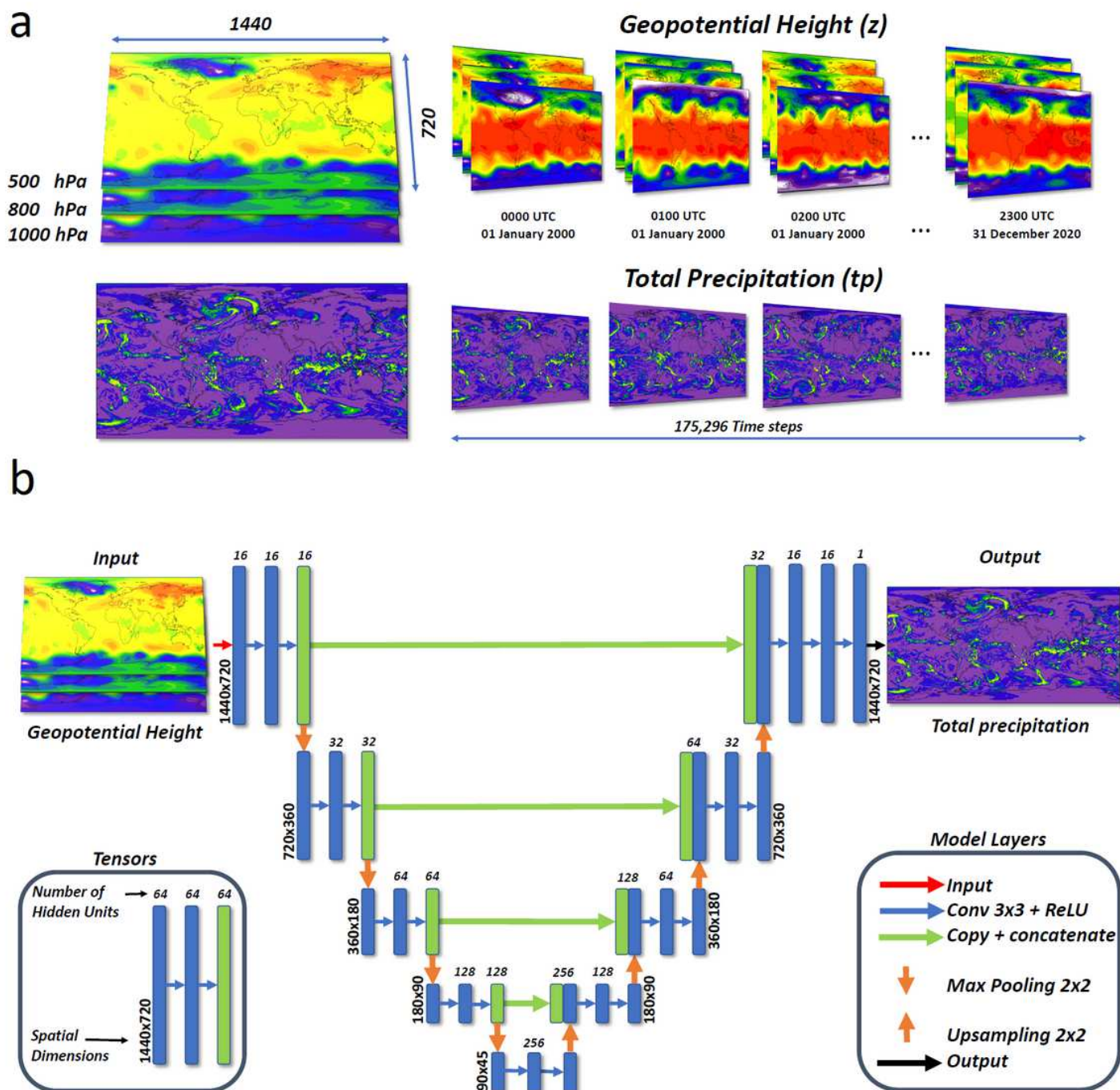


Figure 1

Schematic representation of training data and the DNN model. a. Presents a schematic representation of the data set that we have utilised for model training and validation. The data for model training is selected from the ERA5 data set (Neumann, Dübén, Adamidis, Bauer, Brück, Kornbluh, Klocke, Stevens, Wedi, and Biercamp 2019) and provides hourly estimates of variables with global coverage at a spatial resolution of $0.25^\circ \sim 30\text{km}$. b. Provides a summary of the architecture of the DNN we have trained to

predict surface total precipitation fields. The model is based on the U-net DNN model (Ronneberger, Fischer, and Brox 2015). The model shown was implemented in python using the TensorFlow 2.2 API.

