12/06/2023

Weather Nowcasting with GNNs and LSTM

Course: Theory and practice of Deep Learning
Professor: Fabrizio Silvestri
Sapienza, DIAG, a.y. 2022-2023

Simone Rossetti, PhD student in Engineering in Computer Science

Overview

1. Introduction Weather Prediction
2. Deep Learning in WP literature
3. Copernicus ERA5 dataset
4. WP with deep autoregressive models
    a. Data preprocessing and windowing
    b. Temporal encoding
    c. GNNs: GCNs and GATs
    d. Spatio-temporal encoding
5. Implementation and results
6. Conclusions

Introduction

The theory-driven *Numerical Weather Prediction* (*NWP*) methods, which solves a set of FDEs and nonlinear PDEs (Navier–Stokes equations), face many challenges, e.g.:

- incomplete understanding of *physical mechanisms*,
- *uncertainty* given by small differences in models' initial conditions,
- difficulties in obtaining *useful knowledge* from the observed data (PBs),
- requires *powerful computing* resources (TBs of simulations results),

Introduction (contd’)

Deep Learning is a data drive approach that can act like supplement to NWP:

- to learn the *temporal* and *spatial* features from the spatio-temporal data,
- to enhance *robustness* to noisy data and initial conditions,
- to catch the correlation between features,

DL-based weather prediction (*DLWP*) has attracted attention of many institutions, such as ECMWF, Nature, Google, Alibaba Group, etc.

DLWP architectures towards data characteristic

- Autoencoders are employed to process high-dimensional (multivariate), real-type meteorological data collections;
- CNNs are used to extract spatial patterns in image data from satellites (100s of TBs per day), both for prediction, and detection of extreme weather;
- RNNs are used to recognize temporal relationships between data elements in meteorological long time sequence.

Table 1
Selection of DNN models. [Xiaoli et al.]

Data characteristics	*Potential DNN architectures*
High-dimensional real-type	Autoencoder-based DNNs
Image	CNN-based DNNs
Long time sequence	RNN-, LSTM-based DNNs

DLWP hybrid architectures

Precipitation nowcasting:

- *ConvLSTM* [Shi et al.] implement a location-invariant approach;
- *TrajGRU* [Shi, Gao et al.] captures spatio-temporal correlations in local neighborhood set;
- *U-Net* [Agrawal et al.] realises image to image translation;
- *MetNet* [Sønderby et al.] uses axial attention mechanism and ConvLSTM – the very first model to beat NWP accuracy.

DLWP hybrid architectures (contd')

More general weather forecasting:

- *PredRNN* [Wang et al.] adopt dual memory to catch both spatial and temporal variations;
- *PredRNN++* [Wang, Gao et al.] adopt *CausalLSTM* and residual connections.

PredRNN and its variants are general frameworks and have been extended to precipitation nowcasting successfully [Wang, Hong et al.].

Recently many works have been using GNNs to model physical systems [Pfaff et al.]:

- [Keisler] an *MPNN* encoder learn to model global atmosphere for 6h;
- [Lam et al.] an *MPNN* at multiple spatial scales for 10d forecasts with 6h intervals, beating NWP in at some resolutions and time intervals.

DLWP vs. NWP

- Precise forecasting with high resolutions (low spatial and temporal scales) – *nowcasting*;
- Learn variable mapping at different scales;
- Compress data capturing spatio-temporal features.

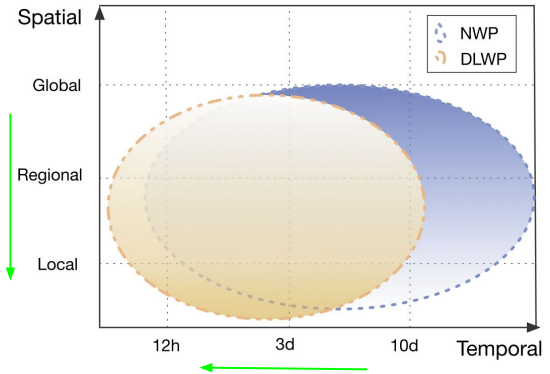


Fig. 4. Performance comparison between DLWP and NWP at different temporal and spatial scales. In the area covered by each approach, the darker the color, the better the performance.

