PartIV: Graph Attention Networks and Spatial-temporal Graph Networks

Haoxuan Sun

Center for Data Science
Peking University

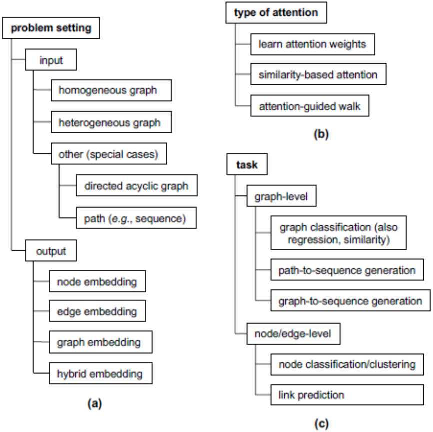
May 26, 2020

Contents

Attention Models in Graphs

Proposed taxonomies to group graph attention model based on

- (a) problem setting
- (b) type of attention used
- (c) task or problem



John Boaz Lee et al. "Attention models in graphs: A survey". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13.6 (2019), pp. 1–25

Attention Models in Graphs

Another taxonomy group graph attention models by the definition of attention.

- Over edges: GAT, GaAN, hGAO and cGAO, AGNN
- Over nodes: GAM

Boris Knyazev, Graham W Taylor, and Mohamed Amer. "Understanding Attention and Generalization in Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4204–4214

Graph Attention Networks

- Input: Homogeneous graph
- Output: Node embedding
- Mechanism: Learning attention weights defined over edges
- Task: Node/Link classification

Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017)

Graph Attention Networks

Attention mechanism $a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$

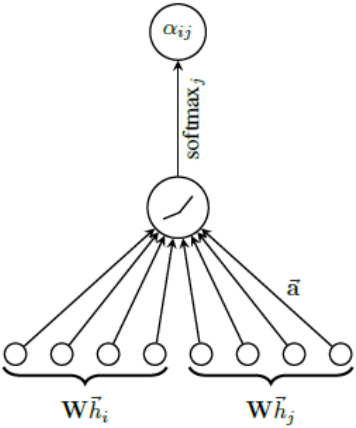
- A set of node features

  $$h = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$$

  Parametrizd by a weight matrix
  $W \in \mathbb{R}^{F' \times F}$

- Single-layer feedforward neural network
  Parametrized by a weight vector
  $\vec{a} \in \mathbb{R}^{2F'}$

$$\alpha_{ij} = \frac{\exp\left(\text{LeakReLU}\left(\vec{a}^T[\mathrm{W}\vec{h}_i \| \mathrm{W}\vec{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakReLU}\left(\vec{a}^T[\mathrm{W}\vec{h}_i \| \mathrm{W}\vec{h}_j]\right)\right)}$$

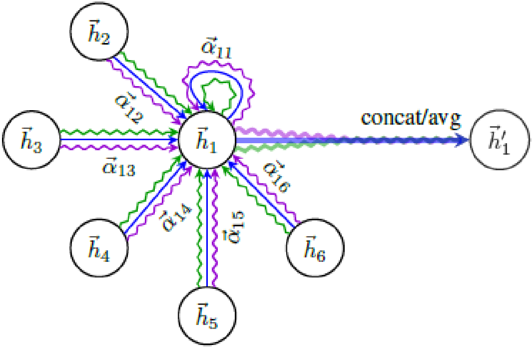


Graph Attention Networks

Multi-head attention

- Stabilize the learning process of self-attention
- Concatenate the feathers
- Specially, employing average on the final layer

$$\vec{h}_i' = \|_{k=1}^{K} \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathrm{W}^k \vec{h}_j \right)$$



Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017)

Graph Attention Networks

- Highly efficient on computation: $O(|V|FF' + |E|F')$ time complexity
- Allow for assigning different importances to nodes of a same neighborhood, enabling a leap in model capacity
- Independ on upfront access to the global graph structure or all of its nodes
- Work with the entirety of the neighborhood and does not assume any ordering within it.

Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017)

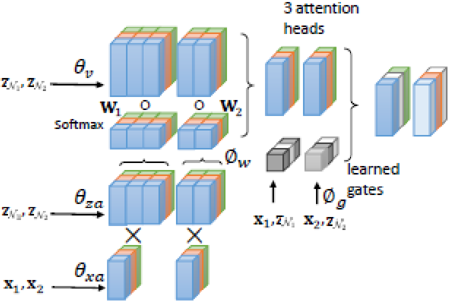
Gated Attention Networks

$$\mathrm{y_i} = \mathrm{FC}_{\theta_o}\left(\mathrm{x_i} \oplus \|_{k=1}^{K} \left(g_i^{(k)} \sum_{j \in \mathcal{N}_i} w_{i,j}^{(k)} \mathrm{FC}^{h}_{\theta_v^{(k)}}(\mathrm{z_j})\right)\right)$$

$$\mathrm{g_i} = [g_i^{(1)}, \ldots, g_i^{(K)}] = \phi_g(\mathrm{x_i}, \mathrm{z}_{\mathcal{N}_i})$$

$$\mathrm{g_i} = \mathrm{FC}^{\sigma}_{\theta_g}\left(\mathrm{x_i} \oplus \max_{j \in \mathcal{N}_j}\left(\{\mathrm{FC}_{\theta_m}(\mathrm{z_j})\}\right) \oplus \frac{\sum_{j \in \mathcal{N}_i} \mathrm{z_j}}{|\mathcal{N}_i|}\right)$$

Gate: combine max pooling and average pooling



Jiani Zhang et al. "Gaan: Gated attention networks for learning on large and spatiotemporal graphs". In: *arXiv preprint arXiv:1803.07294* (2018)

Gated Attention Networks

- Using a small convolutional subnetwork to compute a soft gate at each attention head to control its importance.
- Modulating the amount of attended content via the gates.
- Easy to train: only a simple and light-weighted subnetwork is introduced in constructing the gates

Jiani Zhang et al. "Gaan: Gated attention networks for learning on large and spatiotemporal graphs". In: *arXiv preprint arXiv:1803.07294* (2018)

Hard graph attention operator

For all nodes in graph, use a projection vector $\mathrm{p} \in \mathbb{R}^d$ to select the k-most important nodes to attend:

$$
\begin{aligned}
& y = \frac{|X^T p|}{|p|} && \in \mathbb{R}^N \\
& \textbf{for} \, i = 1, 2, \ldots, N \text{do} && \\
& \qquad \text{idx}_i = Ranking(A_{:i} \circ y) && \in \mathbb{R}^k \\
& \qquad \hat{X}_i = X(:, \text{idx}_i) && \in \mathbb{R}^{d \times k} \\
& \qquad \tilde{y}_i = \text{sigmoid}(y(\text{idx}_i)) && \in \mathbb{R}^k \\
& \qquad \tilde{X}_i = \hat{X}_i \text{diag}(\tilde{y}_i) && \in \mathbb{R}^{d \times k} \\
& \qquad z_i = \text{attn}(x_i, \tilde{X}_i, \tilde{X}_i) && \in \mathbb{R}^d \\
& Z = [z_1, z_2, \ldots, z_N] && \in \mathbb{R}^{d \times N}
\end{aligned}
$$

Hongyang Gao and Shuiwang Ji. "Graph representation learning via hard and channel-wise attention networks". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 741–749.

