AI in Weather Prediction Models

Adithya K Anil

May 29th, 2025

Contents

1. Discussion of Prithvi WxC Foundation Model for Weather and Climate

2. Deep Learning for predicting Monsoon over homogeneous regions of India.

Discussion of Prithvi WxC Foundation Model for Weather and Climate

Motivation

- AI models have been able to produce similar results to numerical models that run on HPC's.
- Can be run on relatively low spec computers depending on the type of model.
- Can address forcasting, downscaling, nowcasting accurately.
- Deep learning models do not simulate the underliying physics. They try to learn the underlying physics by error minimization and probability distributions.

Foundation Model

Foundation models are models that are trained on a large dataset and *need* to be fine tuned to tailor it for a specific task.

Prithvi WxC Model

- It is a 2.3 billion parameter foundation model developed using 160 input feature.
- It has been trained using the MERRA-2 dataset.
- It is based on an encoder decoder based architecture and also includes various new techniques to capture both local and global dependencies which are transformer based deep learning techniques.
- The ideas are mostly based on techniques from NLP and Computer Vision.

Pretraining Objective

Forecasting

Given the state of the atmosphere at t and $t - \delta t$ we should be able to predict the state of the atmosphere at $t + \delta t$.

For the task of forecasting, the best method is to use *Masked Autoencoders (MAEs)*. This is a method of self-supervised learning where the objective is given a partial image, can it reconstruct the entire original image.

The case when $\delta t = 0$ is useful in downscaling and data assimilation

Masked Autoencoders

To train a masked autoencoder what we do is we take an image and hide different patches in the image and then train the model to reconstruct the original image. The advantages of this method of training are:

- If the model is able to reconstruct the image accurately, then the model has most likely understood the most important features about that image.
- Forces the model to understand the global structure and not just memorize it.
- Can learn from unlabeled data
- Works well with transformer based architectures like Vision Transformers.
- Since the dataset is very large, masking allows the process to be memory efficient.

In Prithvi WxC model, the 2D model trained has an output prediction of the form

$$\hat{X}_{t+\delta y} = f_\theta[M_{0.5}(X_t, X_{t-\delta t})]$$

Here X_t is the data, $\hat{X}_t$ a prediction, f_θ a neural network and $M_{0.5}$ the masking operator for 50%.

Climatology

Climatology

Instead of predicting the difference from the current time stamp, we are trying to model the deviation from the historical climate at this time.

$$\frac{\hat{X}_{t+\delta t} - C_{t+\delta t}}{\sigma_C} = f_\theta \left[M_{0.5} \left(\frac{X_t - \mu}{\sigma}, \frac{X_{t-\delta\tau} - \mu}{\sigma} \right); \frac{C_{t+\delta t} - \mu}{\sigma}, S, \delta t, \delta\tau \right]$$

Here, μ and σ are per-parameter means and standard deviations (computed across space and time). $\sigma_C^2 = \mathrm{Var}(X_t - C_t)$ is the variance of the historical anomaly over all space and time. S are static inputs and δt and $\delta\tau$ are the time steps for the target and the inputs respectively.

Contd..

To get high resolution and smoothed climatology, the following are some of the conventions adopted in the model:

- For each day, we take 8 timestamps (every 3 hours). So that gives us $365 \cdot 8$ climatology points for each cell in the grid.
- Considering past 20 years data with a weighted 61 day rolling average (30 days ahead, 30 days back, and current day). So each pixel in the grid is based on $20 \cdot 61$ data points for each (day, hour) combination.
- A second order polynomial is used to give the weights so that more recent changes have more weight compared to later changes in the rolling window period.

CHECK: Using a second order polynomial it is able to account for diurnal and seasonal changes.

Normalization is avoided because it results in instabilities in the calculation.

Architecture

- Prithvi WxC is a scalable 2D vision transformer
- The model has been trained in such a way that it is not solely restricted to 2D datasets.
- The model is able to surpass the performance of a standard swin transformer by using simpler architectural models.
- Usage of MaxViT (Multi Axis Vision Transformer) which makes use of axial attention.

MaxViT

1. Convolutions layers to extract low level features.
2. Image is split into non-overlapping windows and multi head self attention is applied on each window
3. Attention is applied on each spatial axis rather than on the entire 2D. This makes it more efficient.

Contd..