[Xiaoli et al.]

Table 3
Analysis of temporal and spatial scales.

[Xiaoli et al.]

Scales	*Key problems*	*Limitations of NWP*	*Solutions of DLWP*
Small-scale	High forecast accuracy requires high resolution	Computational limits due to resolution	Precise forecasting with high resolution
Large-scale (for extreme weather detection)	Big datasets inputs, thresholds setting	Dependent on physical model, subjective thresholds, long simulation time	Capturing spatio-temporal features from data
Multi-scale	Multi-resolution datasets	Computational limits, dependent on completely different physical and dynamical models	Learning mapping between variables of different scales

Table 4
Typical datasets and benchmarks for DLWP.

Datasets	Observation	IGRA[1] CHIRPS[2] Station Observations
	Reanalysis	CFSR[3] ERA-Interim reanalysis[4] 20 century reanalysis[5] NCEP-NCAR reanalysis[6]
	Simulations	CAM run Published by ECMWF
	Hybrid	Hybrid of observation, reanalysis, simulation or datasets generated by data mining
Benchmarks	Numerical models	ECMWF (global model) ALADIN (regional model) HRRR (high resolution model)
	Traditional ML methods	SVM, LR, *etc.*
	Specific for DLWP	rainymotion[7] ExtremeWeather[8,9]

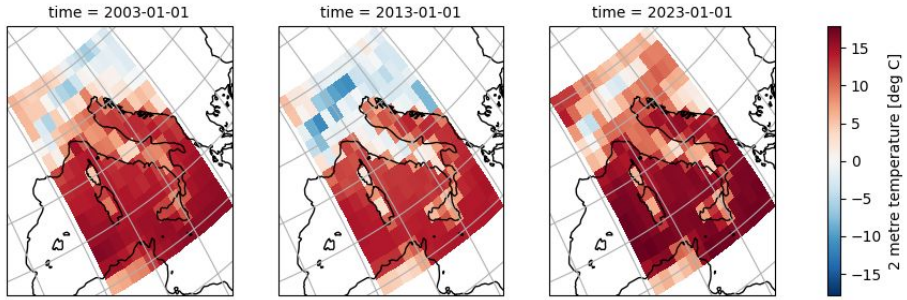
Our dataset

We use a real-world dataset reanalysis ERA5 [Hersbach et al.] from ECMWF:

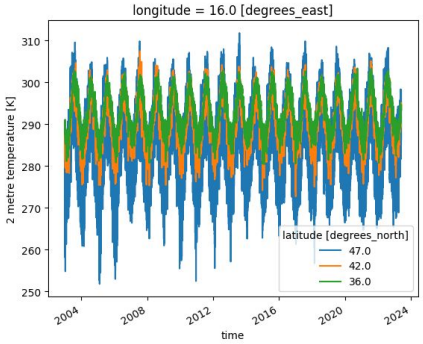
- high temporal and spatial resolution;
- real-type data;
- multivariate time series in a grid structure;

```
Dataset:
      period: 2003-01-01 to 2023-06-04, step: 1 h
      longitude: 6.0 to 19.0, step: 1.0 deg
      longitude: 48.0 to 36.0, step: 1.0 deg
Variables:
      u100: [m s**-1] 100 metre U wind component
      v100: [m s**-1] 100 metre V wind component
      u10: [m s**-1] 10 metre U wind component
      v10: [m s**-1] 10 metre V wind component
      t2m: [K] 2 metre temperature
      d2m: [K] 2 metre dewpoint temperature
      sp: [Pa] Surface pressure
      tp: [m] Total precipitation
      z: [m**2 s**-2] Geopotential
      mcc: [(0 - 1)] Medium cloud cover
Shape: (179019,13,14,10), 13*14=182 nodes
```

