Technical Sheet

Directory Contents

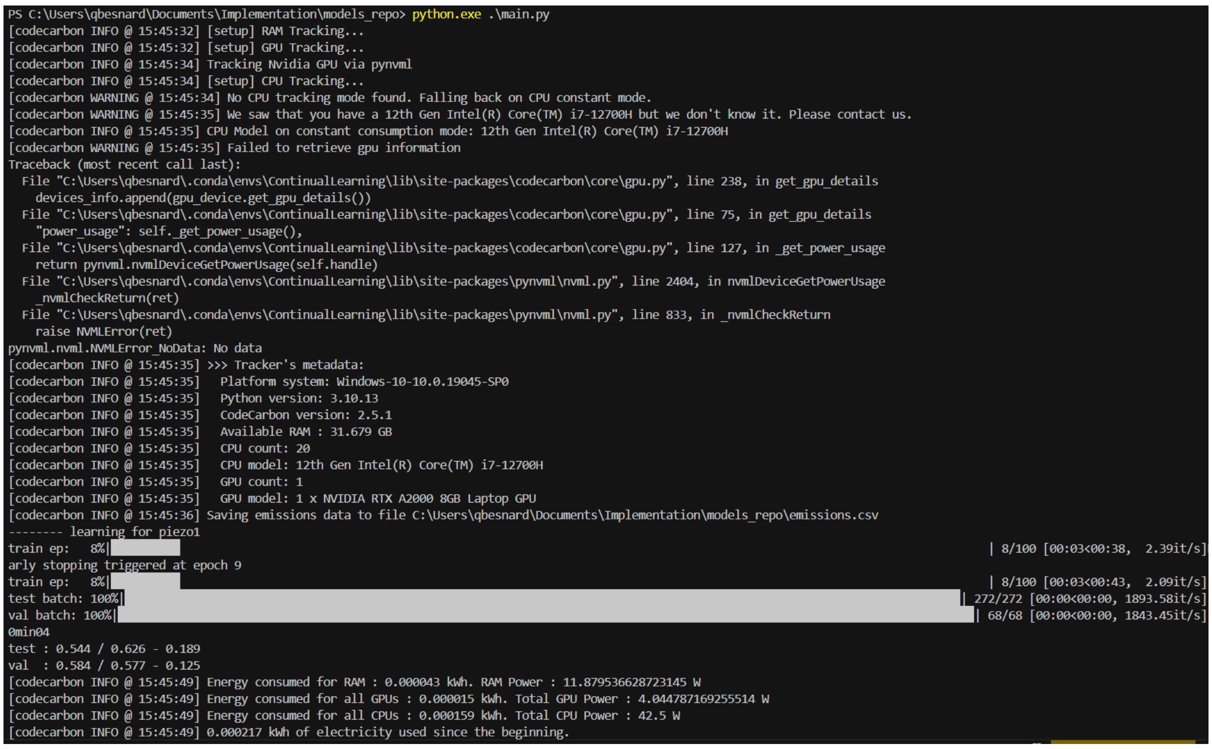
- **data**: Folder containing the piezometric data (`data/piezos/`)
- **data_provider**: Data preprocessing (train/validation/test split and dataloader)
- **exp**: Main execution code for the training and testing phases
- **layers**: File containing the internal layers of the models
- **models**: File handling the main model calls
- **tools**: Utility functions for processing and analyzing results
- **config**: Configuration file in `.json` format
- **main**: Main executable of the project
- **requirements**: Python libraries required for the environment

Execution:

To run the code, simply execute the `main.py` file with a command such as:

- "python main.py"

