NinoLearn Predictions Manual

Miriam Sterl

March 5, 2021

Introduction

NinoLearn is a research framework for the application of machine learning (ML) methods for the prediction of the El Niño-Southern Oscillation (ENSO). It contains methods for downloading relevant data from their sources, reading raw data, postprocessing it and then access the postprocessed data in an easy way. Moreover, it contains models for the ENSO forecasting. For code, information and documentation on the NinoLearn framework, see `https://pjpetersik.github.io/ninolearn/intro.html` and `https://github.com/pjpetersik/ninolearn`.

This manual outlines how to use the code to make forecasts of the Oceanic Niño Index (ONI) together with an uncertainty in the predictions, using Gaussian Density Neural Network (GDNN) models. For more information, see Petersik and Dijkstra (2020).

1 Installation and set-up

You only have to go through these steps if you don't already have the NinoLearn files on your local machine.

In order to be able to run all the code, you will need to have the following Python packages installed:

- `numpy`
- `matplotlib`
- `pandas`
- `xarray`
- `dask`
- `scikit-learn`
- `netcdf4`
- `tensorflow`

Clone the following repository from GitHub and create a copy on your local machine:
`https://github.com/MiriamSterl/ninolearn`

2 Running the code

The best time to use the code to create predictions from the next month onward is around the 19th of the current month. By then, most of the data from the previous months are in, and this is when the IRI/CPC ENSO Predictions Plume is updated (`https://iri.columbia.edu/our-expertise/climate/forecasts/enso/current/?enso_tab=enso-sst_table`). However it can be done on each arbitrary day.

In the cloned `ninolearn` repository, go to the folder `predictions`. Here you find six numbered files.

In the first file, `s0_start.py`, you are required to fill in the current month and year as well as the paths to your root directory and data directory.

After that, you can run all six files in order of their numbers, without having to change anything else in the code. You do not have to do this all at once; the only important thing is that you stick to the order. Below, it is outlined what happens in each step.

Step 0: Start (`s0_start.py`). First, fill in the full path to the directory where you have the cloned `ninolearn` folder, and the full path to the directory where you want to store data. Second, fill in the current month and year. Then, run the code.

Step 1: Data downloading and preprocessing (`s1_data.py`). Run the code. Some new directories are created and the data used for training the models is downloaded and preprocessed. Furthermore, the latest common date of all data sets is saved; this will be the final date of the data sets to be used for the training.

Step 2: Training the models (`s2_training.py`). Run the code. The GDNN models are trained and saved. This takes a while!

Step 3: Making the predictions (`s3_predict.py`). Run the code. The trained models are loaded and used to make predictions of the mean and standard deviation of the ONI for the nine coming three-month seasons. These predictions will be saved as a `.csv` file in the subfolder `forecasts` within your data directory. You can load it into Python later as a Pandas DataFrame using the `pandas.read_csv` function.

Step 4: Plotting the predictions (`s4_predict.py`). This step is optional. Run the code to generate a plot of the predictions together with their uncertainty (1 standard deviation). This plot is saved as a `.png` file in the subfolder `forecasts` within your data directory.

Step 5: Plotting the predictions (`s5_finish.py`). Run the code. The directories and files with (raw and processed) data, trained models, and information saved in between (such as lead times) are removed. Next month you can have a fresh start! Note: the folder with the forecast data (`.csv` file and possibly plot of the forecast) is NOT removed. You can keep it to save the predictions every month.

References

Petersik, P. J. and Dijkstra, H. A. (2020). Probabilistic Forecasting of El Niño Using Neural Network Models. *Geophysical Research Letters*, 47(6):e2019GL086423.