t2m at 10y distance

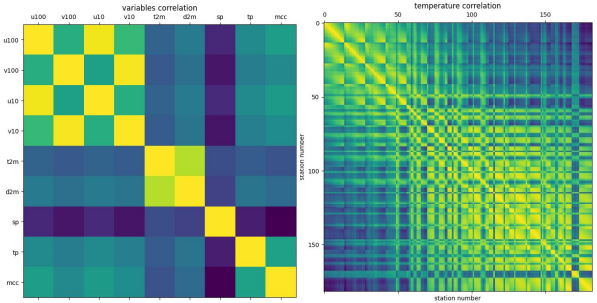


t2m series for lon=16, len={36,42,47}:
seasonality and decreasing *correlation* with latitude changing



Seasonality and correlation

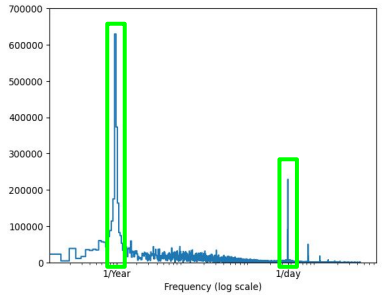
- Seasonality in a time series of hourly temperatures is of interest and we want to directly model it (e.g., using sine and cosine terms);
- Variables correlations are useful for forecasting, even when there is no causal relationship between the two variables, but not for analysis of the contributions;
- Correlation is both temporal and spatial.



Temporal correlation of variables

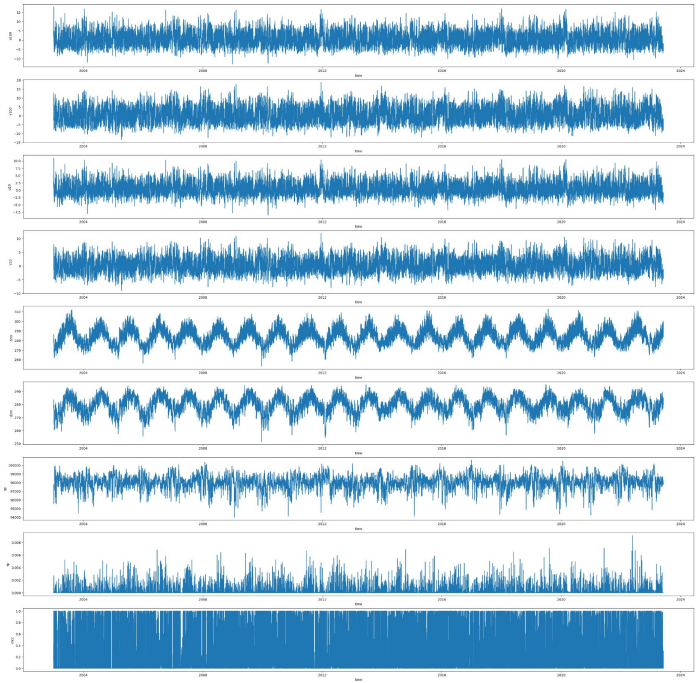
Spatial correlation of temperature

Frequencies of 't2m' time series (20y)



We can determine important frequencies of time series using Fast Fourier Transform (FFT)

Dataset visualization, station 0



Weather nowcasting with deep autoregressive models

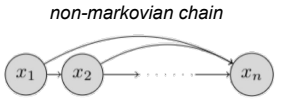
TLDR: *Deep autoregressive models [Graves] are sequence models, yet feed-forward; generative models, yet supervised. They are a compelling alternative to RNNs for sequential data, and GANs for generation tasks* (https://www.georgeho.org/deep-autoregressive-models/#fn:1).