Two different approaches of masking are used:

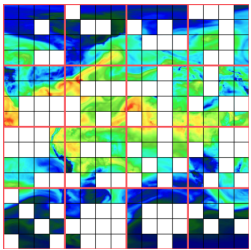


Figure: Local Masking

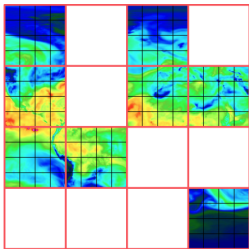


Figure: Global Masking

Overall Architecture

- The MERRA-2 re-analysis data takes the shape $T \times [P_S + (P_V \times L)] \times H \times W$. Here, T is time, fixed to 2 (Time step: current and previous). P_S are the 20 surface parameters and P_V the 10 vertical parameters at L $= 14$ vertical model levels for a total of 160 inputs.
- If we include the time T as well, we get 320 inputs. So the total spatial grid has $320 \times 360 \times 576$ (longitude) features.
- There are additional static inputs from the static parameters of the MERRA-2 data set.
- It contains 25 encoder and 5 decoder blocks. Since both encoder and decoder have attention 13+3 of these blocks do local attention and 12+2 do global attention.

Contd..

- This in total results in 2.3 billion parameters.
- Fully Sharded Data Parallelism: FSDP is a memory-efficient parallel training technique used to distribute large models across multiple GPUs.
- Flash attention: Flash Attention is an optimized algorithm for computing attention in transformer models. It's fast and memory-efficient, enabling training and inference of large transformers more effectively.

Zero Shot Validation: Masked Reconstruction

Zero Shot Performance

Generalize to new tasks, labels, or domains it wasn't trained on, simply by interpreting the task from instructions, context, or prompt.

To analyze the model's zero shot performance, we can use *Reconstruction* and *Forecasting* as two different metrics.

- The model is able to reconstruct the state of the atmosphere from $\geq 5\%$ of the original data when the samples are dense and $\geq 25\%$ when large areas are masked.
- Reconstruction performance is little affected by lead times at small values of masking.

Contd..

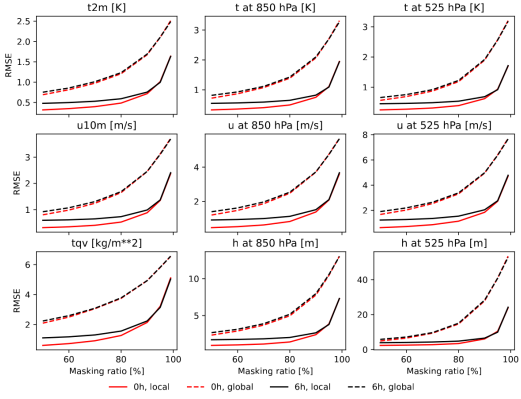


Figure: Image showing that the forecast is not much affected by the lead times at small values of masking

Forecasting

To analyze the performance in this aspect, we perform autoregressive forecasts with dense data and upto 5 days ahead

Autoregressive Forecasts

Autoregressive forecasts refer to predictions where future values are predicted based on past values of the same variable — one step at a time, in sequence.

CHECK:*Moreover, our model generates a number of forecasts for which no reference AI prediction exists. Most notably the "cloud" variables.*

Results

- Prithvi WxC performs really well at 6 hours and 12 hours lead times.
- Prithvi WxC falls behind Pangu at 66 hours lead time.

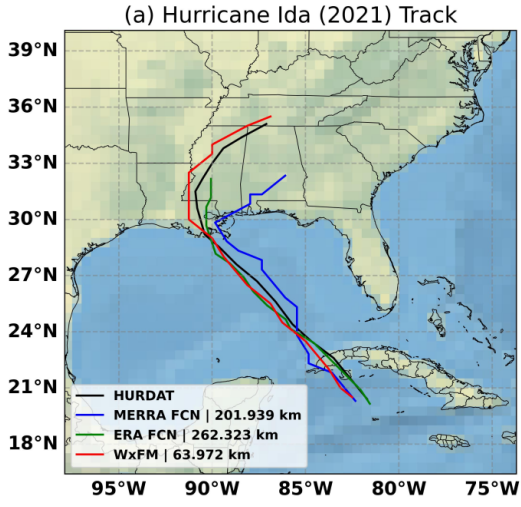
The high accuracy at short lead times implies that this method of training can be very useful in nowcasting.

Hurricane Track Forecasting

Goal: Assess its capability in forecasting the formation, dissipation, intensification, and tracking of hurricanes ranging from Category 3 to Category 5, formed over the Atlantic Ocean between 2017 and 2023.

