Notes 2025-03-17

Carrie Hashimoto

Table of contents

Meeting notes

- Check if you can include forecast points and not lake and make geogrid
- Try the one source tools instead
- Choosing version - need to use 5.3.0 because of impervious runoff scheme
- Goal: run example case with 5.3
- Or could artificially change soil moisture or Ksat to prevent infiltration
- Need to read about impervious runoff scheme
- For now just try using older versions with our data
  - Extra step before running model is to intersect NLCD impervious cover layer with soil layer and create a modified KSat layer based on this so that you have KSat = 0 there
  - How Noemi did it before: make porosity very small and then use transfer functions to recompute KSat based on that

Model

- COuld keep 360x360, a bit bigger than the paper
- multiple channels x 10 - getting heavy
- COuld do PCA analysis on static data to reduce dimensionality from 10 to 5-6

- Mask for missing data adds another layer
- FOr predictors, do nearest neighbor interpolation
- Report on how well the model is doing: RMSE, R2, etc.

Goals this week - SMAP-HB

- Check if it's possible to make the routing stack including forecast points but not lakes
- Try the open source tools
- Run the example case (WPS, ArcGIS toolkit, WRF-Hydro itself) with recent versions of software (WPS 4.6.0 ? and WRF-Hydro 5.3.0)
- So subtasks here:
- make Dockers for the old and new versions (including different datasets for each version)
- make sure the old version works
- Get new software and try running it using the training notebooks
- At some point, need to just run scripts directly without notebooks

Goals this week - WRF-Hydro

- Update data to have zarr format
- New data: include metadata, use nearest neighbor interpolation, remove unnecessary coordinates, band and spatial ref
- Try Adnan's script and adjust as needed to work with a subset of the zarr data
- Test the resulting model on new data
- Write up the results so far

Notes

Fixing the tiles

- "band" coordinate was coming from ICLUS and POLARIS, so reprocess it and remove band
- Went back to DEM and saved it with elevation as the variable name
- Took a while to figure out how to overwrite fill value and projection without this error:

ValueError: failed to prevent overwriting existing key _FillValue in attrs. This is probably an encoding field used by xarray to describe how a variable is serialized. To proceed, remove this key from the variable's attributes manually.

- Issue: fill value is nan, even though I set it to -9999… which might be fine, but the POLARIS data aren't having -9999s being filled with NaN

So the question: is fill value the value you set NaNs to, or is it the value you read as NaNs?

- The totally raw polaris data just have -9999 representing NaNs

- Super frustrating error where I can't upen netcdfs bc getting the error "did not find a match in any of xarray's currently installed IO backends" which occurs itermittently for unknown reasons, seemingly when no changes have occurred
- Reran all the scripts to make them and it worked – maybe they got corrupted due to hard drive getting ripped off
- Another dumb error: even though I set the crs in the exact same way as polaris and iclus, the dem is missing a projection… some somewhat complicated metadata issue, so just set the crs again when I load it - it's fine until you save it as a nc

Tile prep

- Saved all the inputs, static, dynamic, and target, as zarr files with 10x10 lat/lon chunks, but it doesn't seem like there's an easy way to leverage this structure for training, so just going to store in tiles again
- Check the tiles and then adapt Adnan's script to work with them
- First attempt at interpolation didn't work – try interpolating the all-domain files first so I don't miss nearest neighbors, which might happen if I interp for the tiles afterward
- Seems like rio.interpolate_na works fine – need to have spatial dims and crs set, which the tiles don't; nearest is the default interp method
- No problem settting spatial dims and crs in jupyter notebook and then doing interp, but having issues with python script to do it for all the input data – possibly due to difference between static and dynamic data spatial dims

"Only 2D/3D arrays with dimensions 'x'/'y' are currently supported. Others are appended as is. Requires either a grid mapping variable with 'spatial_ref' or a 'crs' attribute to be set containing a valid CRS. If using a WKT (e.g. from spatiareference.org), make sure it is an OGC WKT." then why does it work with one tile of the target data that have lat/lon as coords???