Problem: Let's have a dataset ***D*** with *n*-dimensional data points ***x***: $\mathbf{x} \in \mathbb{R}^n$ $\mathbf{x}_{<i} = [x_1, x_2, \ldots, x_{i-1}]$

By the chain rule we factorize the joint distribution over the *n*-dimensions: $p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots, x_{i-1}) = \prod_{i=1}^{n} p(x_i | \mathbf{x}_{<i})$

Objective: Minimize the divergence of the distributions is equivalent to *MLE*, given *i.i.d.* assumption:

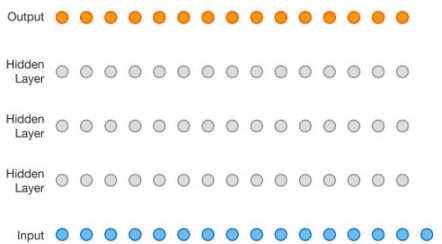
$$\min_{\theta \in \mathcal{M}} d_{KL}(p_{\text{data}}, p_\theta) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}}\left[\log p_{\text{data}}(\mathbf{x}) - \log p_\theta(\mathbf{x})\right] \approx \min_{\theta \in \mathcal{M}} -\frac{1}{|D|} \sum_{\mathbf{x} \in \mathcal{D}} \sum_{i=1}^{n} \log p_{\theta_i}(x_i | \mathbf{x}_{<i}) = \mathcal{L}(\theta | \mathcal{D})$$



Implementation: We assume data distributes normal with unit covariance, thus we parametrize only the mean and minimise the *MSE*:

$$p_{\theta_i}(x_i | \mathbf{x}_{<i}) = p_{\theta_i}\left(x_i; \mu_{\theta_i}, {\sigma_{\theta_i}}^2\right) = \frac{1}{\sqrt{2\pi {\sigma_{\theta_i}}^2}} exp\left(-\frac{\left(x_i - \mu_{\theta_i}\right)^2}{2{\sigma_{\theta_i}}^2}\right)$$

$$\mathcal{L}(\theta | \mathcal{D}) \equiv \min_{\theta \in \mathcal{M}} \frac{1}{|D|} \sum_{\mathbf{x} \in \mathcal{D}} \sum_{i=1}^{n} \log \sigma_{\theta_i} \sqrt{2\pi} + \log 2\sigma_{\theta_i} + \left(x_i - \mu_{\theta_i}\right)^2 \approx \min_{\theta \in \mathcal{M}} \frac{1}{|D|} \sum_{\mathbf{x} \in \mathcal{D}} \sum_{i=1}^{n} \left(x_i - \mu_{\theta_i}\right)^2 for\ \sigma_{\theta_i} = 1$$



https://deepmind.com/blog/wavenet-generative-model-raw-audio/

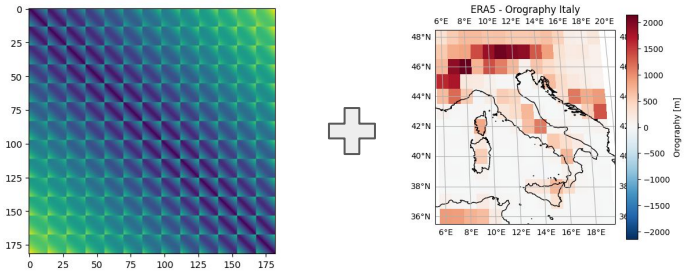
Data preprocessing and windowing

1. data inspection for outliers removal;
2. data temporal split (train 70%, val 20%, test 10%);
3. features engineering of *wind* data i.e. from polar to cartesian;
4. features normalization via standardization using training data;
5. cartesian positional encoding (lat and lon to meters + altitude);
6. 2 sinusoidal temporal encoding (Y and D).

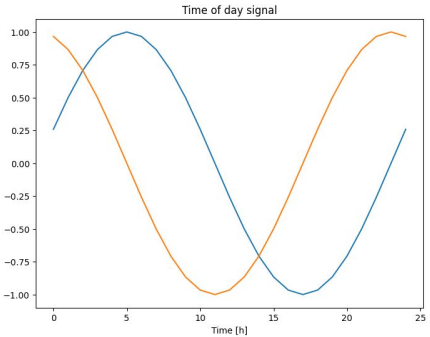
The final dataset:

- *24h history length;*
- *1h shift*;
- *8h forecast horizon*;
- train set size 125313;
- val set size 35803;
- test set size 17901;

