GRAPH CONVOLUTIONAL RECURRENT NETWORK FOR STRUCTURED SEQUENCE MODELING

Youngjoo Seo
EPFL, Lausanne, Switzerland
youngjoo.seo@epfl.ch

Michaël Defferrard
EPFL, Lausanne, Switzerland
michael.defferrard@epfl.ch

Pierre Vandergheynst
EPFL, Lausanne, Switzerland
pierre.vandergheynst@epfl.ch

Xavier Bresson
Nanyang Technological University, Singapore
xavier.bresson@ntu.edu.sg

ABSTRACT

This paper introduces and studies the Graph Convolutional Recurrent Network, a Deep Learning model able to represent and predict structured sequences. It is a generalization of classical recurrent neural networks to data structured by a weighted graph. Such structured sequences can be e.g. videos, a spatio-temporal sequence where the structuring graph is a 2D grid, measurements on a network of sensors or random walks on a vocabulary graph for language modeling. This work studies two possible architectures and apply the model to two practical problems: a benchmark moving MNIST dataset and a language model on the Penn Treebank. Experiments show that our network

1 INTRODUCTION

Why study graph-structured sequences: capture statistical properties in the joint domain.

Which real-world problems can be casted in that setting.

Shi et al. have proposed an LSTM model for spatio-temporal sequence modeling which uses 2D convolutions to leverage the spatial correlations in the input data, where the spatial structure is given by a 2D grid. This work is a direct extension of theirs to the more general setting where the structure is given by a weighted graph. This work builds on our recently proposed graph ConvNet framework (Defferrard et al., 2016) and apply it to time-varying graph signals.

2 PRELIMINARIES

2.1 SEQUENCE MODELING

Sequence modeling is the problem of predicting the most likely next element $\hat{x}_t \in \mathbf{R}^{d_x}$ given a sequence of previous observations $(x_0, \dots, x_{t-1})$:

$$\hat{x}_t = \arg\max_{x_t \in \mathbf{R}^{d_x}} P(x_t | x_{t-1}, \dots, x_1), \tag{1}$$

where $\mathbf{R}^{d_x}$ denotes the domain of the observed features. The archetypal application being the language model (Graves, 2013).

In this paper, we are interested in structured sequences, i.e. sequences $(x_1, \dots, x_t)$ where the elements of x are not independent but linked by pairwise relationships. We propose a generalization to any structure who can be modeled with graphs, which are universal representations of heterogeneous pairwise relationships.

structured sequence model (2)

2.2 LONG SHORT-TERM MEMORY

Long short-term memory (LSTM), a recurrent neural network (RNN) architecture introduced by Hochreiter & Schmidhuber (1997) designed to prevent the gradient from vanishing too quickly, has proven stable and powerful for modeling long-range dependencies in various general-purpose sequence modeling tasks (Graves, 2013; Sutskever et al., 2014). A fully-connected LSTM (FC-LSTM) may be seen as a multivariate version of LSTM where the input $x_t \in \mathbb{R}^{d_x}$, cell output $h_t \in [-1, 1]^{d_h}$ and states $c_t \in \mathbb{R}^{d_h}$ are all vectors. In this paper, we follow the FC-LSTM formulation of Shi et al., that is:

$$\begin{aligned} i &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + w_{ci} \odot c_{t-1} + b_i), \\ f &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + w_{cf} \odot c_{t-1} + b_f), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \\ o &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + w_{co} \odot c_t + b_o), \\ h_t &= o \odot \tanh(c_t), \end{aligned} \tag{3}$$

where $\odot$ denotes the Hadamard product, $\sigma(\cdot)$ the sigmoid function $\sigma(x) = 1/(1+e^{-x})$ and $i, f, o \in [0, 1]^{d_h}$ are the input, forget and output gates. The weight matrices $W_{x\cdot} \in \mathbb{R}^{d_h \times d_x}$, $W_{h\cdot} \in \mathbb{R}^{d_h \times d_h}$, weight vectors $w_{c\cdot} \in \mathbb{R}^{d_h}$ and biases $b_i, b_f, b_c, b_o \in R^{d_h}$ are the model parameters.[1] Such a model is called fully-connected because the dense matrices W linearly combine all the components of x and h. The optional peephole connections $w_{c\cdot} \odot c_t$, introduced by Gers & Schmidhuber (2000), have been found to improve performance for certain tasks.

