The World as a Graph

Improving El Niño Forecasts with Graph Neural Networks

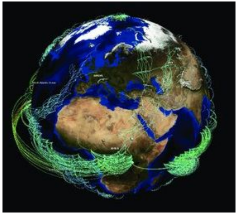
Cachay, Erickson, Bucker, Pokropek, Potosnak, Bire, Lütjens (2021).

Christian Fröhlich
MLCS Journal Club

22 June 2021

Main idea

Graph neural networks for ENSO forecasting

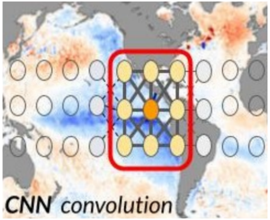


Donges, 2015.

- We want to forecast ENSO.
- Climate networks offer a neat method to represent the world as a graph.
- Can we exploit such graph structure for ENSO forecasting?
- But how to construct the network?
  - In the deep learning spirit: let's just **learn it**.

Why not CNNs?

Convolutional networks achieve SOTA.

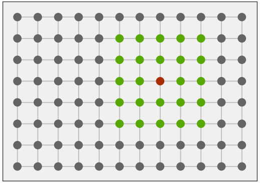


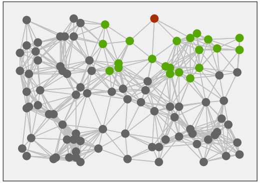
Disadvantages of CNNs for seasonal and long range forecasting

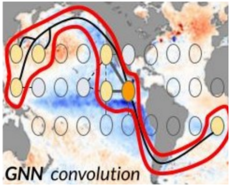
- Translational equivariance.
  - But: location is important.
- Spatial locality bias.
  - But: teleconnections are important.
- CNNs use all grid cells.
  - But: sometimes, only oceanic variables suffice.

Why GNNs?

Graph neural networks. [Credits: Geiger, 2021]







Advantages of GNNs

- Scales better than MLPs.
- More flexible than CNNs.
- More efficient than RNNs.
- Can model teleconnections due to non-Euclidean neighborhoods.
- Improves interpretability (structure encoded in graph).

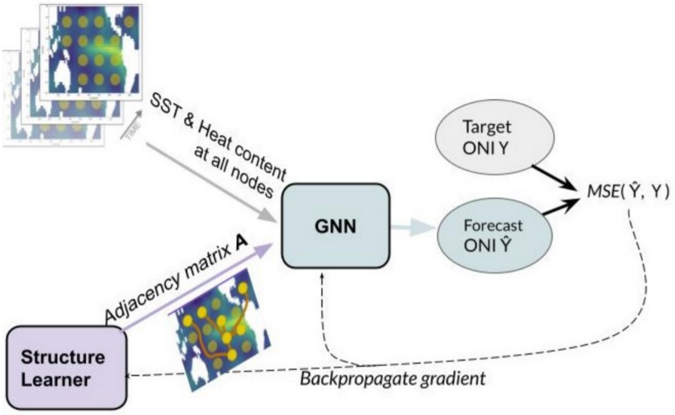
Contributions

A dense paper.

- We propose the **first application of GNNs** to long range and seasonal forecasting.
- Building upon established previous research we develop and **open-source Graphino**, a flexible graph convolutional network architecture for long range forecasting applications in the climate and earth sciences.
- We introduce a novel **graph structure learning module**, which makes our model applicable even **without a predefined connectivity structure**.
- We show that our model is competitive to state-of-the-art statistical and dynamical ENSO forecasting systems, and **outperforms** them for forecasts of **up to six months**.
- We exploit our model's **interpretability**, to show how it learns sensible connections that are consistent with existing theories on ENSO dynamics predictability.

Overview

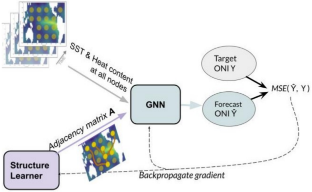
of the structure.



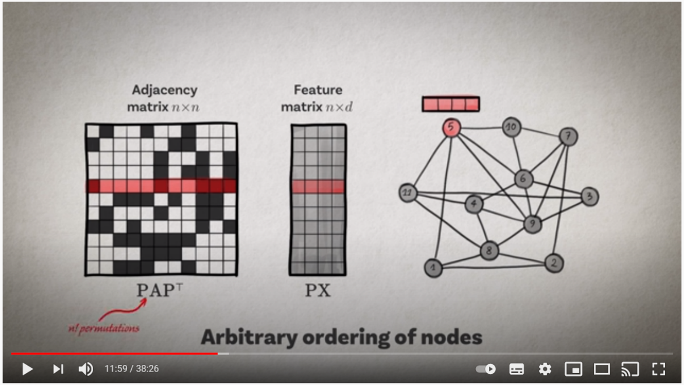
- Goal: Forecast Oceanic Niño Index (ONI) for a fixed lead time.