- The results are benchmarked from HURDAT (Actual trajectory database), FourCastNet trained on MERRA-2, FourCastNet trained in ERA-5.
- Prithvi WxC demonstrated superior accuracy while predicting the track and intensity of the hurricanes.
- Among the models, the WxC model predicts the hurricane landfall most accurately in terms of both spatial location and timing, compared to the HURDAT reference.

Contd..



The error rates as compared to other models increase as the lead time increases.

Downstream Validation

The typical idea is that we use Prithvi WxC as the main core and keep adding additional embedding layers or other additional layers over it when we have to work with a specific task.

Downscaling

Downscaling is the process of refining low-scale data so that it can be used for local purposes.

Prithvi WxC is used to recover the spatial structure of coarsened near surface temperature for MERRA-2 and CORDEX-CMIP5-RCP8.5 with different input variables and input resolutions.

Contd..

- These inputs are passed through a patch embedding layer, which encodes the input into a tokenized format suitable for transformer-based processing
- An initial shallow feature extraction module processes coarse-scale patterns and controls the spatial resolution of the tokens fed into the main Prithvi WxC transformer model.
- The Prithvi WxC transformer—kept mostly frozen during fine-tuning—performs deep feature extraction, learning higher-frequency (fine-scale) spatial patterns
- Because the model uses all input tokens (no masking), a Swin-shift attention mechanism may be applied to better capture local spatial relationships while maintaining efficiency.
- To improve spatial consistency and translational equivariance, a convolutional layer is added after the transformer, addressing the transformer's limitations in handling local shifts in grid-based climate data.
- Finally, a refinement and upscaling module generates the high-resolution output fields

Analyze Performance

- Prithvi WxC improves spatial and temporal RMSE values by over a factor of 4 and also shows the best temporal correlation.
- Performs better than ClimateLearn Benchmark.

Rather than downscaling at a global context, we can also downscale at a local/regional level. To do so, a subset of the EURO-CORDEX dataset it used. With a similar architecture and method of training, the model shows very competitive results.

Gravity-Flux Parameterization

Gravity Waves

Atmospheric gravity waves (GWs) are intermittent, small-scale perturbations generated around thunderstorms, jet disturbances, flow over mountains, etc. Gravity waves couple the different layers of the atmosphere by carrying surface momentum to stratospheric and even mesospheric heights.

- The difficult in analyzing gravity waves is their low resolution data.
- We need an improved parameterized representation of gravity waves for models to show better prediction results. These include seasonal transitions, clear air turbulence, Antarctic extreme heat, and tropical predictability.

Contd..

- The usual task done using Prithvi WxC was to predict the large scale features in the atmosphere. Now we need something more small scale and specific
- The fine-tuning task uses Prithvi WxC and adapts it to generate missing small-scale details in climate predictions, using AI-learned representations instead of manually coded physics
- It also does this instantaneously (at zero-lag) to enhance the resolution and realism of coarse climate models.
- For this task, the model is fine- tuned using high-fidelity, high-resolution gravity wave data extracted from ERA5

Goal

Given the background atmospheric state around a mountain or around tropical storm, we need our ML model to predict whether the waves are spontaneously generated, and if they are, calculate the net momentum fluxes they carry (similar to predicting the cloud cover for a given set of atmospheric conditions.)

Architecture

- We are basically fine tune the existing Prithvi WxC model.
- freeze the encoder and decoder parts of Prithvi WxC and add new learnable layers before and after these layers.
- This helps extract richer representations from the raw input before feeding into Prithvi and these additional layers are trained on the high resolution data from ERA5 dataset.
- Since it's instantaneous the lead time δt is set to 0.
- This resembles a U-net like architecture.

Deep Learning for predicting Monsoon over homogeneous regions of India.

Introduction

- Predicting rainfall has become important not only at a national level, but even at regional level
- The architecture used here is a stacked auto encoder along with an ensemble of regression trees.

Goal

To be able to give accurate predictions for the different homogeneous regions of India over a long lead time so that the government can make educated decisions

Indian Monsoon

- Indian Summer Monsoon contributes for more than 70% of annual rainfall.
- India is an agricultural country and hence is highly dependent on monsoon.
- IMD has segregated India into 4 homogeneous regions:
  1. central
  2. north-east
  3. north-west
  4. south-peninsular
- The inter-annual variability at regional scales is higher compared to the national rainfall, and thus, it is more challenging to predict.
- Other works include "supervised principal component regression on the outcomes of the general circulation models."