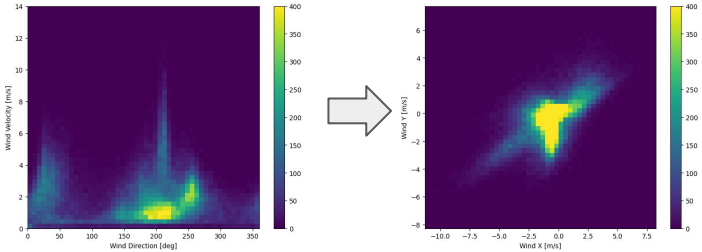
Positional encoding



Temporal encoding

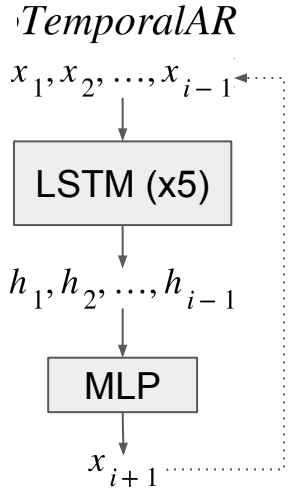


Polar to cartesian



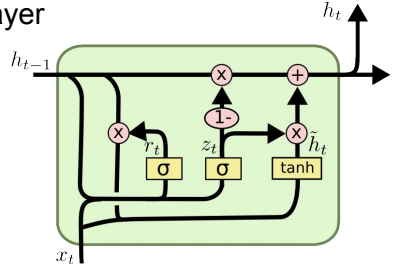
Baseline: temporal encoding

- Deep multivariate autoregressive model:
  - 9 spatio-temporal variables
  - LSTM (5 layers) perform a temporal encoding of the sequence;
  - MLP (2 layers - nonlinear activation) project encoding into data domain.
  - MSE loss
  - MAE accuracy metric



Notice: *this network ignores spatial relations*

LSTM layer



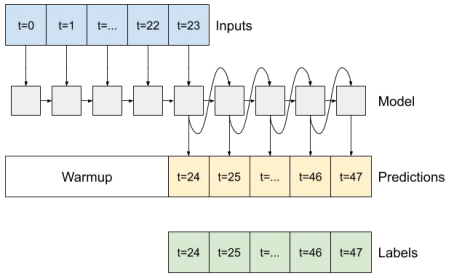
$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Autoregressive scheme



Graph Neural Networks (GNNs)

GNNs are neural networks that can be directly applied to *graphs*, and provide an easy way to do node-level, edge-level, and graph-level prediction tasks:

- Graph Convolution (GCNs) – local aggregation of hidden features [Kipf at al.];

$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \frac{\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}}{|\mathcal{N}(v)|} + B^{(k)} \cdot h_v^{(k-1)}\right) \quad \text{for all } v \in V.$$

- Graph Attention (GATs) – relearning edge weights for local aggregation of hidden features [Veličković et al.].

$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \left[\sum_{u \in \mathcal{N}(v)} \alpha_{vu}^{(k-1)} h_u^{(k-1)} + \alpha_{vv}^{(k-1)} h_v^{(k-1)}\right]\right) \quad \text{for all } v \in V.$$

$$\alpha_{vu}^{(k)} = \frac{A^{(k)}(h_v^{(k)}, h_u^{(k)})}{\sum_{w \in \mathcal{N}(v)} A^{(k)}(h_v^{(k)}, h_w^{(k)})} \quad \text{for all } (v,u) \in E.$$

GNN+LSTM: spatio-temporal encoding

ERA5 weather observations present a natural grid structure, writable into temporal graph *V* [Yu et al.]:

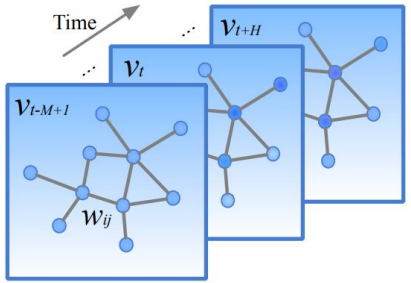
- nodes are weather variables observed at station *i* at time *t*;
- edges *w* between nodes are euclidean distances.