Problem Setup

The formalities.



- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each $V_i \in \mathcal{V}$, $1 \leq i \leq N$, is a node of a gridded climate dataset.
- For each time $t = 1..T$, node feature vector $\mathbf{V}_i^t \in \mathbb{R}^D$ of climatic variable.
- Adjacency $\mathbf{A} \in \{0,1\}^{N \times N}$ with $(i,j) \in \mathcal{E} \Leftrightarrow \mathbf{A}_{ij} = 1$.
- Snapshot measurement $\mathbf{X}_t = (\mathbf{V}_1^t, ..\mathbf{V}_n^t) \in \mathbb{R}^{N \times D}$
- For window size w, concatenate to obtain $\mathbf{X} = hstack(\mathbf{X}_{\mathbf{t_1}}, ..\mathbf{X}_{t_w}) \in \mathbb{R}^{N \times wD}$.
- Target $Y = Y_{t_w+h} \in \mathbb{R}$, ONI index for lead time h.
- Loss $\mathcal{L}$: MSE.



ICLR 2021 Keynote - "Geometric Deep Learning: The Erlangen Programme of ML" - M Bronstein

If you haven't yet, watch it!

- Node embeddings $\mathbf{Z}_i^l$ for layer l and node i, set $\mathbf{Z}^0 = \mathbf{X}$.
- Next layer: $\mathbf{Z}^l = \sigma\left(\mathbf{A}\mathbf{Z}^{l-1}\mathbf{W}^l\right) \in \mathbb{R}^{N \times D_l}$.
- For continuous $\mathbf{A}$, this is a weighted sum inside the sigmoid.
- Aggregate output of last layer L to obtain graph embedding: $\mathbf{g} = \mathit{Aggregate}\left(\mathbf{Z}_1^L, .., \mathbf{Z}_N^L\right) \in \mathbb{R}^{D_L}$.
- Finally, use an MLP to forecast ONI: $\hat{Y} = \mathit{MLP}(\mathbf{g})$.

Implementation details

Deep Learning = bag of tricks.

- GCNs are typically shallow, in this case 2 and 3 layers.
- Followed by 2 layer MLP.
- Batch normalization, no dropout.
- Residual connections and jumping knowledge.
- Aggregation functions: mean, sum.

The Structure Learner

A data-driven adjacency.

- To obtain the adjacency $\mathbf{A}$, use static node representations $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times \tilde{d}_1}$.
- $\tilde{\mathbf{X}}$: SST, heat content anomalies, latitude and longitudes.

$$\mathbf{M}_1 = \tanh\left(\alpha_1 \tilde{\mathbf{X}} \tilde{\mathbf{W}}_1\right) \in \mathbb{R}^{N \times \tilde{d}_2}, \tag{1}$$

$$\mathbf{M}_2 = \tanh\left(\alpha_1 \tilde{\mathbf{X}} \tilde{\mathbf{W}}_2\right) \in \mathbb{R}^{N \times \tilde{d}_2}, \tag{2}$$

$$\mathbf{A} = \mathrm{sigmoid}\left(\alpha_2 \mathbf{M}_1 \mathbf{M}_2^\intercal\right) \in \{0, 1\}^{N \times N}. \tag{3}$$

- α_1, α_2 hyperparameters controlling the spread of values and confidence in edges.
- Finally, set all but largest e edge weights to 0 to enforce desired sparsity.
- Add self-loops to the graph.

Experiments

Data

- SODA reanalysis dataset (1871-1973).
- Climate model simulations from CMIP5.
  - Augmentation is needed for deep learning.
- Test set: GODAS dataset (1984-2017).
- Grid resolution 5 degrees, locations in $55S - 60N$ and $0 - 360W$.
- $N = 1345$ nodes after filtering out terrestrial ones.
- Features: SST and heat content anomalies, window $w = 3$ months.

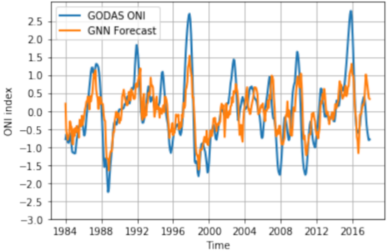
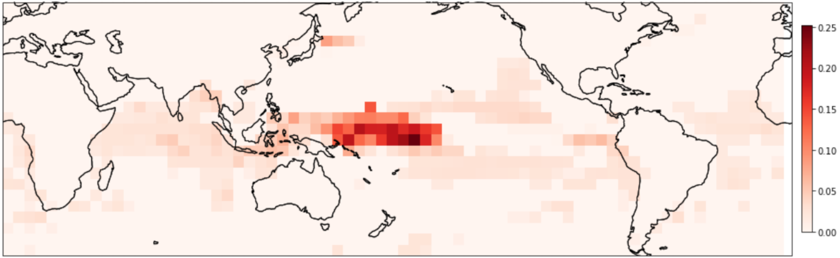


Figure: 6 month lead predictions.