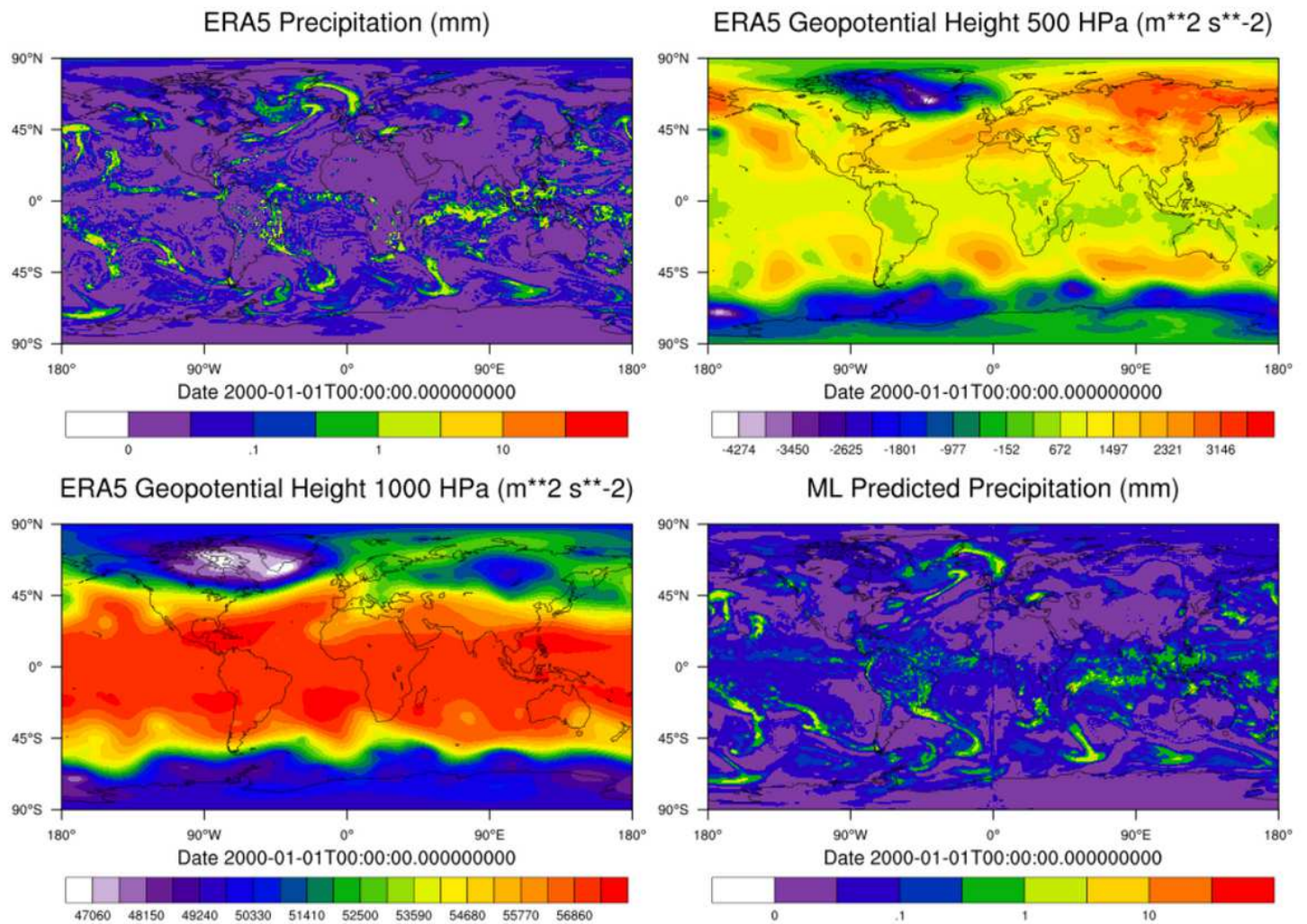


Figure 2

Prediction of surface total precipitation using the trained DNN model. a. The ERA5 total precipitation field which we wish to estimate (mm). Note that the plot scale is non-linear in order to reveal the variation in precipitation over several orders of magnitude. b. The input geopotential height field at 500 hPa pressure level ($m^2 s^{-2}$) c. The input geopotential height field at 1000 hPa pressure level ($m^2 s^{-2}$) d. DNN estimates of surface total precipitation (mm) corresponding to 2a. Note that as in 2a., the plot scale is non-linear in order to reveal the variation in predicted precipitation over several orders of magnitude.

