Physics-informed neural networks for high-resolution weather reconstruction from sparse weather stations

Álvaro Moreno Soto, Alejandro Cervantes and Manuel Soler
Prepared by Álvaro Moreno Soto

Introduction

This document includes a set of instructions for the correct implementation of the script used for the reconstruction of high-resolution weather information starting from a sparse set of measurements provided by weather stations, as detailed in the publication under the same name and authors as this document, recently submitted to *Open Research Europe*.

The dataset

The dataset consist of a *.mat* file, which can be downloaded from Zenodo. It contains the necessary information to compute a high-resolution fluidic weather field from the reference measurements provided by weather stations (WS) on ground via the implementation of a physics-informed neural network (PINN). The script to run the training of the PINN is available at GitHub.

The dataset contains the following information for 21 different weather stations in the region of Brussels-Zaventem airport during 14 days in the year 2018:

- *Date* in format yyyy-mm-dd hh:mm:ss,
- *Lat* (latitude) and *Lon* (longitude) in degrees,
- *Alt* (altitude) in meters over sea level (SL),
- *Temperature* in Celsius,
- *WindSpeed* (wind speed) in m/s,
- projections of the wind direction into the two Cartesian coordinates, X (*WindDirectionX*) and Y (*WindDirectionY*),
- and *Pressure* in mbar.

The script **PINN_weather.py** executes the necessary transformations to allow for a proper training of the PINN in dimensionless units according to the procedure described in the article (recently submitted, DOI to be indicated once accepted for publication). The PINN regularizes the velocity and pressure fields by imposition of the Navier-Stokes equations while keeping as reference the information provided by the available weather stations (i.e. without including those removed for validation). The output of the algorithm is a high-resolution fluid domain which has been regularized by physics constraints. The final reconstruction is also better spatially-resolved, since the output grid in which the velocity and pressure fields are computed is significantly finer than the original one represented by a sparse set of WS locations.

The output of this script is a *.mat* file with name *Brussels_*N_lambda_*l_R_*r_*val.mat*, where $^*N$ is the number of epochs selected for training, $^*l$ is the value of the hypertuning parameter λ, $^*r$ is the resolution of the output grid and *val is the validation case selected. It contains:

- a high resolution velocity and pressure field on a fine structured grid at the same times of the intakes of reference information (indicated with subscript ***_PINN*),
- the reference information used for training on the retained WS (subscript ***_WS*),
- the predicted values with the PINN at the location of the WS used for training (subscript ***_WS_pred*),
- the reference information provided by the WS removed for validation purposes (subscript ***_val*),
- the predicted values with the PINN at the location of the WS used for validation (subscript ***_val_pred*),
- and finally the total loss function and the individual contributions that add up to it.

Note that for a proper execution, the dataset need be allocated in the same folder as the Python script.