Hard graph attention operator

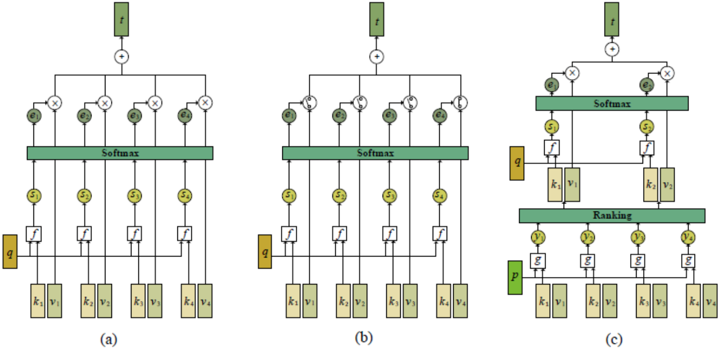


Illustration of GAO (a), another hard attention operator (b), and hGAO (c)

Hongyang Gao and Shuiwang Ji. "Graph representation learning via hard and channel-wise attention networks". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 741–749.

Channel-wise graph attention operator

- For each channel $X_{i:}$, compute its responses by attending it to all channels
- The layer-wise forward propagation function can be expressed as

$$E = XX^T \qquad \in \mathbb{R}^{d\times d}$$

$$O = \text{softmax}(E)X \qquad \in \mathbb{R}^{d\times N}$$

- Avoid the use of adjacency matrix A, the similarity score between two feature maps $X_{i:}$ and $X_{j:}$ are calculated by $e_{ij} = \sum_{k=1}^{N} X_{ik} \times X_{jk}$

Hongyang Gao and Shuiwang Ji. "Graph representation learning via hard and channel-wise attention networks". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 741–749

hGAO and cGAO

Operator	Time Complexity	Space Complexity
GAO	$O(Cd)/O(N^2 \times d)$	$O(N^2)$
hGAO	$O(N \times \log N \times k + N \times k \times d^2)$	$O(N^2)$
cGAO	$O(N \times d^2)$	$O(d^2)$

- hGAO achieves significantly better performance than GAO on both node and graph embedding tasks.
- cGAO leads to dramatic savings in computational resources, making them applicable to large graphs.

Hongyang Gao and Shuiwang Ji. "Graph representation learning via hard and channel-wise attention networks". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 741–749.

AGNN

Similarity-based attention

$$\alpha_{0,j} = \frac{\exp\left(\beta \cdot \cos\left(Wx_0, Wx_j\right)\right)}{\sum_{k \in \Gamma_{v_0}} \exp\left(\beta \cdot \cos\left(Wx_0, Wx_j\right)\right)}$$

where β is trainable bias and cos represents cosine-similarity; W is a trainable weight matrix to map input features to the hidden space.

John Boaz Lee et al. "Attention models in graphs: A survey". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13.6 (2019), pp. 1–25

Graph Attention Model

- Input: Homogeneous graph
- Output: Graph embedding
- Mechanism: Learning attention weights over nodes; Attention-guild walk
- Task: Graph classification

John Boaz Lee et al. "Attention models in graphs: A survey". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13.6 (2019), pp. 1–25

Boris Knyazev, Graham W Taylor, and Mohamed Amer. "Understanding Attention and Generalization in Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4204–4214

Graph Attention Model

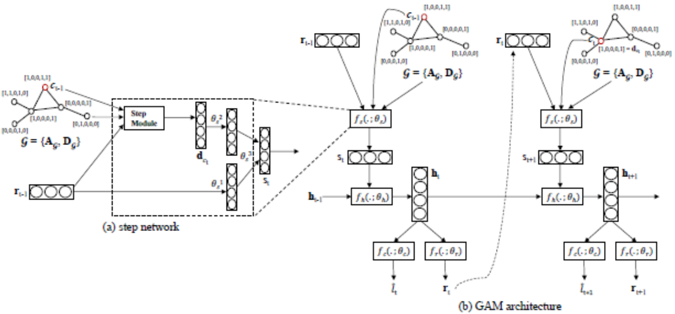
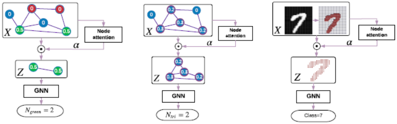