- Maybe when you do it at the variable level, it doesn't work - or need to set the spatial dims at the variable level

UNet

- Where is the model on RAPID? Adnan said he was running it there but I haven't found the model
- Model structure choices: if at all possible, want to have 2 branches, one for the 30 m and one for the 10 km, rather than upscaling the 30 m or filling in the 10 km with a bunch of the same values

Tiles

- static interp ran. Check to see if it worked, then use the same process on dynamic and target

Model structure outline:

- Still trying to interpolate the data :(so for now just interpolate as needed for the tiny training set
  - Had to install rioxarray into pytorch_tensorflow environment, which is already super custom (see prev notes) … create a duplicate environment in case it gets ruined
  - Try doing a diy helper function, but still have to install scipy for it in pytorch_tensorflow2 env

- Ended up using larger time chunks for the original interp files because it seemed more efficient

IMERG “Precipitation values are scaled by a factor of x10 (0.1mm) for 30 minute, 3 hour, 1 day, 3 day and 7 day files, and are scaled by a factor of x1 (1mm) for 1 month files.” – does this mean the data are in 0.1 mm instead of 1 mm? Doesn’t seem right, ex on 2017/08/27 max was around 280, which is probably 28 cm, right?

Goal: modify Adnan’s script so it loads from the zarr files and does nearest neighbor interpolation

- 9 blocks
- Skip connections - discard last skip connection so you don’t add it back on the way up
- Discrepancies in # channels when convolving up

conda install mkl intel-openmp <- to avoid issues with inconsistent libiomp5md.dll

WRF-Hydro work

Tutorial

- parent id 1,1,2,3…
- parent grid ratio 1, second can be anything
- e we and e sn are numbers of grid points followed by subdomain number of gridpoints
- dx and dy are distance between gridpoints in meters (30000m = 30 km)
- ref lat and lon are centerpoint of domain
- true lat 1 and 2 are intersection of Lambert plane on surface of earth
- stand lon - usually set to reference lon
- geog data path… will discuss later

https://www.youtube.com/watch?v=5AVCPGiO1H0&list=PL93HaRiv5QkBw_k3chrXX3pVYcOlXJiQ7

- PostWRF can be used to visualize and postprocess WRF / ERA5 data
- More on the values
- e_we first value is length of x axis, e_sn first value is number of grid points on y axis
- second domain starts at i_parent_start, j_parent_start within first domain, ex (50,20) is the starting point of a subdomain where i+parent_start is 1,50 and j_parent_start is 1,20
- Then, the second domain’s grid size is 1/3 of the parent grid size if your parent_grid_ratio is 3
- Consequently, e_we and e_sn for the subdomain must be integer multiplications*the ratio + 1

New WPS Docker using new software

Things that I need so I can run what they run in the tutorial:

S1

- example_case/suppelenental/namelist.wps to define domain and grids
- WRF_WPS – in the training, this is provided: need geogrid.exe and GEOGRID.TBL from it – use these to create geo_em.d01.nc

- wrf_hydro_model_tools/parameters/create_wrfinput/create_wrfinut.R used on geogrid to create the wrfinput_d01.nc file
- wrf_hydro_model_tools/parameters/create_soilproperties/*.R and WRF-Hydro itself for NoahMP tables so you can create soil_properties.nc

S2

- GIS_Training folder, which contains example_case subfolder with domain boundary – need this for making the routing stack
  - Critical files for the routing stack: geo_em.d01.nc (should have created it, might need to move), lake_shapes, forecast_points, NED_30m_DEM.tif – output will be saved to pocono_test.zip (or croton, in our case…)
  - Geographic dataset - included in wps training docker, not sure about standard
- Other things in GIS_Training folder: Create_Domain_Boundary_Shapefile.py and Build_GeoTiff_From_Geogrid_File.py, which take in the geogrid