The resulting undirected graph has:

- #nodes 182;
- #edges 3220;
- max spatial distance 250 Km;
- node's degree: min 7, max 24, mean 17.7, std 3.7.

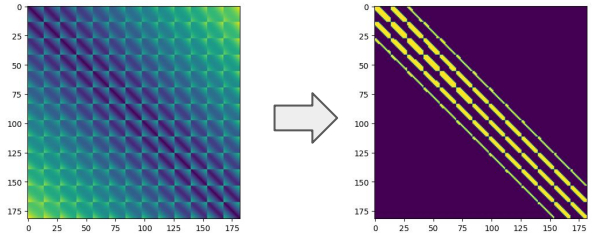
$$\hat{v}_{t+1}, ..., \hat{v}_{t+H} = \underset{v_{t+1},...,v_{t+H}}{\arg\max} \log P(v_{t+1}, ..., v_{t+H} | v_{t-M+1}, ..., v_t),$$

[Yu et al.]



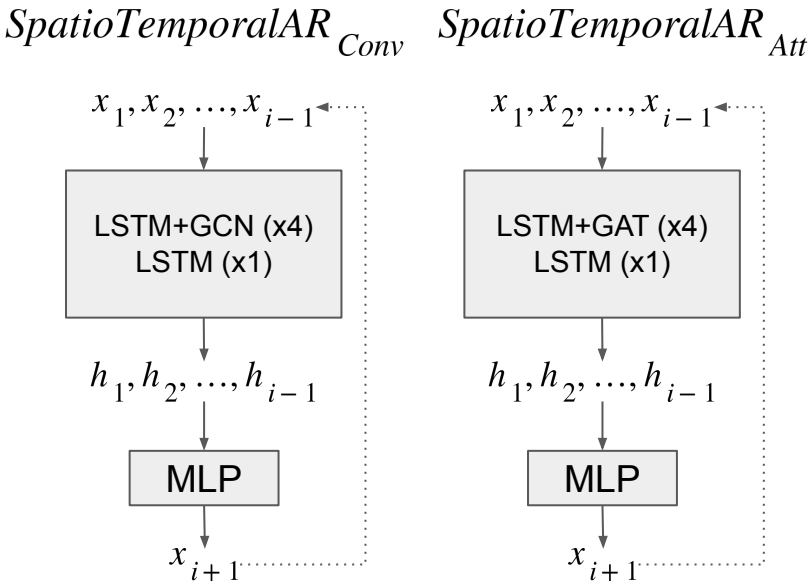
Distance matrix

Affinity matrix



GNN+LSTM: spatio-temporal encoding (contd')

- Following [Yu et al.] we compute an adjacency matrix from distance matrix;
- We implement a GCN layer [Kipf at al.];
- We implement a GAT layer [Veličković et al.];
- We implement spatio-temporal layers which extend the baseline LSTM layer, following the work by [Yu et al.]:
  - LSTM+GCN block – fixed local aggregation of temporal preprocess;
  - LSTM+GAT block – learned local aggregation from temporal preprocess;



Implementation details

- Tensorflow 2.12 framework;
- 1 Nvidia A6000 with 48GB RAM;
- History length 24h, prediction length 8h;
- We use single hidden unit (h) reference to build the model;
- All models are 5 sequential blocks with units [h, 2*h, 3*h, 2*h, h];
- All models have final 2 layer MLP with 2*h hidden units and RELU activation;
- Dropout of 0.05 is applied after every projection in all blocks;
- Both GCN and GAT uses RELU activation after output projection;
- GAT uses LeakyRELU with alpha=0.2 for attention scores [Veličković et al.];
- Adam optimizer with lr from 1e-2 to 1e-3 (next slide);
- MSE loss and MAE metrics;
- All trainings run for 80 epochs over the entire train set.