Figure 3: (a) Step network: Given a labeled graph $\mathcal{G}$ (composed of the adjacency matrix $\mathbf{A}_{\mathcal{G}}$, and the attribute matrix $\mathbf{D}_{\mathcal{G}}$), a current node c_{t-1}, and a stochastic rank vector $\mathbf{r}_{t-1}$, the step module takes a step from the current node c_{t-1} to one of its neighbors c_t, prioritizing those whose type (*i.e.*, node label) have higher rank in $\mathbf{r}_{t-1}$. The attribute vector of c_t, $\mathbf{d}_{c_t}$, is extracted and mapped to a hidden space using the linear layer parameterized by θ_s^2. Similarly, $\mathbf{r}_{t-1}$ is mapped using another linear layer parameterized by θ_s^1. Information from these two sources are then combined using a linear layer parameterized by θ_s^3 to produce $\mathbf{s}_t$, or the step embedding vector which represents information captured from the current step we took. (b) GAM architecture: We use an RNN as the core component of the model; in particular, we use the Long Short-Term Memory (LSTM) variant [12]. At each time step, the core network $f_h(.;\theta_h)$ takes the step embedding $\mathbf{s}_t$ and the internal representation of the model's history from the previous step $\mathbf{h}_{t-1}$ as input, and produces the new history vector $\mathbf{h}_t$. The history vector $\mathbf{h}_t$ can be thought of as a representation or summary of the information we've aggregated from our exploration of the graph thus far. The rank network $f_r(.;\theta_r)$ uses this to decide which types of nodes are more "interesting" and should thus be prioritized in future exploration. Likewise, the classification network $f_c(.;\theta_c)$ uses $\mathbf{h}_t$ to make a prediction on the graph label.

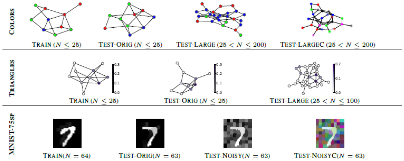
Understanding Attention

- Task: Colors, Traiangles, MNIST



(a) COLORS (b) TRIANGLES (c) MNIST

- Test subset: Orig, Large, LargeC / Noise, NoiseC



Boris Knyazev, Graham W Taylor, and Mohamed Amer. "Understanding Attention and Generalization in Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4204–4214.

Understanding Attention

Weakly-supervised attention supervision

Base on generating attention coefficients α^{WS}:

After training a model, remove node $i \in [1, N]$ and compute an absolute difference from prediction y for the original graph

$$\alpha_i^{WS} = \frac{|y_i - y|}{\sum_{j=1}^{N} |y_j - y|}$$

Use them as labels to train attention model with the loss function $\mathcal{L}$

$$\mathcal{L} = \mathcal{L}_{MSE/CE} + \frac{\beta}{N} \sum_i \alpha_i^{GT} \log\left(\frac{\alpha_i^{GT}}{\alpha_i}\right)$$

Agnostic to the choice of a dataset and model, that does not require ground truth attention labels, but can improve a model's ability to generalize

Boris Knyazev, Graham W Taylor, and Mohamed Amer. "Understanding Attention and Generalization in Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4204–4214

Understanding Attention

Table 1: **Results on three tasks for different test subsets.** $\pm$ denotes standard deviation, not shown in case of small values (large values are explained in Section 4). ATTN denotes attention accuracy in terms of AUC and is computed for the combined test set. The best result in each column (ignoring upper bound results) is bolded. ▢ denotes poor results with relatively low accuracy and/or high variance; ▢ denotes failed cases with accuracy close to random and/or extremely high variance. † For COLORS and MNIST-75SP, ChebyNets are used instead of ChebyGINs as described in the *Supp. Material*.