Important points about Indian monsoon

- From using shared nearest neighbor clustering approach, we get that there are non linear dependencies between seasonal monsoon and clustering parameters.
- All the 4 regions of India have different rainfall distributions and different influencing parameters
- South peninsualar region receives much of its rainfall because of monsoon and orographic rainfall.
- Necessary to predict the monsoon for different homogeneous regions separately for better framing of policies and improving the gross productivity of the country.
- Set of climatic factors influencing monsoon evolve with time.

The focus of this model is to produce important climatic predictors for each homogeneous region of India.

Method used in this model

- unsupervised automated identification of predictors using stacked auto-encoders
- Stacked Autoencoder helps in achieving the non-linear composition of climatic variables to identify the novel monsoon predictors.
- All the climatic variables across the globe are considered and the auto encoder provides us new predictors which could be non linear combinations of these features from distant locations in the world.

Auto Encoders

They are a type of neural network used to learn efficient, compressed representations of data — often for purposes like dimensionality reduction, denoising, or generative modeling.

An important point is that though the backbone architecture is the same for all regions, the model that is trained for each region is different because each region has a different set of *Important Features*

Contd..

- The actual prediction model used is an ensemble of regression trees with bagging approach.
- The number of trees in the ensemble, the number of iterations, and other tunable parameters are set individually for each model developed for four regions.
- The process involved is a 2 step approach:
  1. Use the stacked auto encoder to identify the most dominant and influential features contributing to monsoon in the particular region.
  2. Use the ensemble of regression trees along with the identified features to predict the monsoon based on different lead times.
- The stacked auto encoders help in finding new features that have significant important and finally the top-k features are selected based on their correlation values.
- The predictors could be a combination of climatic predictors from different geographic locations as well.

Climatic Variables and Data

- Sea Level Pressure
- The zonal-wind at the pressure level of 200 hecPascal (UWND) is taken into account.
- The geo-potential height at 200 hecPascal pressure level (HGT).

All the above mentioned climatic variables are acquired from the NCEP reanalysis derived data, which are present at $2.5^\circ \times 2.5^\circ$ spatial resolution

The period between 1948 and 2000 is considered for feature learning. A test-period of 2001–2014 is considered for judging the prediction skills of the identified predictors of the monsoon.

Pre Processing

- Anomaly Data(yr, month) is calculated as

$$Anomaly(yr, month) = Var(yr, month) - mean(Var(month))$$

- Spatial rectangular regions of 20° longitude $\times 10^{\circ}$ latitude is considered all over the world, which condenses to 324 rectangular regions ((360/20) longitudinal $\times$ (180/10) latitudinal).
- A single time series is obtained by averaging all the series of a specific climatic variable within the cover of the rectangular region.

Architecture of the Stacked Auto Encoder

- The training process begins with unsupervised learning, i.e. without any knowledge about regional rainfall. This means that this process is same for all regions.
- It has 3 layers
- Different autoencoders are built by providing different climatic variables as inputs
- Identification of the potential predictors are done by the initial unsupervised learning and then supervised learning.
  1. Unsupervised learning using SAE
  2. Thresholding for feature extraction from the internal layers.
  3. Supervised ranking of the predictors based on their correlation with the regional monsoon.
- The first two steps are independant of region however the last step results in differnet results for each region.

Thresholding and Supervised Correlation

$$hid_j = \sum_{i=1}^{m} Wt_{ij} var_i, \forall i, |Wt_{ij} > threshold_j$$

This is the basic equation that controls and thresholding. This ensures that we always only consider those weights that have significant impact on the prediction for every hidden layer.

The correlation coefficient is computed for every predictor with all the regions of rainfall and they are sorted in descending order based on these values. To do so, **Pearson Correlation** is used. This process is done for several lead times from 1 month - 12 months

Ensemble of Regression Trees

- To finally make the prediction we use multiple **Regression Trees**.
- The basic idea is to use multiple predictors to eventually make a superior prediction.
- The model makes the prediction by aggregating the forecasts from each weak learning model.
- The regression tree is trained using **Bagging approach**
- The ensemble size is a hyperparameter which can be tuned based on accuracy and speed.

Future Scope of Improvement

- use of deep architecture as the convolution
- The rectified linear units can increase the non-linear characteristics of the trained network assisting in building highly com- plicated and non-linear predictors of monsoon. neural network
- the over-fitting of the network can be avoided using the dropout method

Thank you!