S3

- NLDAS Regridding data – supposed to be in the tutorial, but they aren't so have to figure out how to dowload

WPS docker

docker pull wrfhydro/wps:conus

docker run –name wps -p 8889:8888 -v C:/Users/carri/Desktop/wrfhydro/docker_mount_example:/home/docker/mount -it wrfhydro/wps:conus

Tried this but got an error – just don't know enough about the software to run without interactive components yet – so going back to troubleshooting changes to the tutorial docker

Try to debug the standard WPS docker by opening without running the starting point? docker run –name wps_debug -p 8889:8888 -v C:/Users/carri/Desktop/wrfhydro/docker_mount_example:/home/docker/mount -it –entrypoint /bin/bash wrfhydro/wps:conus

- Great, this allows me to look at the Docker in bash without running anything

Successfully created geogrid file :)))))

Ran into same problem as before when trying to create soil properties; line 95 of mptable.tbl did not have 28 elements

Try debugging this interactively in R :(

- Tried copying in the MPTABLE.TBL from the newer version of the software and I think it fixed it… maybe a type I caused when editing
  - Might fuck things up later bc the version is different, but we'll see
- No ogr again… fml
- Try just running this locally since it's just a python script and doesn't need gfort or anything:
- Had to add ogr to the geospatial_env I've been using for most things
- Probably can do other R and python components locally and just use the Docker for compiling the models…. but why is ogr missing if it's needed? Maybe this means nobody uses the boundary script within Docker?

- Updated get boundary and wrfhydro_functions to recent versions so I can make them work with gdal
- Also had to install whitebox
- Then got this error: TypeError: in method 'Geometry_AddPoint', argument 2 of type 'double'
- Going into wrfhydro_functions.py, I got that the values are numpy floats instead of regular python floats, so they need to be converted with float() —-> result was the shapefile boundary, which matches what's expected when I plop it in Arc

python ../../GIS_Training/Build_GeoTiff_From_Geogrid_File.py -i geo_em.d01.nc -v HGT_M -o Outputs/HGT_M.tiff

- Also can't run in the Docker because it doesn't have the packaging library… useless fucks
- No issues running locally, though! TIFF also shows up when I put it in GIS

python ../../GIS_Training/Build_Routing_Stack.py -i geo_em.d01.nc -l ../../GIS_Training/Croton_Lambert/lake_shapes/lakes.shp –CSV ../../GIS_Training/Croton_Lambert/forecast_points.csv -d ../../GIS_Training/Croton_Lambert/NED_30m_DEM.tif -R 4 -t 32 -o croton_test.zip

- Routing factor is 4, cells is 32 – check notes from 2/24 for reasoning
- Also can't run this mf because I don't have packaging in the docker, so try locally

Success!!!

Tiffs in output:

- frxst_pts.tif
- FLOWDIRECTION.tif
- FLOWACC.tif
- CHANNELGRID.tif (includes lake mask)
- basn_msk.tif
- BASIN.tif
- STREAMORDER.tif (doesn't have lake mask)
- RETDEPRTFAC.tif (uniform)
- OVROUGHRTFAC.tif (uniform)
- LONGITUDE.tif
- LKSATFAC.tif (uniform)
- LATITUDE.tif
- landuse.tif
- LAKEGRID.tif
- TOPOGRAPHY.tif

Next step: get forcing data from NLDAS

- Later will want to do ERA5… take a look at the software and how to do this
- Main problem is the forcing data are missing from the tutorial, so I need to reproduce them
- Install hyplot too
- The data are in grib format, but the tutorial expects them to be netcdf … edit script and see what happens
- When I try to direcly open one of the grib files with xarray, I get this error: DatasetBuildError: multiple values for unique key, try re-open the file with one of: filter_by_keys={'typeOfLevel': 'heightAboveGround'} filter_by_keys={'typeOfLevel': 'surface'} filter_by_keys={'typeOfLevel': 'pressureFromGroundLayer'}

https://ral.ucar.edu/sites/default/files/public/8_WRF_Hydro_v5.0_Forcing_Pre_processing_Fall2018.pdf – presentation on met data prep