		COLORS				TRIANGLES			MNIST-75SP			
		ORIG	LARGE	LARGEC	ATTN	ORIG	LARGE	ATTN	ORIG	NOISY	NOISYC	ATTN
Global pool	GCN	97	72±15	20±3	99.6	46±1	23±1	79	78.3±2	38±4	36±4	72±2
	GIN	96±10	71±22	26±11	99.2	50±1	22±1	77	87.6±3	55±11	51±12	71±5
	ChebyGIN†	**100**	93±12	15±7	99.8	66±1	30±1	79	**97.4**	80±12	79±11	72±3
Unsuperv.	GIN, top-k	99.6	17±4	9±3	75±6	47±2	18±1	63±5	86±6	59±26	55±23	65±34
	GIN, ours	94±18	13±7	11±6	72±15	47±3	20±2	68±3	82.6±8	51±28	47±24	58±31
	ChebyGIN†, top-k	**100**	11±7	6±6	79±20	64±5	25±2	76±6	92.9±4	68±26	67±25	52±37
	ChebyGIN†, ours	80±30	16±10	11±6	67±31	67±3	26±2	77±4	94.6±3	80±23	77±22	78±31
Supervised	GIN, topk	87±1	39±18	28±8	**99.9**	49±1	20±1	88	90.5±1	85.5±2	79±5	99.3
	GIN, ours	**100**	**96±9**	**89±18**	99.8	49±1	22±1	76±1	90.9±0.4	85.0±1	80±3	99.3
	ChebyGIN†, topk	**100**	86±15	31±15	99.8	83±1	39±1	**97**	95.1±0.3	90.6±0.8	83±16	**100**
	ChebyGIN†, ours	**100**	94±8	75±17	99.8	**88±1**	**48±1**	96	95.4±0.2	**92.3±0.4**	**86±16**	**100**
Weak sup.	ChebyGIN†, ours	**100**	90±6	73±14	**99.9**	68±1	30±1	88	95.8±0.4	88.8±4	**86±9**	96.5±1
Upper bound	GIN	100	100	100	100	94±1	85±2	100	93.6±0.4	90.8±1	90.8±1	100
	ChebyGIN†	100	100	100	100	99.8	99.4±1	100	96.9±0.1	94.8±0.3	95.1±0.3	100

Boris Knyazev, Graham W Taylor, and Mohamed Amer. "Understanding Attention and Generalization in Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4204–4214.

Understanding Attention

- Attention can be extremely powerful in graph neural networks, but only if it is close to optimal
  - Three key factors influencing performance of GNNs with attention: initialization of the attention model, strength of the main GNN model, and finally other hyperparameters of the attention and GNN models
  - Highlight initialization as the critical factor, classification accuracy depends exponentially on attention correctness
- Attention can make GNNs more robust to larger and noisy graphs
  - GNNs with supervised training of attention are significantly more accurate and robust, although in case of a bad initialization it can take a long time to reach the performance of a better initialization
- Weakly-supervised approach brings advantages similar to the ones of supervised models, yet at the same time can be effectively applied to datasets without annotated attention

Boris Knyazev, Graham W Taylor, and Mohamed Amer. "Understanding Attention and Generalization in Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4204–4214

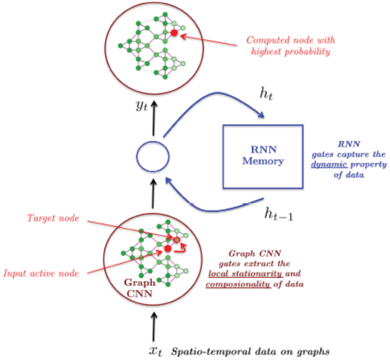
Spatial-temporal GNNs

- Occupy important positions in capturing the dynamicity of graphs.
- Follow two directions, RNN-based methods and CNN-based methods.
  - RNN-based: Capture spatial-temporal dependencies by filtering inputs and hidden states passed to a recurrent unit using graph convolutions.
  - CNN-based: Tackle spatial-temporal graphs in a non-recursive manner with the advantages of parallel computing, stable gradients, and low memory requirements.

Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020)

GCRN

- Merge CNN for graph-structured data and RNN to identify simultaneously meaningful spatial structures and dynamic patterns.
- RNN can easily exchanged with LSTM and GRU.