2.3 CONVOLUTIONAL NEURAL NETWORKS ON GRAPHS

We are interested in processing graph-structured sequences, i.e. signals defined on undirected and connected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is a finite set of $|\mathcal{V}| = n$ vertices, $\mathcal{E}$ is a set of edges and $A \in \mathbb{R}^{n \times n}$ is a weighted adjacency matrix encoding the connection weight between two vertices. A signal $x_t : \mathcal{V} \to \mathbb{R}^{d_x}$ defined on the nodes of the graph may be regarded as a matrix $x_t \in \mathbb{R}^{d_x \times n}$ where $x_i \in \mathbb{R}^{d_x}$ is the value of x at the i^{th} node and d_x is the number of features.

Generalizing convolutional neural networks (CNNs) to arbitrary graphs is a recent area of interest. The proposed methods take two approaches: (i) an application of the spatial definition of a convolution and (ii), using the convolution theorem, a multiplication in the Fourier domain. introduce the new graph CNNs and motivate our choice In this work we focus on the framework introduced by Defferrard et al. (2016), but note that the proposed model is agnostic to the choice of the graph convolution operator $*_{\mathcal{G}}$.

As it is difficult to express a meaningful translation operator in the vertex domain, Defferrard et al. (2016) chose a spectral formulation for the convolution operator on graph $*_{\mathcal{G}}$. Given this definition, a graph signal x is filtered by a non-parametric kernel $g_\theta(\Lambda) = \mathrm{diag}(\theta)$, where $\theta \in \mathbb{R}^n$ is a vector of Fourier coefficients, as

$$y = g_\theta *_{\mathcal{G}} x = g_\theta(L)x = g_\theta(U\Lambda U^T)x = U g_\theta(\Lambda)U^T x, \tag{4}$$

where U is the matrix of eigenvectors and Λ the diagonal matrix of eigenvalues of the normalized graph Laplacian $L = I_n - D^{-1/2}AD^{-1/2} = U\Lambda U^T$, where I_n is the identity matrix and $D \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. The graph Fourier transform of x is given by $U^T x$ (Shuman et al., 2013). Evaluating (4) is however expensive, as the multiplication with U is $\mathcal{O}(n^2)$. Furthermore, computing the eigendecomposition of L might be prohibitively expensive for large graphs. To circumvent this problem, Defferrard et al. (2016) parametrizes g_θ as a truncated expansion, up to order $K - 1$, of Chebyshev polynomials T_k such that

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}), \tag{5}$$

where the parameter $\theta \in \mathbb{R}^K$ is a vector of Chebyshev coefficients and $T_k(\tilde{\Lambda}) \in \mathbb{R}^{n \times n}$ is the Chebyshev polynomial of order k evaluated at $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I_n$. The filtering operation can then

[1] A practical trick is to initialize the biases b_i, b_f and b_o to one such that the gates are initially open.

be written as

$$y = g_\theta *_\mathcal{G} x = g_\theta(L)x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x, \tag{6}$$

where $T_k(\tilde{L}) \in \mathbb{R}^{n \times n}$ is the Chebyshev polynomial of order k evaluated at the scaled Laplacian $\tilde{L} = 2L/\lambda_{max} - I_n$. Using the stable recurrence relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$, one can evaluate (6) in $\mathcal{O}(K|\mathcal{E}|)$ operations, i.e. linear in the number of edges. The reader is referred to Defferrard et al. (2016) for details and an in-depth discussion. Note that the filtering operation (6), as it is an order K polynomial of the Laplacian, is K-localized, i.e. it depends only on nodes that are at maximum K hops away from the central node, the K-neighborhood.