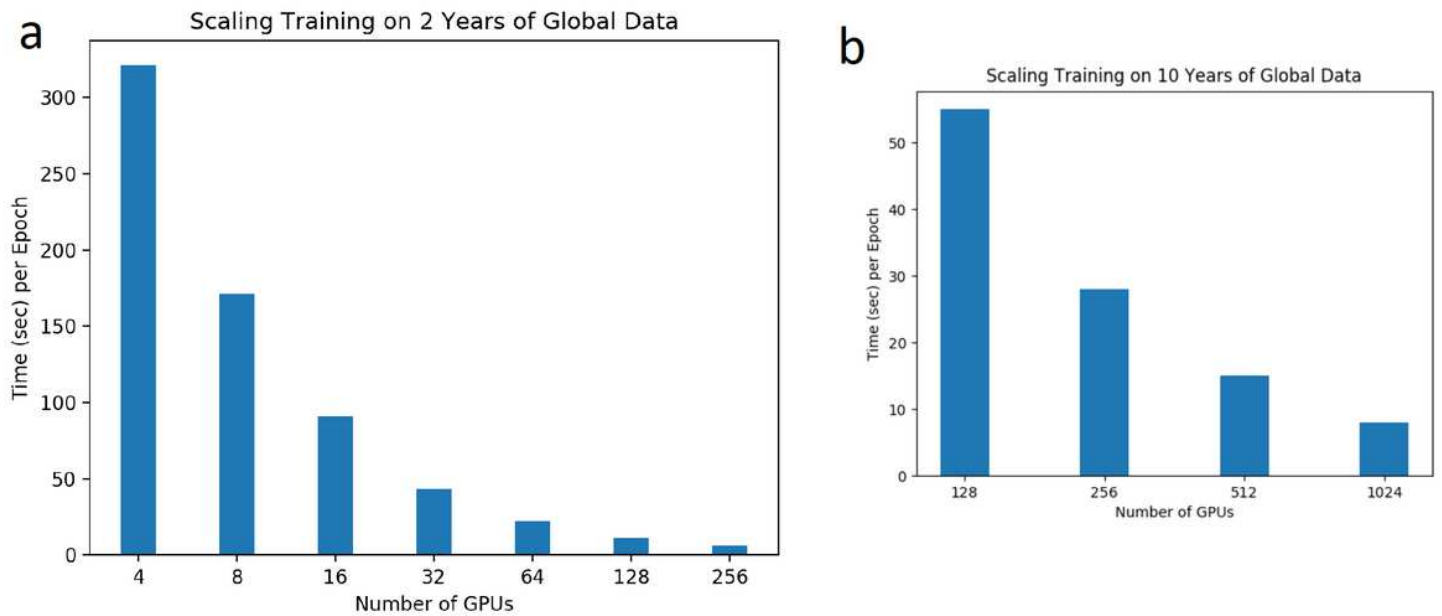


Figure 3

Scaling DNN training on multiple GPU devices. a. We scale DNN training on 4-256 Nvidia V100 GPUs with 16GB memory running on the Lassen supercomputer at the Lawrence Livermore National Laboratory, USA. We used 2 years of input data 1980-81, consisting of 17521 time steps, to train the DNN model at each GPU count. We report the time to complete training on one epoch (s) as reported by TensorFlow, representing a full pass of the model parameter optimisation process over the entire 2 year training data set. b. We scale DNN training on 128-1024 Nvidia V100 GPUs with 16GB memory running on the Lassen supercomputer at the Lawrence Livermore National Laboratory, USA. We used 10 years of input data 1980-89, consisting of 87602 time steps, to train the DNN model at each GPU count. We report the time to complete training on one epoch (s) as reported by TensorFlow, representing a full pass of the model parameter optimisation process over the entire 10 year training data set.

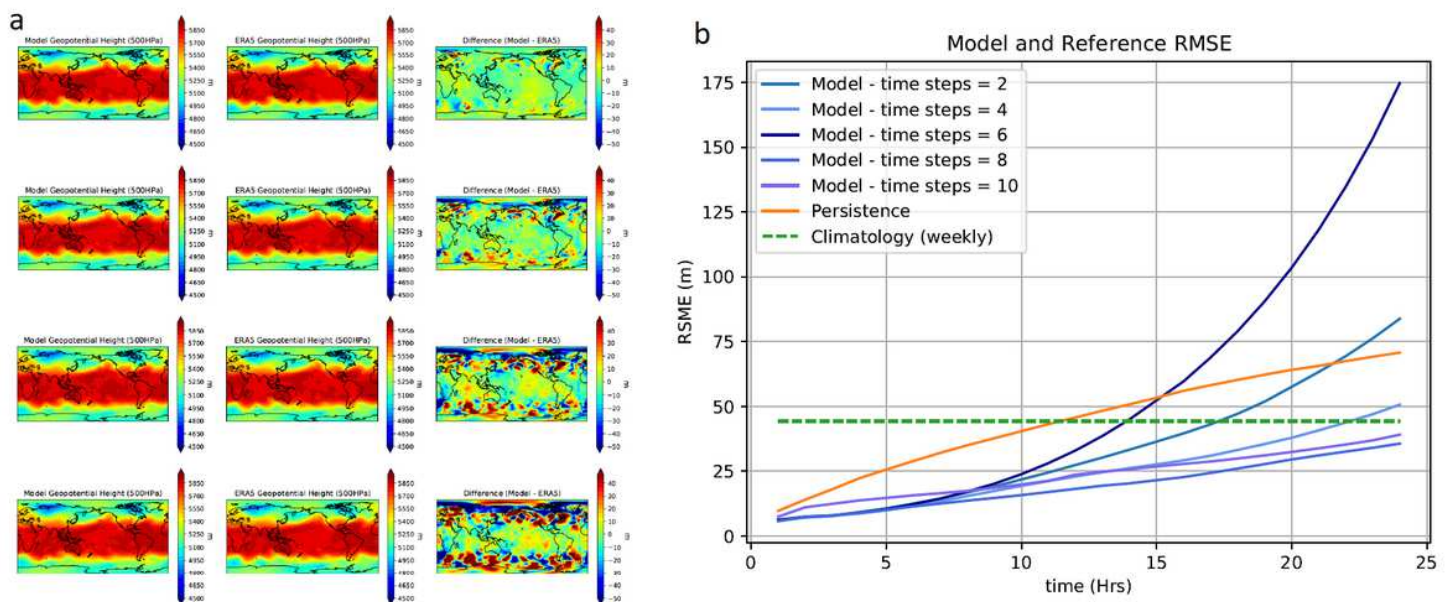


Figure 4

Forecast DNN model predictions. a. An example time series prediction of 500 hPa geopotential height at lead times $t=1, 6, 12, 18$ hours (m). each panel includes the actual geopotential height, the corresponding forecast DNN model estimates of geopotential height and the difference between these two plots. At mid-latitudes, the location of the greatest variation in geopotential height, we observe the greatest prediction errors. b. A comparison of the forecast error as RMSE (m) using a forecast DNN models trained with a range of input time steps (2-10) compared with persistence and a weekly climatology.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Taylorsupplementary.mp4](#)