Youngjoo Seo et al. "Structured sequence modeling with graph convolutional recurrent networks". In: *International Conference on Neural Information Processing*. Springer. 2018, pp. 362–373

GCRN

Model1. Stack a graph CNN for feature extraction and an LSTM for sequence learning

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

Model2. Generalize convLSTM (Shi et al, 2015) to graph by replacing the Euclidean 2D convolution $*$ by graph convolution $*_{\mathcal{G}}$

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

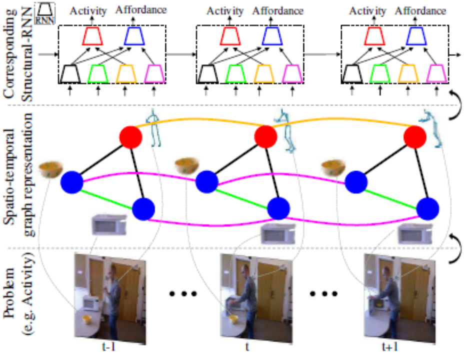
Youngjoo Seo et al. "Structured sequence modeling with graph convolutional recurrent networks". In: *International Conference on Neural Information Processing*. Springer. 2018, pp. 362–373

SHI Xingjian et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems*. 2015, pp. 802–810

S-RNN

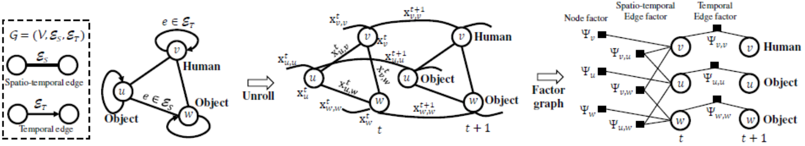
Transform an arbitrary st-graph into a feedforward mixture of RNNs

- Unrolled through time and decompose into a set of contributing factor components
- Group the factor components and represent each group using one RNN
- Allow the factors with similar semantic functions to share an RNN

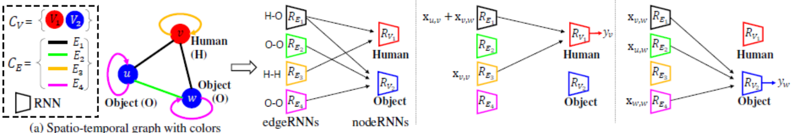


Ashesh Jain et al. "Structural-rnn: Deep learning on spatio-temporal graphs". In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2016, pp. 5308–5317

S-RNN



(a) Spatio-temporal graph representing an activity (b) Unrolled through time (c) Factor graph parameterization



(a) Spatio-temporal graph with colors indicating sharing of factors (b) Corresponding S-RNN (c) Forward-pass for human node v (d) Forward-pass for object node w

Ashesh Jain et al. "Structural-rnn: Deep learning on spatio-temporal graphs". In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2016, pp. 5308–5317

DCRNN

Diffusion Convolution

$$X_{:,p} *_{\mathcal{G}} f_\theta = \sum_{k=0}^{K-1} \left(\theta_{k,1}\left(D_O^{-1}W\right)^k + \theta_{k,2}\left(D_I^{-1}W^T\right)^k\right) X_{:,p} \quad \text{for } p \in \{1, \dots, P\}$$

Diffusion Concolution Layer

$$H_{:,q} = a\left(\sum_{p=1}^{P} X_{:,p} *_{\mathcal{G}} f_{\Theta_{q,p,:,:}}\right) \quad \text{for } q \in \{1, \dots, Q\}$$