3 Related Works

Convolutional LSTM (convLSTM). Shi et al. introduced a model for grid-structured data, which can be seen as a special case of (2) where the graph is a grid and the nodes are ordered. Their model is essentially the classical FC-LSTM where the multiplication by dense matrices W in (3) have been replaced by convolutions with kernels W:

$$\begin{aligned}
i &= \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + w_{ci} \odot c_{t-1} + b_i), \\
f &= \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + w_{cf} \odot c_{t-1} + b_f), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c), \\
o &= \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + w_{co} \odot c_t + b_o), \\
h_t &= o \odot \tanh(c_t),
\end{aligned} \tag{7}$$

where $*$ denotes the 2D convolution by a set of kernels. In that setting, the input tensor $x_t \in \mathbb{R}^{d_x \times n_r \times n_c}$ is the observation of d_x measurements at time t of a dynamical system over a spatial region represented by a grid of n_r rows and n_c columns. The model holds spatially distributed hidden and cell states of size d_h given by the tensors $c_t, h_t \in \mathbb{R}^{d_h \times n_r \times n_c}$. The size m of the convolutional kernels $W_{h\cdot} \in \mathbb{R}^{d_h \times d_h \times m \times m}$ and $W_{x\cdot} \in \mathbb{R}^{d_h \times d_x \times m \times m}$ determines the number of parameters, which is independent of the grid size $n_r \times n_c$.

They claim that "although the number of free variables in a length-K sequence can be up to $\mathcal{O}(M^K N^K P^K)$, in practice we may exploit the structure of the space of possible predictions to reduce the dimensionality and hence make the problem tractable."

Ad-hoc models. Alternative models have been proposed in ... Describe and say why they are ad-hoc to the problem.

4 Proposed Model

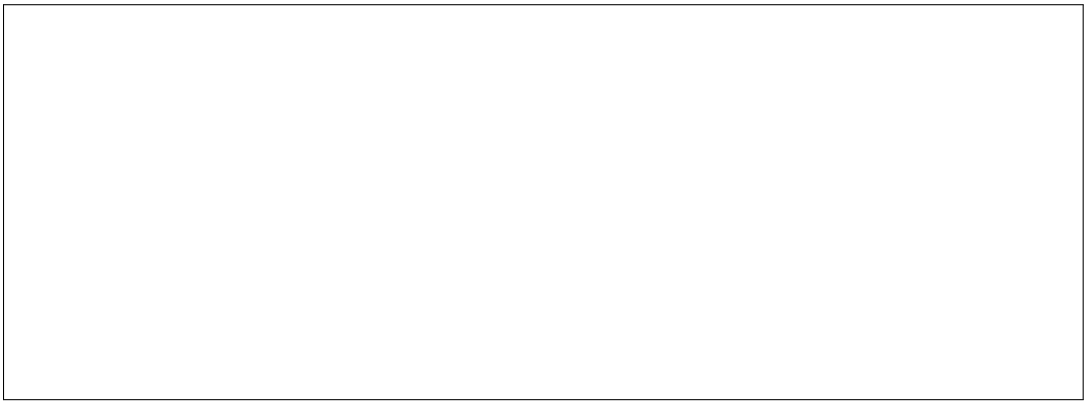

Figure 1: Nice illustration.

We propose to take the convLSTM model (7) and replace the 2D convolution $*$ by the generalized graph convolution $*_{\mathcal{G}}$, as defined in (6), such that:

$$\begin{aligned} i &= \sigma(W_{xi} *_{\mathcal{G}} x_t + W_{hi} *_{\mathcal{G}} h_{t-1} + w_{ci} \odot c_{t-1} + b_i), \\ f &= \sigma(W_{xf} *_{\mathcal{G}} x_t + W_{hf} *_{\mathcal{G}} h_{t-1} + w_{cf} \odot c_{t-1} + b_f), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc} *_{\mathcal{G}} x_t + W_{hc} *_{\mathcal{G}} h_{t-1} + b_c), \\ o &= \sigma(W_{xo} *_{\mathcal{G}} x_t + W_{ho} *_{\mathcal{G}} h_{t-1} + w_{co} \odot c_t + b_o), \\ h_t &= o \odot \tanh(c_t). \end{aligned} \tag{8}$$