Quantitative results

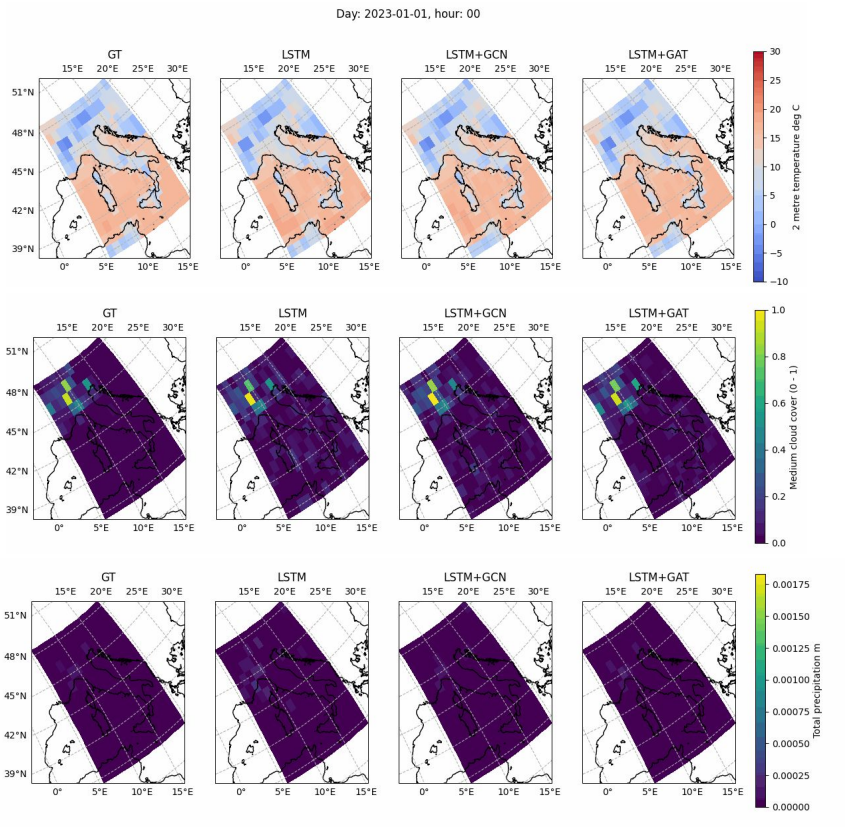
Model	Batch size	Learning rate	Training Time	Hidden dim (h)	Aggregation type	Combination type
LSTM	512	1e-2	4h	32	—	—
LSTM+GCN	192	4e-3	7h	32	mean	concat
LSTM+GAT	72	1e-3	16h	32	att sum	add

Model 24h/8h	VAL		TEST	
	MAE	MSE	MAE	MSE
Last value	0.2715	0.3325	0.2734	0.3295
LSTM	0.2380	0.2151	0.2411	0.2120
LSTM+GCN	0.2173	0.1713	0.2179	0.1737
LSTM+GAT	**0.1984**	**0.1687**	**0.1936**	**0.1594**

Model 24h/16h	VAL		TEST	
	MAE	MSE	MAE	MSE
Last value	0.3617	0.5095	0.3574	0.4895
LSTM	0.3050	0.3117	0.3016	0.2995
LSTM+GCN	0.2664	0.2485	0.2560	0.2433
LSTM+GAT	**0.2431**	**0.2296**	**0.2335**	**0.2307**

Model 24h/8h	TEST — MAE								
	u100	v100	u10	v10	t2m	d2m	sp	tp	mcc
Last value	0.3564	0.3596	0.3491	0.3568	0.1984	0.1227	**0.0177**	0.2665	0.4329
LSTM	0.2962	0.3002	0.2864	0.2927	0.1090	**0.1184**	0.0629	0.2698	0.4334
LSTM+GCN	0.2638	0.2655	0.2568	0.2669	0.1031	0.1266	0.0735	0.2347	0.3700
LSTM+GAT	**0.2354**	**0.2328**	**0.2229**	**0.2376**	**0.0987**	0.1216	0.0635	**0.2125**	**0.3176**

Simulations 24h/1h for 5D



Conclusions

We demonstrated how to adopt GNNs for weather nowcasting:

- Spatial features are extremely important in WP and need proper modeling;
- DL methods can successfully model WP for high resolution scales;
- GNN are successful tool to model spatial interactions for WP;

Given the promising preliminary results, possible research directions are:

- GNN capable to interpret longitude, latitude and time at different scales;
- Apply more sophisticated GNN to WP, i.e. GraphTransformer [Dwivedi et al.];
- Explore new types of node relations modelling edge embeddings.