Diffusion Convolutional Gated Recurrent Unit

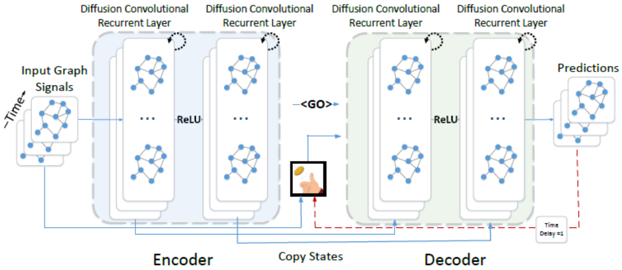
$$r^{(t)} = \sigma\left(\Theta_r *_{\mathcal{G}} [X^{(t)}, H^{(t-1)}] + b_r\right) \qquad u^{(t)} = \sigma\left(\Theta_u *_{\mathcal{G}} [X^{(t)}, H^{(t-1)}] + b_u\right)$$

$$C^{(t)} = \tanh\left(\Theta_C *_{\mathcal{G}} [X^{(t)}, (r^{(t)} \odot H^{(t-1)})] + b_c\right) \qquad H^{(t)} = u^{(t)} \odot H^{(t-1)} + (1 - u^{(t)}) \odot C^{(t)}$$

Yaguang Li et al. "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting". In: *arXiv preprint arXiv:1707.01926* (2017)

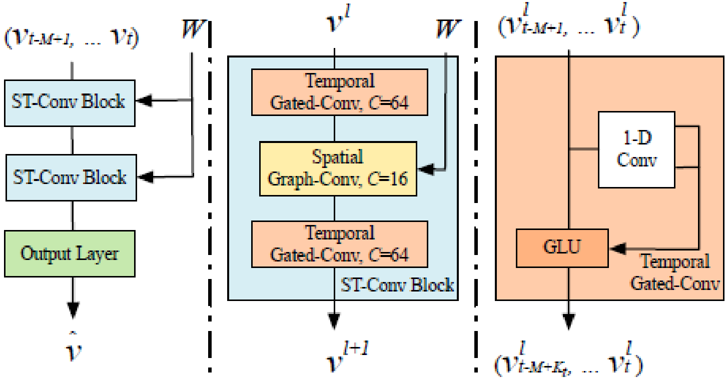
DCRNN

- Trained by maximizing the likelihood of generating the target future time series using backpropagation through time.
- Capture spatio-temporal dependencies among time series
- Apply to various spatio-temporal forecasting problems



Yaguang Li et al. "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting". In: *arXiv preprint arXiv:1707.01926* (2017)

STGCN



Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting". In: *arXiv preprint arXiv:1709.04875* (2017)

STGCN

Graph CNNs

Chebyshev Poly Approx $$\Theta *_{\mathcal{G}} x = \Theta(L)x \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x$$

1st-order Approximation $$\Theta *_{\mathcal{G}} x = \theta(I_n + D^{-\frac{1}{2}} W D^{-\frac{1}{2}})x = \theta(\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}})x$$

Gated CNNs

$$\Gamma *_{\mathcal{T}} Y = P \odot \sigma(Q) \in \mathbb{R}^{(M-K_t+1)\times C_o}$$

ST-Conv Block

$$v^{l+1} = \Gamma_1^l *_{\mathcal{T}} \mathrm{ReLU}\left(\Theta^l * \mathcal{G}(\Gamma_0^l *_{\mathcal{T}} v^l)\right)$$

Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting". In: *arXiv preprint arXiv:1709.04875* (2017)

STGCN

- A universal framework to process structured time series applied to spatio-temporal sequence learning tasks.
- Extract the most useful spatial features and capture the most essential temporal features coherently.
- Achieve parallelization over input with fewer parameters and faster training speed; handle large-scale networks with more efficiency.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting". In: *arXiv preprint arXiv:1709.04875* (2017)

Graph WaveNet

- **Self-adaptive Adjacency Matrix**

$$\tilde{A}_{adp} = \mathrm{softmax}(\mathrm{ReLU}(E_1 E_2^T))$$

$$Z = \sum_{k=0}^{K} P_f^k X W_{k1} + P_b^k X W_{k2} + \tilde{A}_{adp}^k X W_k$$

  - Generalize the diffusion convolution layer, P^k represents the power series of the transition matrix
  - $P = A/\mathrm{rowsum}(A)$ in case of undirected graph; $P_f = A/\mathrm{rowsum}(A)$, $P_b = A^T/\mathrm{rowsum}(A^T)$ in case of directed graph
  - Do not require prior knowledge and is learned end-to-end through stochastic gradient descent