In that setting, the input matrix $x_t \in \mathbb{R}^{d_x \times n}$ may represent the observation of d_x measurements at time t of a dynamical system over a network of $n = |V|$ sensors whose organization is given by the weighted graph $\mathcal{G}$. The model holds spatially distributed hidden and cell states of size d_h given by the matrices $c_t, h_t \in \mathbb{R}^{d_h \times n}$. The support K of the graph convolutional kernels $W_{h\cdot} \in \mathbb{R}^{d_h \times d_h \times K}$ and $W_{x\cdot} \in \mathbb{R}^{d_h \times d_x \times K}$ determines the number of parameters, which is independent of the number of nodes n. In a distributed computing setting, K controls the communication overhead, i.e. the number of nodes a node i should exchange with in order to compute its local states.

The proposed blend of RNNs and graph CNNs is not limited to LSTMs and is straightforward to apply to any kind of recursive networks. For example, a vanilla RNN $h_t = \tanh(W_x x + W_h h)$ would be modified as

$$h_t = \tanh(W_x *_{\mathcal{G}} x_t + W_h *_{\mathcal{G}} h_{t-1}), \tag{9}$$

and a Gated Recurrent Unit (GRU) (Cho et al., 2014) as

$$\begin{aligned} z &= \sigma(W_{xz} *_{\mathcal{G}} x_t + W_{hz} *_{\mathcal{G}} h_{t-1}), \\ r &= \sigma(W_{xr} *_{\mathcal{G}} x_t + W_{hr} *_{\mathcal{G}} h_{t-1}), \\ \tilde{h} &= \tanh(W_{xh} *_{\mathcal{G}} x_t + W_{hh} *_{\mathcal{G}} (r \odot h_{t-1})), \\ h_t &= z \odot h_{t-1} + (1 - z) \odot \tilde{h}. \end{aligned} \tag{10}$$

As demonstrated by Shi et al., structure-aware LSTM cells can be stacked and used for sequence-to-sequence models using an architecture composed of an encoder (which processes the input sequence) and a decoder (which generates an output sequence), a standard practice for machine translation using RNNs (Cho et al., 2014; Sutskever et al., 2014).

5 Experiments

Proposing Graph Convolutional Recurrent Neural Networks(GCRNN) applies two benchmark datasets: moving MNISTSrivastava et al. (2015) and Penn Tree BankMarcus et al. (1993). Experiments are mainly focuses on how the graph convolution give effects on RNN model by comparing conventional RNN models with GCRNN. In moving MNIST experiments, we shows 2D grid structure is one of special case of graph structure which GCRNN can easily fit to the model as convolutional RNN, while Language modeling on Penn Tree Bank demonstrates graph structure on words helps

5.1 Moving MNIST

empirical proof that graph RNN works on grid structured data

Following the experimental setup of Shi et al..

Our first experiment on the moving MNIST dataset Srivastava et al. (2015) shows the ability of our model (8) to learn spatio-temporal structures.

moving MNIST: similar as (7) convLSTM because of lack of orientation / node ordering

rotating MNIST: better thanks to that property

5.2 Meteorological Prediction

network of sensors (with local processing): model aggregates information from local neighborhood, good for distributed processing.

5.3 LANGUAGE MODELING ON PENN TREEBANK

Uses (9) followed by a softmax layer.

6 CONCLUSION AND FUTURE WORK

We introduced the Graph Convolutional Recurrent Network, an architecture designed to model graph-structured and time-varying data. The model is an extension of Shi et al. which leverages recent advances in graph ConvNets (Defferrard et al., 2016). Numerical experiments have shown the capability of the model to ... Future works will both refine the model with advances in graph ConvNets, a recent area of interest, and apply the model to real-world problems.

REFERENCES

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 30*. 2016.

Felix A Gers and Jürgen Schmidhuber. Recurrent nets that time and count. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pp. 189–194. IEEE, 2000.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems 28*, pp. 802–810.

D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and other Irregular Domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 843–852, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.