References

- [Xiaoli et al.] Ren, Xiaoli and Li, Xiaoyong and Ren, Kaijun and Song, Junqiang and Xu, Zichen and Deng, Kefeng and Wang, Xiang, *Deep learning-based weather prediction: a survey, Big Data Research, 2021.*
- [Yu et al.] Bing Yu and Haoteng Yin and Zhanxing Zhu, *Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting*, IJCAI, 2018.
- [Veličković et al.] Veličković, Petar and Cucurull, Guillem and Casanova, Arantxa and Romero, Adriana and Liò, Pietro and Bengio, Yoshua, *Graph Attention Networks*, ICLR, 2018.
- [Kipf at al.] *Semi-Supervised Classification with Graph Convolutional Networks*, Thomas N. Kipf and Max Welling, ICLR, 2017.
- [Hersbach et al.] Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., Thépaut, *ERA5 hourly data on single levels from 1940 to present, Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*, 2023.
- [Shi et al.] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, *Convolutional LSTM network: a machine learning approach for precipitation nowcasting*, NIPS, 2015.
- [Shi, Gao et al.] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, W.-c. Woo, *Deep learning for precipitation nowcasting: a benchmark and a new model*, NIPS, 2017.
- [Wang et al.] Y. Wang, M. Long, J. Wang, Z. Gao, S.Y. Philip, *PredRNN: recurrent neural networks for predictive learning using spatiotemporal LSTMs*, NIPS, 2017.
- [Wang, Gao et al.] Y. Wang, Z. Gao, M. Long, J. Wang, P.S. Yu, *PredRNN++: towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning*, PMLR, 2018.
- [Sønderby et al.] C.K. Sønderby, L. Espeholt, J. Heek, M. Dehghani, A. Oliver, T. Salimans, S. Agrawal, J. Hickey, N. Kalchbrenner, *MetNet: a neural weather model for precipitation forecasting*, arXiv, 2020.
- [Agrawal et al.] S. Agrawal, L. Barrington, C. Bromberg, J. Burge, C. Gazen, J. Hickey, *Machine learning for precipitation nowcasting from radar images*, arXiv, 2019.
- [Wang, Hong et al.] C. Wang, Y. Hong, *Application of spatiotemporal predictive learning in precipitation nowcasting*, AGU, 2018.
- [Dwivedi et al.] Vijay Prakash Dwivedi and Xavier Bresson, *A Generalization of Transformer Networks to Graphs*, 2021.
- [Keisler] Ryan Keisler, Forecasting Global Weather with Graph Neural Networks, arXiv, 2022.
- [Lam et al.] Remi Lam and Alvaro Sanchez-Gonzalez and Matthew Willson and Peter Wirnsberger and Meire Fortunato and Alexander Pritzel and Suman Ravuri and Timo Ewalds and Ferran Alet and Zach Eaton-Rosen and Weihua Hu and Alexander Merose and Stephan Hoyer and George Holland and Jacklynn Stott and Oriol Vinyals and Shakir Mohamed and Peter Battaglia, GraphCast: Learning skillful medium-range global weather forecasting, arXiv, 2022.
- [Pfaff et al.] Tobias Pfaff and Meire Fortunato and Alvaro Sanchez-Gonzalez and Peter W. Battaglia, *Learning Mesh-Based Simulation with Graph Networks*, arXiv, 2021.
- [Graves] Alex Graves, Generating Sequences With Recurrent Neural Networks, arXiv, 2014.

Weather Nowcasting with GNNs and LSTM

Course: Theory and practice of Deep Learning
Professor: Fabrizio Silvestri
Sapienza, DIAG, a.y. 2022-2023

Simone Rossetti, PhD student in Engineering in Computer Science