On the Covid 19 prediction in Brazil

Felipe Schreiber Fernandes
felipesc@cos.ufrj.br

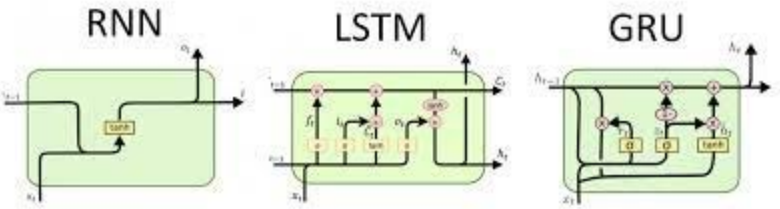
19 de Junho de 2023

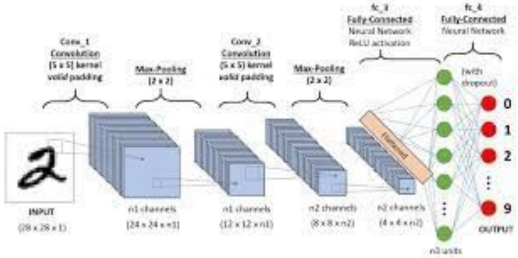
Engenharia de Sistemas e Computação
PESC/Coppe

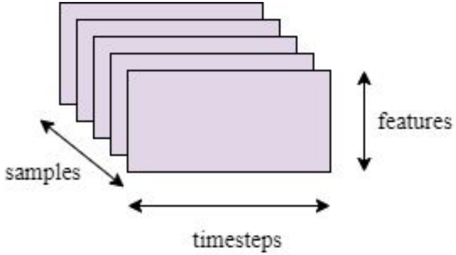
Problem Statement

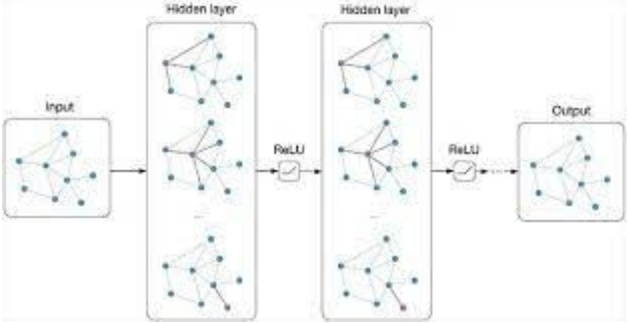
Hence the full problem statement can be viewed as follows: Given a graph $G = (V, E, W)$, and a signal $X \in R^{|V| x d}$, the objective is to learn a function that given T' historical graph signals to future T graph signals, given a graph G:

$$\left[X^{t-T'+1}, \ldots, X^{t}\right] \xrightarrow{h(.)} \left[X^{t+1}, \ldots, X^{t+T}\right] \qquad (1)$$









Dataset

Engenharia de Sistemas e Computação PESC/Coppe

PyTorch Geometric Temporal

The final dataset hence contains 5570 nodes, 32944 weighted edges with 761 observations

Related Works

Yaguang Li et al. *Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting*. 2018. arXiv: 1707.01926 [cs.LG].

A. Spatial Dependency

weighted combination of infinite random walks on the graph, and be calculated in closed form:

$$\mathcal{P} = \sum_{k=0}^{\inf} \alpha(1-\alpha)^k (D^{-1}W)^k \quad (2)$$

[10] Shang-Hua Teng. "Scalable Algorithms for Data and Network Analysis". In: *Found. Trends Theor. Comput. Sci.* 12.1–2 (May 2016), pp. 1–274. ISSN: 1551-305X. DOI: 10.1561/0400000051. URL: https://doi.org/10.1561/0400000051.

B. Diffusion Convolution

$$X * f_{G(V,E)}(\theta) = \sum_{k=0}^{\inf} \theta_k (D^{-1}W)^k X \quad (3)$$

$$H = \sigma\left(\sum_{d} X * f_{G(V,E)}(\Theta_{m,n})\right) \quad (4)$$

[1] James Atwood and Don Towsley. *Diffusion-Convolutional Neural Networks*. 2016. arXiv: 1511.02136 [cs.LG].

Engenharia de Sistemas e Computação PESC/Coppe

Related Works

Yaguang Li et al. *Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting*. 2018. arXiv: 1707.01926 [cs.LG].

C. Temporal Dependency

$$z_t = \sigma(W^{(z)} X_t + U^z h_{t-1}) \quad (5)$$

$$r_t = \sigma(W^{(r)} X_t + U^r h_{t-1}) \quad (6)$$

$$h_t' = \tanh(W^h X_t + r_t \odot U h_{t-1}) \quad (7)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t' \quad (8)$$

$$u^t = \sigma(\Theta_{u*G}[X^t, H^{(t-1)}] + b_u) \quad (9)$$

$$r^t = \sigma(\Theta_{r*G}[X^t, H^{(t-1)}] + b_r) \quad (10)$$

$$C^t = \tanh(\Theta_{C*G}[X^t, r^{(t-1)} \odot H^{t-1}] + b_c) \quad (11)$$

$$H^t = u^t \odot H^{t-1} + (1 - u^t) \odot C^t \quad (12)$$

Lei Bai et al. *Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting*. 2020. arXiv: 2007.02842 [cs.LG].