- WRF example period is 2010-08-01 to 2011-08-26, so get data for then

Variables needed:

- SWDOWN: shortwave rad
- LWDOWN: longwave rad
- Q2D: specific humidity
- T2D: air temperature
- PSFS: surface pressure
- U2D: near surface wind u dir
- V2D: near surface wind v dir
- RAINRATE: precip rate
- Can't get much information from the data using xr.open_dataset, so install pygrib to investigate

Not much luck figuring out NLDAS so far – seems like there's considerable processing to get it into the right NC format w correct labels, but not much documentation on how people do that

Take a look at the ERA5 tutorial? It's for WRF… https://forum.mmm.ucar.edu/threads/how-to-use-era5-data-from-copernicus-database.19293/

Regridding scripts to try, maybe have more info? https://ral.ucar.edu/dataset/earth-system-modeling-framework-esmf-regridding-scripts

New WRF-Hydro Docker using new software

Hi everyone,

Here are some updates on what I've done this week:

Missing value interpolation: - I'm using ds.rio.interpolate_na() for two-dimensional interpolation - It's been taking a pretty long time (30+ mins for 1 timestep out of 1046) to interpolate the 30 m-resolution SMAP-HydroBlocks data, so I'm trying to do it in RAPID instead to speed things up. - I've interpolated the dynamic and static data, so once the target data have been interpolated, I'll split up the interpolated data into updated tiles in RAPID

Unet architecture: - The data loader is working - I'm just using one time step of data for a starting model - Currently using 9 layers and fusing the dynamic data with the convolved layers from the static data once they've been transformed to 1x1 for each tile (since the dynamic data are 1x1 for each tile and the static data start at 360x360)

WRF-Hydro: - Figured out how to run the Dockers with up-to-date software without having to automatically run the preprocessing or model; these Dockers are designed so they automatically recognize the input files and run automatically, but that design isn't conducive to debugging (whereas the training Dockers have out-of-date software but allow don't automatically run the scripts) - Now able to use the up-to-date Dockers to run scripts manually; these dockers don't have the issue with R being out of date

- Tried using the open-source GIS pre-processing tools instead of the ArcGIS tools, and they successfully produced the hydrologic routing stack for a model with a lake and forecast points

- The Docker environment is still missing some Python packages like ogr, but I realized I can just run the Python scripts locally without issue; the Docker is mainly for running WRF-Hydro software requiring Fortran compilation
- By using the standard, non-training versions of the Dockers in this interactive mode and using local Python scripts for processing that requires specific libraries, I can resolve most of the software issues I've been having
- Next step for running WRF-Hydro is preprocessing the meteorological forcings myself; they recommend using NLDAS
- The forcing data I need are
- SWDOWN: shortwave rad
- LWDOWN: longwave rad
- Q2D: specific humidity
- T2D: air temperature
- PSFS: surface pressure
- U2D: near surface wind u dir
- V2D: near surface wind v dir
- RAINRATE: precip rate

and they need to have a specific netcdf structure with correct labels to work with the preprocessing software, so I need to write a script to convert the grib files that NLDAS data are available in to the desired format

- After I create the netcdfs, I will regrid them using more open-source preprocessing scripts
- Then I should be able to use the "routing stack" of rasters that I made with the open-source preprocessor and the regridded NLDAS data for the test case as inputs for WRF-Hydro
- Since I ran WRF-Hydro using their provided forcing data and routing stack before, I'll now be able to test the same model using inputs I created, which will help me work toward running the model in our study area with inputs I create

Next steps:

- finish interpolation of target data
- update tiles in RAPID
- continue troubleshooting starter unet and evaluate its performance
- write script to reformat raw NLDAS data so it can be used with WRF-Hydro Preprocessing System regridding software
- run WRF-Hydro with test case using the routing stack and meteorological data I prepared myself rather than the inputs given in the tutorial