FC with RG – Neuro Jetson

Srijan Ponnala, Ziyan Ishani, Derek Nguyen, Yashila Ramesh

Project Goals

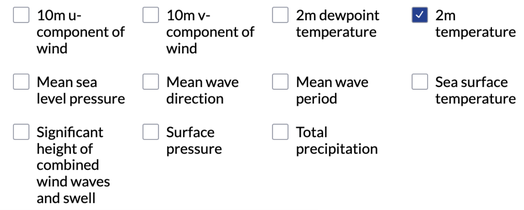
The Neuro Team is working on deploying a lightweight weather forecasting model on the Jetson Platform utilizing real-world data. The project leverages models like Microsoft ClimaX and NVIDIA FourCastNet, with the goal of determining which is better suited for deployment. Once the optimal model is selected, the team will focus on optimizing it for efficiency. As a reach goal, the team intends to configure sensors around the Jetson to collect real-time environmental data.

Project Overview and Methods:

- Sourced ERA5 hourly gridded reanalysis from Copernicus CDS, specifically for the Atlanta region [33.3°N-34.2°N, 85.0°W-83.8°W].
- Requested specific variables (temperature 2 meter), time steps, and geographic area via CDS web interface/API
- Formatted output into standardized NetCDF with dimensions (time, latitude, longitude)
- ClimaX takes gridded data from NetCDF inputs
- Model is trained and performs forecasting and projection
- We converted the Docker containers in FourCastNet to Apptainer containers. This helped ensure compatibility with the Quorra nodes.

ClimaX

- Flexible Foundation Model: Pre-trains on diverse data like ERA5; fine-tunes for varied climate tasks & variables.
  - ERA5 provides hourly atmospheric reanalysis of the global climate covering the period from January 1940 to present
- High Performance: Scalable architecture captures complex spatio-temporal patterns.
- GPU-Accelerated: Transformer's parallel matrix operations map efficiently to NVIDIA GPU cores.
- CUDA & Tensor Core Ready: Optimized via CUDA libraries (cuDNN) & specialized hardware (A30 Tensor Cores in quorra1).




```
arch.py             global_forecast  parallelpatchembed.py  regional_forecast
climate_projection  __init__.py      pretrain               utils
(climax_env) sponnala6@rg-login:~/ClimaX/src/climax$
```

FourCastNet

```
(fourcastnet) python -c "import torch; print(f'✅ FourCastNet container ready | torch: {torch
.__version__}, cuda: {torch.cuda.is_available()}')"mport torch; print(f'✅ FourCastNet conta
iner ready | torch: {torch.__version__}, cuda: {torc
✅ FourCastNet container ready | torch: 2.2.2, cuda: True
```

- FourCastNet is a deep learning model developed by NVIDIA for weather forecasting. It's designed to predict global climate and weather conditions based on large-scale data sets.

- By using CUDA, we can efficiently train FourCastNet. This is because Quorra supports multi-GPU and distributed computing, which is essential for scaling FourCastNet to handle larger datasets.

Lessons Learned

We learned that using newer models and programs means that there is not a lot of support for GUI output and visualizations, hence the command line output. Utilizing apps with more built-in features and support would lead to stronger weather and global climate visualizations.

Next Steps and Future Challenges

- **Add sensors to the Jetson:** Allows for local weather predictions using ClimaX or FourCastNet
- **Optimize model for large datasets:** With more training data the Jetson can make more accurate predictions, but must be optimized
- **Run FourCastNet within the container:** We need to ensure that FourCastNet is able to take advantage of the Quorra's parallel processing.