(DAGG) module to infer inter-dependencies. First, they use the findings of [5] that graph convolutions can be approximated by 1^{st} order Chebyshev polynomial expansion:

$$Z = (I_N + D^{-1/2} A D^{-1/2}) X \Theta + b \quad (13)$$

$\Theta \in R^{NxCxF} \approx EW$, where $E \in R^{Nxd}$ are the node embeddings and $W \in R^{dxCxF}$ is a learnable matrix. This insight leads the reformulation of equation 13 as:

$$Z = (I_N + D^{-1/2} A D^{-1/2}) X E W + E b \quad (14)$$

In order to let the adjacency matrix to be learned, the following trick is applied:

$$D^{-1/2} A D^{-1/2} = \text{softmax}(\text{ReLU}(E E^T)) \quad (15)$$

Related Works

Lei Bai et al. *Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting*. 2020. arXiv: 2007.02842 [cs.LG].

$$\tilde{A} = \text{softmax}(\text{ReLU}(EE^T)) \quad (16)$$

$$z_t = \sigma(\tilde{A}[X_{:,t}, h_{t-1}]EW_z + Eb_z) \quad (17)$$

$$r_t = \sigma(\tilde{A}[X_{:,t}, h_{t-1}]EW_r + Eb_r) \quad (18)$$

$$h_t' = \tanh(\tilde{A}[X_{:,t}, r_t \odot h_{t-1}]EW_h + Eb_h) \quad (19)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t' \quad (20)$$

[12] Ling Zhao et al. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction". In: *IEEE Transactions on Intelligent Transportation Systems* 21.9 (Sept. 2020), pp. 3848–3858. DOI: 10.1109/tits.2019.2935152. URL: https://doi.org/10.1109%2Ftits.2019.2935152.

Another paper that tackles the problem of traffic forecasting is the Temporal Graph Convolutional Network [12], which replaces the X_t in GRU equations by the result of a two layer GCN:

$$f(X, A) = \sigma(\tilde{A}\text{ReLU}(\tilde{A}XW_0)W_1) \quad (21)$$

[8] Youngjoo Seo et al. *Structured Sequence Modeling with Graph Convolutional Recurrent Networks*. 2016. arXiv: 1612.07659 [stat.ML].

[9] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. *Sequence to Sequence Learning with Neural Networks*. 2014. arXiv: 1409.3215 [cs.CL].

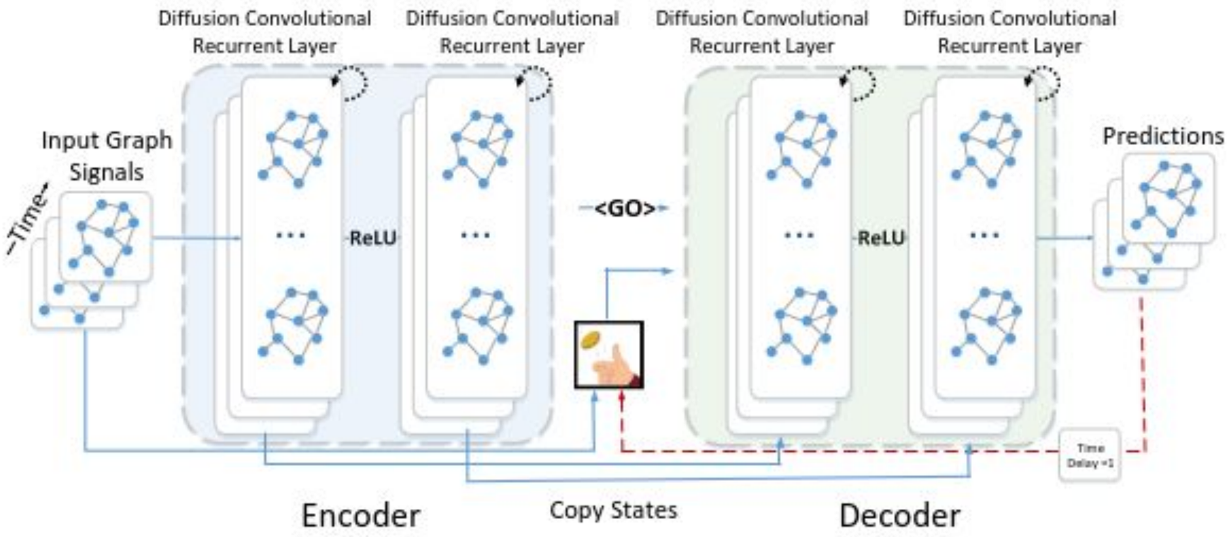
significantly outperformed shallow LSTMs, so we chose an LSTM with four layers. Third, we found it extremely valuable to reverse the order of the words of the input sentence. So for example, instead of mapping the sentence a, b, c to the sentence α, β, γ, the LSTM is asked to map c, b, a to α, β, γ, where α, β, γ is the translation of a, b, c. This way, a is in close proximity to α, b is fairly close to β, and so on, a fact that makes it easy for SGD to "establish communication" between the input and the output. We found this simple data transformation to greatly improve the performance of the LSTM.