- Outperforms state-of-the-art CNN of for up to 6 lead months
- Outperforms the competitive dynamical model SINTEX-F for all lead times.
- Why the decrease in performance for more than six lead months?
- Hypothesis: learning connectivity structure makes the model more prone to overfitting.

Interpretability

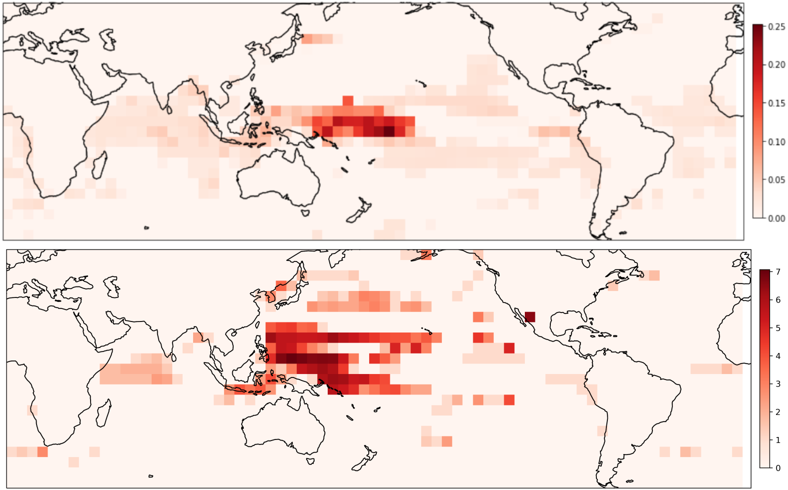
We can analyze the graph!



- The authors employ eigenvector centrality to visualize connectivity.
- But: importance $\neq$ centrality.

Interpretability

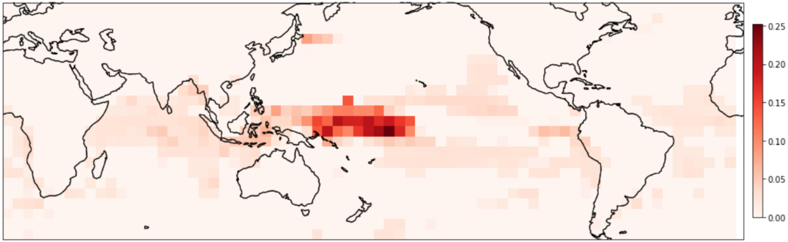
Most Positive Ollivier Ricci Curvature

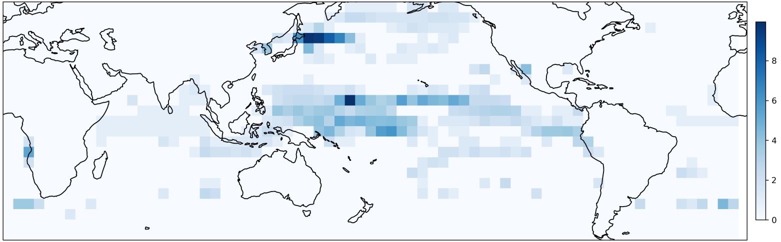


- Top: Eigenvector centrality. Bottom: nodes with top 10% positive edges.

Interpretability

Most Negative Ollivier Ricci Curvature

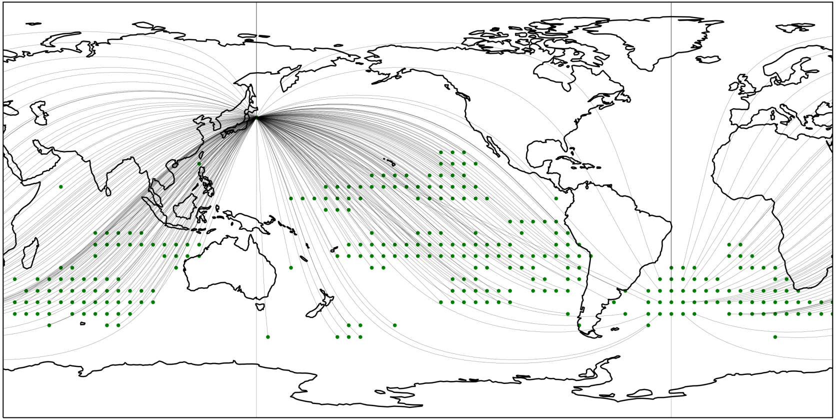




- Top: Eigenvector centrality. Bottom: nodes with top 10% negative edges.

Interpretability

Edges Connected to Node with Highest Negative Unnormalized Ollivier Ricci Curvature



Interpretability

Edges Connected to Node with Highest Eigenvector Centrality