within a Python environment.
Personally, I use a Conda environment via Anaconda Navigator and open the repository in Visual Studio Code.
Below is an example of an execution trace.



```
PS C:\Users\qbesnard\Documents\Implementation\models_repo> python.exe .\main.py
[codecarbon INFO @ 15:45:32] [setup] RAM Tracking...
[codecarbon INFO @ 15:45:32] [setup] GPU Tracking...
[codecarbon INFO @ 15:45:34] Tracking Nvidia GPU via pynvml
[codecarbon INFO @ 15:45:34] [setup] CPU Tracking...
[codecarbon WARNING @ 15:45:34] No CPU tracking mode found. Falling back on CPU constant mode.
[codecarbon WARNING @ 15:45:35] We saw that you have a 12th Gen Intel(R) Core(TM) i7-12700H but we don't know it. Please contact us.
[codecarbon INFO @ 15:45:35] CPU Model on constant consumption mode: 12th Gen Intel(R) Core(TM) i7-12700H
[codecarbon WARNING @ 15:45:35] Failed to retrieve gpu information
Traceback (most recent call last):
  File "C:\Users\qbesnard\.conda\envs\ContinualLearning\lib\site-packages\codecarbon\core\gpu.py", line 238, in get_gpu_details
    devices_info.append(gpu_device.get_gpu_details())
  File "C:\Users\qbesnard\.conda\envs\ContinualLearning\lib\site-packages\codecarbon\core\gpu.py", line 75, in get_gpu_details
    "power_usage": self._get_power_usage(),
  File "C:\Users\qbesnard\.conda\envs\ContinualLearning\lib\site-packages\codecarbon\core\gpu.py", line 127, in _get_power_usage
    return pynvml.nvmlDeviceGetPowerUsage(self.handle)
  File "C:\Users\qbesnard\.conda\envs\ContinualLearning\lib\site-packages\pynvml\nvml.py", line 2404, in nvmlDeviceGetPowerUsage
    _nvmlCheckReturn(ret)
  File "C:\Users\qbesnard\.conda\envs\ContinualLearning\lib\site-packages\pynvml\nvml.py", line 833, in _nvmlCheckReturn
    raise NVMLError(ret)
pynvml.nvml.NVMLError_NoData: No data
[codecarbon INFO @ 15:45:35] >>> Tracker's metadata:
[codecarbon INFO @ 15:45:35]   Platform system: Windows-10-10.0.19045-SP0
[codecarbon INFO @ 15:45:35]   Python version: 3.10.13
[codecarbon INFO @ 15:45:35]   CodeCarbon version: 2.5.1
[codecarbon INFO @ 15:45:35]   Available RAM : 31.679 GB
[codecarbon INFO @ 15:45:35]   CPU count: 20
[codecarbon INFO @ 15:45:35]   CPU model: 12th Gen Intel(R) Core(TM) i7-12700H
[codecarbon INFO @ 15:45:35]   GPU count: 1
[codecarbon INFO @ 15:45:35]   GPU model: 1 x NVIDIA RTX A2000 8GB Laptop GPU
[codecarbon INFO @ 15:45:36] Saving emissions data to file C:\Users\qbesnard\Documents\Implementation\models_repo\emissions.csv
-------- learning for piezo1
train ep:   8%|█████                                                         | 8/100 [00:03<00:38,  2.39it/s]
arly stopping triggered at epoch 9
train ep:   8%|█████                                                         | 8/100 [00:03<00:43,  2.09it/s]
test batch: 100%|███████████████████████████████████████████████████████████| 272/272 [00:00<00:00, 1893.58it/s]
val batch: 100%|██████████████████████████████████████████████████████████████| 68/68 [00:00<00:00, 1843.45it/s]
0min04
test : 0.544 / 0.626 - 0.189
val  : 0.584 / 0.577 - 0.125
[codecarbon INFO @ 15:45:49] Energy consumed for RAM : 0.000043 kWh. RAM Power : 11.879536628723145 W
[codecarbon INFO @ 15:45:49] Energy consumed for all GPUs : 0.000015 kWh. Total GPU Power : 4.044787169255514 W
[codecarbon INFO @ 15:45:49] Energy consumed for all CPUs : 0.000159 kWh. Total CPU Power : 42.5 W
[codecarbon INFO @ 15:45:49] 0.000217 kWh of electricity used since the beginning.
```

Note: The "Traceback" warning message originates from **CodeCarbon**, as it cannot detect a GPU (the execution was done in a CPU-only local environment).

Information

The configuration file is currently set to use the **DLinear** model on the **piezo1** dataset, with an input window of 15 data points and a prediction horizon of 30 (daily time steps, i.e., one-month forecasts). Further details about parameters can be found in the `main.py` file.
The model and parameters are lightweight, allowing the experiment to run locally on CPU in just a few seconds.

Upon execution, three folders are generated: **results**, **figs**, and **emissions**, each containing the following:

- **results**:
  You will find `results/piezo1/`, this folder contains:
  - A file named `execution_name.csv`, holding the last predictions on the test set (if `pred_len=30`, it includes all 30-step-ahead predictions, as the model generates outputs from step 1 to 30).
  - A `metrics.json` file containing all evaluation metrics computed over the test predictions.
- **figs**:
  Contains the figure representing the test phase results, highlighting the top 25% of attention weights derived from the **Grad-CAM** analysis.
- **emissions**:
  Contains CSV files reporting the **environmental cost** of the model execution (e.g., energy consumption, CO_2 trace, etc.).

Note

If a pre-trained model is required, I can provide one upon request. However, given the lightweight nature of the current model and configuration, if local training is not feasible, deploying a pre-trained model might not be either.