ai-models

Running data-driven NWP models

Baudouin Raoult, Jesper Dramsch, Florian Pinault, Mat Chantry
Introduction

% module load ai-models
% ai-models panguweather
Introduction

2023-09-03 13:25:00,810 INFO Writing results to panguweather.grib.
2023-09-03 13:25:00,810 INFO Loading pressure fields from MARS
2023-09-03 13:25:02,350 INFO Loading surface fields from MARS
2023-09-03 13:25:02,476 INFO ONNXRuntime providers: ['CUDAExecutionProvider', 'CPUExecutionProvider']
2023-09-03 13:25:02,476 INFO Using device 'GPU'. The speed of inference depends greatly on the device.
2023-09-03 13:25:37,420 INFO Model initialisation: 36 seconds
2023-09-03 13:25:40,575 INFO Starting inference for 40 steps (240h).
2023-09-03 13:25:42,718 INFO Done 1 out of 40 in 3 seconds (6h), ETA: 2 minutes 6 seconds.
2023-09-03 13:25:44,851 INFO Done 2 out of 40 in 2 seconds (12h), ETA: 1 minute 43 seconds.
2023-09-03 13:25:47,196 INFO Done 3 out of 40 in 2 seconds (18h), ETA: 1 minute 34 seconds.
2023-09-03 13:25:47,196 INFO Done 4 out of 40 in 2 seconds (24h), ETA: 1 minute 30 seconds.

[...]

2023-09-03 13:27:05,223 INFO Done 38 out of 40 in 2 seconds (228h), ETA: 6 seconds.
2023-09-03 13:27:07,400 INFO Done 39 out of 40 in 2 seconds (234h), ETA: 4 seconds.
2023-09-03 13:27:09,587 INFO Done 40 out of 40 in 2 seconds (240h), ETA: 2 seconds.
2023-09-03 13:27:09,588 INFO Average: 2 seconds per step.
ai-models --input mars --date ...
ai-models --input cds --date ...
ai-models --input file ...

ai-models

• ai-models is a command line tool
  – Designed for batch production
  – (not for notebooks)

• Uses Python’s plugin mechanism (entrypoints)
  – A plugin is a Python package that wraps a model
  – Each plugin can be installed separately
  – Solve issues of different development life cycles, ownership and licenses

• Plugins for:
  – Pangu-weather
  – Fourcastnet
  – FourcastnetV2-small
  – GraphCast
  – AIFS (ECMWF’s upcoming model)
Data pipeline
Where can I find your assets (checkpoints)?

https://download.ecmwf.int/...
What data do you need as initial conditions?

Z500, T850, LSM, 2T, 10U, 10V… on a 0.25° grid

Here some NumPy arrays

Call MARS and interpolate at desired resolution

Here are some GRIBs

Convert to GRIB

Here are some results as NumPy arrays for step N

Here some NumPy arrays

MARS

ai-models

plugin

AIFS checkpoint
Issues

• Loading Python modules is slow
  – It may take up to 30s to load pytorch on Lustre which is 50% of the time to run the inference

• Loading weights is also time consuming

• So is writing the results to disk

• Version dependency hell
  – Python, cuda, cudnn, etc
  – pytorch wants nvidia-cudnn-cu11==8.5.0.96
  – jaxlib wants nvidia-cudnn-cu11==8.9.4.25

• Models may be trained on variables (from ERA5) not generated by HRES
• **prepml** is the companion tool to **ai-models**
• It allows to run inferences over many years
• Archive all outputs in the MARS archive in research mode
• It feeds ECMWF’s scores database to that models can evaluated
• It allows user to run development code a well
• It can create ensembles using various combination of models, inputs, ...
% prepml inference config.yaml

description: Just a test

dates:
  start: 2022-01-01
  end: 2022-12-31

model:
  name: aifs
  checkpoint: /home/checkpoints/test.ckpt

runner:
  name: ai-models-dev

conda:
  clone: /home/conda/env/dev
  pip:
    - git+ssh://git@github.com/ecmwf/aifs.git@dev
Ensembles

description: ensemble with 4 models

dates:
  start: 2022-01-01
  end: 2022-03-31

ensemble:
  model:
    name:
      loop:
        - aifs
        - panguweather
        - graphcast
        - fourcastnetv2-small

output:
  number: "\{member_number\}"
  stream: enfo
  type: pf

description: ensemble with 5 checkpoints

dates:
  start: 2023-06-01
  end: 2023-08-01

ensemble:
  model:
    checkpoint:
      loop:
        - genial_surf.ckpt
        - mat_model.ckpt
        - scarlet_elevator.ckpt
        - worthy_elevator.ckpt
        - zany_serenity.ckpt

output:
  number: "\{member_number\}"
  stream: enfo
  type: pf
Future

• Short term: More models!!!

• Medium term: keep track of provenance: code, checkpoints, initial condition...

• Medium term: understand bottlenecks (mostly I/O)

• Long term: how to run large ensembles and postprocess their output on the fly?