Notes 2025-04-14

SMAP-HB / WRF-Hydro Project

Table of contents

This week

- ☐ Make a pip environment in my folder so it doesn't get deleted
- ☒ Standardize the data
- ☒ Just use 0-5 cm soils data
- ☐ Try adding data in decoder step
- ☐ Do PCA on static data

Make pip env

- Libraries needed are xarray rioxarray zarr netcdf4
- Issue with operating system: "On Debian/Ubuntu systems, you need to install the python3-venv"
- But when I try to apt install python3.10-venv, can't open lock file
- conda isn't found either
- Doesn't seem like there's an easy fix, so I'll just reinstall packages as needed for now

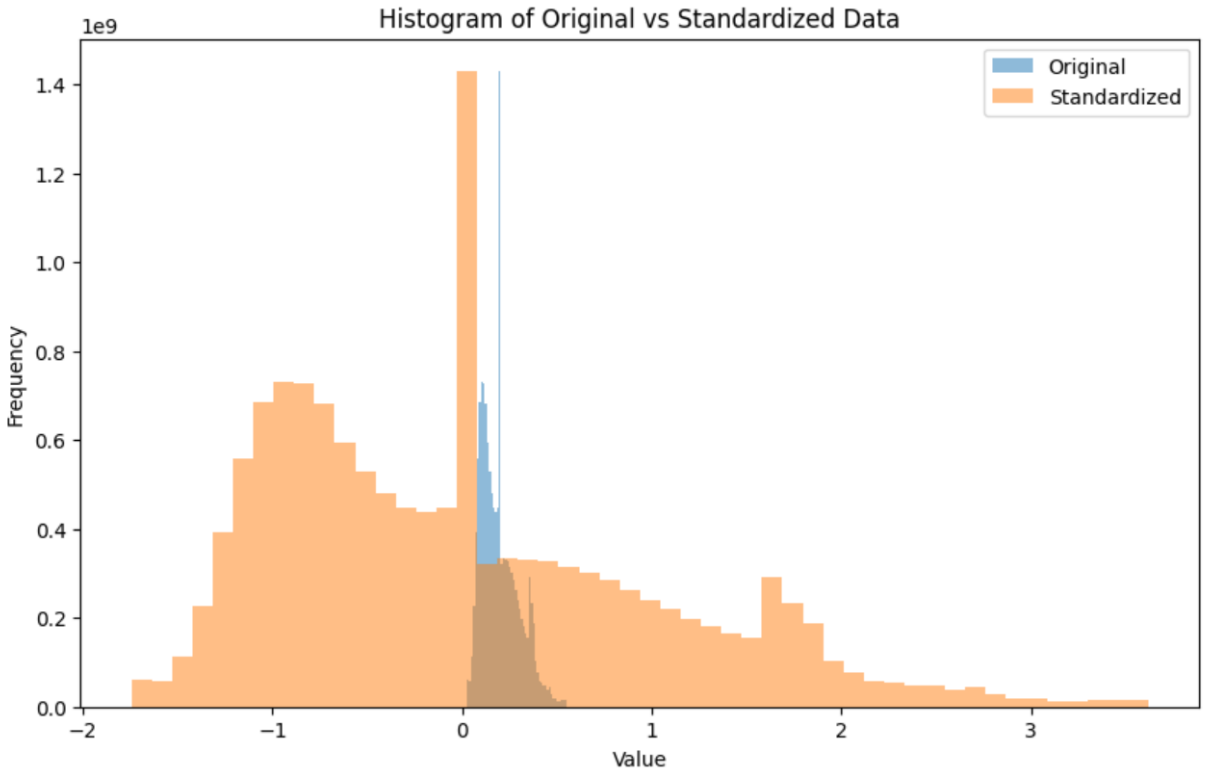
Normalize the data

Approach: Currently, data are stored as tiles, but that's inconvenient for normalizing because we need the entire dataset to get the distributions from. Use the all-data files – now is a good time to make a new dynamic zarr file with the fixed downsampled SMAP data. In order to keep the geographic information (as opposed to the bandaid I used last week with numpy), use xarray to reproject the original SMAP data to the IMERG resolution.

Fix dynamic data with xarray

- Modify zarr generation script to make dynamic data that contains downsampled SMAP
- After that, need to interpolate dynamic data again
- Then split into tiles (or just split dynamic into tiles)
- To standardize the data, do it to the zarr interpolated arrays as xarray before creating a new tile set

SMAP standardization



- Initially did resampling with nearest neighbor (reporject_match's default)

- But I think using average resampling is what we were doing before. Which is a better approach for our application?
- Split normalized data into tiles to prepare for modeling

Normalize data

- I standardized the data and then realized I should probably normalize them instead

UNet

- If I take out all the POLARIS data for deeper than 5 cm, I have 35 features, so I tried running that to start with normalized data before doing PCA
- Accidentally used a mix of standardized and normalized data, so need to redo it
- UNet still took ~7-8 hours to run – need to scrutinize architecture for possible issues

Notes for this week

- Try installing conda in home directory
- Try “module avail” - I think I tried this and module wasn’t found
- Check mode of response variable where there’s a spike; could be urban areas
- Check non-interpolated data to see if they have the spike
- Work on PCA
- Should use mean instead of nearest neighbor (which I’ve been doing)
- Add more layers after fusing at the bottleneck
- Try adding it before the last encoder layer
- Just before pooling to 1024, add the precip and SM