Engenharia de Sistemas e Computação PESC/Coppe

Related Works

Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, Noam Shazeer



Proposed Model

Engenharia de Sistemas e Computação PESC/Coppe

$$X * f_{G(V,E)}(\theta) = \sum_{k=0}^{\inf} \theta_k (D^{-1}W)^k X \qquad (3)$$

$$H = \sigma\left(\sum_{d} X * f_{G(V,E)}(\Theta_{m,n})\right) \qquad (4)$$

IV. Proposed Model

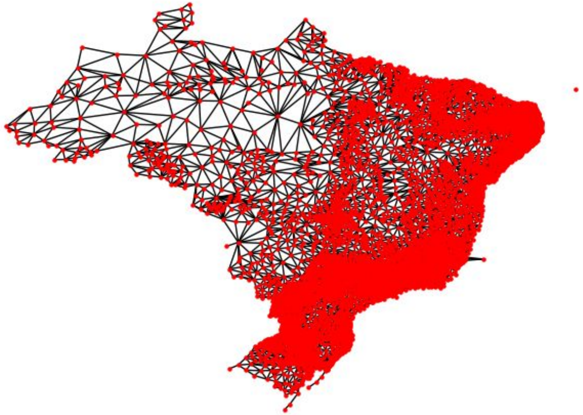
From TGCN, DCRNN and AGCRN, we can derive three improvements. First is to replace the $D^{-1}W$ component of DCRNN by the trick softmax(PReLU(EE^T)), in other words we're letting the model learn the underlying diffusion network. Second, we could potentially replace X in DCRNN by the output of two GIN layers. Third, in the spirit of AGCRN, we could add the terms EW to factorize Θ. However, only the first insight was implemented in the following tests, leading to the Adaptive Diffusion Convolutional Recurrent Neural Network.

Methodology

class ChickenpoxDatasetLoader [source]

A dataset of county level chicken pox cases in Hungary between 2004 and 2014. We made it public during the development of PyTorch Geometric Temporal. The underlying graph is static - vertices are counties and edges are neighbourhoods. Vertex features are lagged weekly counts of the chickenpox cases (we included 4 lags). The target is the weekly number of cases for the upcoming week (signed integers). Our dataset consist of more than 500 snapshots (weeks).

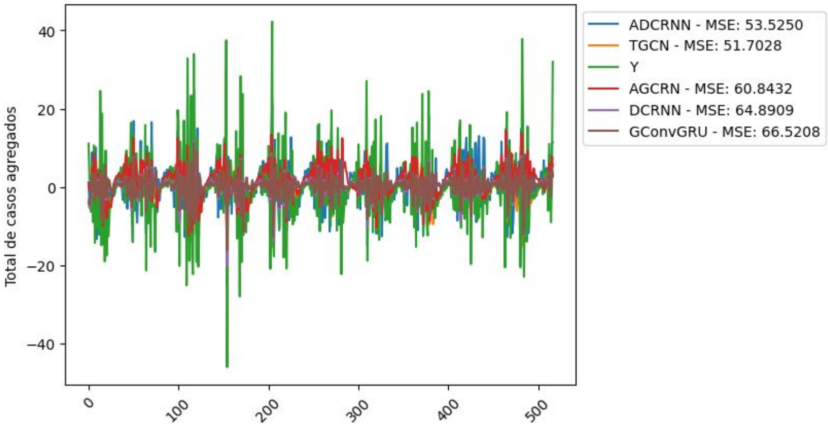
Systematic comparison

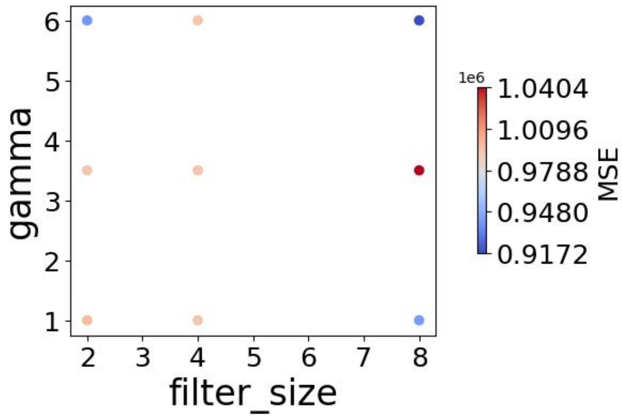


DCRNN X ADCRNN
filter sizes and gamma

$$L(Y,\hat{Y},\gamma,\tilde{A}) = \sum(Y^t - \hat{Y}^t)^2 + \gamma||\tilde{A}||_1 \qquad (22)$$

Results





K	MSE
2	83165357.15
4	85851151.47
8	87918307.10
16	91113848.26

TABLE I

EFFECT OF FILTER SIZE OF DCRNN IN COVID 19 DATASET