---

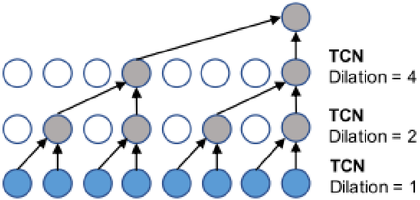
Zonghan Wu et al. "Graph wavenet for deep spatial-temporal graph modeling". In: *arXiv preprint arXiv:1906.00121* (2019).

Graph WaveNet

- **Gated TCN**

  $h = g\left(\Theta_1 * \mathcal{X} + b\right) \odot \sigma\left(\Theta_2 * \mathcal{X} + c\right)$

- Adopt the dilated causal convolution as temporal convolution layer (TCN)
- To learn Complex temporal dependencies
- Other forms of Gated TCN can be easily filter into the framework

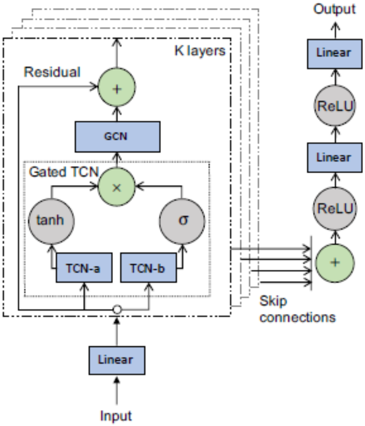


Dilated causal convolution with kernel size 2. With a dilation factor k, it picks inputs every k step and applies the standard $1D$ convolution to the selected inputs

Zonghan Wu et al. "Graph wavenet for deep spatial-temporal graph modeling". In: *arXiv preprint arXiv:1906.00121* (2019).

Graph WaveNet

- Capture spatial-temporal dependencies efficiently and effectively by combining graph convolution with dilated casual convolution
- Learn hidden spatial dependencies automatically from data
- Handle spatial-temporal graph data with long-range temporal sequences.



Zonghan Wu et al. "Graph wavenet for deep spatial-temporal graph modeling". In: *arXiv preprint arXiv:1906.00121* (2019).

References I

John Boaz Lee et al. "Attention models in graphs: A survey". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13.6 (2019), pp. 1–25.

Boris Knyazev, Graham W Taylor, and Mohamed Amer. "Understanding Attention and Generalization in Graph Neural Networks". In: *Advances in Neural Information Processing Systems.* 2019, pp. 4204–4214.

Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

Jiani Zhang et al. "Gaan: Gated attention networks for learning on large and spatiotemporal graphs". In: *arXiv preprint arXiv:1803.07294* (2018).

References II

Hongyang Gao and Shuiwang Ji. "Graph representation learning via hard and channel-wise attention networks". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 741–749.

John Boaz Lee, Ryan Rossi, and Xiangnan Kong. "Graph classification using structural attention". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1666–1674.

Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

References III

Youngjoo Seo et al. "Structured sequence modeling with graph convolutional recurrent networks". In: *International Conference on Neural Information Processing*. Springer. 2018, pp. 362–373.

SHI Xingjian et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting". In: *Advances in neural information processing systems*. 2015, pp. 802–810.

Ashesh Jain et al. "Structural-rnn: Deep learning on spatio-temporal graphs". In: *Proceedings of the ieee conference on computer vision and pattern recognition*. 2016, pp. 5308–5317.

Yaguang Li et al. "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting". In: *arXiv preprint arXiv:1707.01926* (2017).

References IV

- Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting". In: *arXiv preprint arXiv:1709.04875* (2017).
- Zonghan Wu et al. "Graph wavenet for deep spatial-temporal graph modeling". In: *arXiv preprint arXiv:1906.00121* (2019).