Brief Introduction to GNN

Jingru Li

04.21

Content

- Introduction
- Models
  - Graph Neural networks
  - Variants of GNN
  - General Frameworks
- Applications
- Open Problems
- Conclusion

Introduction

- A unique non-Euclidean data structure of machine learning
- Deep Learning Method operated on graph domain
- Motivation:
  - CNN
  - Graph Embedding

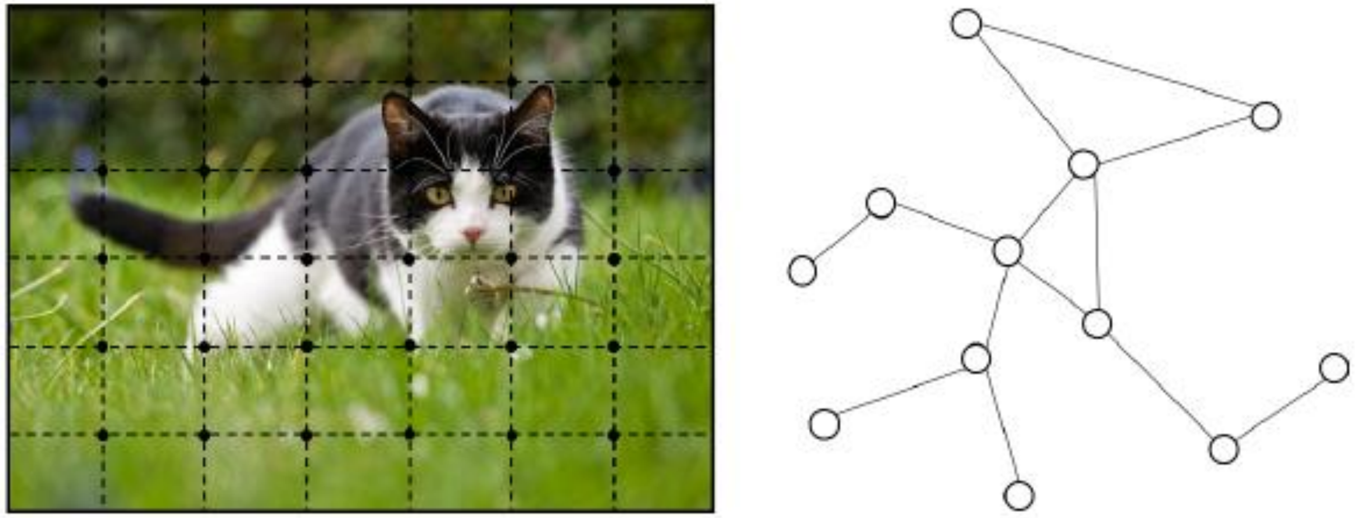


Fig. 1. Left: image in Euclidean space. Right: graph in non-Euclidean space

Introduction – Why GNN is worthy

Firstly, the standard neural networks like CNNs and RNNs cannot handle the graph input properly in that they stack the feature of nodes by a specific order.

Secondly, an edge in a graph represents the information of dependency between two nodes.

Thirdly, reasoning is a very important research topic for high-level artificial intelligence and the reasoning process in human brain is almost based on the graph which is extracted from daily experience.

Model – GNN[1]

Target of GNN:

- Learn a state embedding $\mathbf{h}_v$ containing info of neighbor.

Local learning:

$$\mathbf{h}_v = f(\mathbf{x}_v, \mathbf{x}_{co[v]}, \mathbf{h}_{ne[v]}, \mathbf{x}_{ne[v]}) \quad (1)$$

$$\mathbf{o}_v = g(\mathbf{h}_v, \mathbf{x}_v) \quad (2)$$

Global learning:

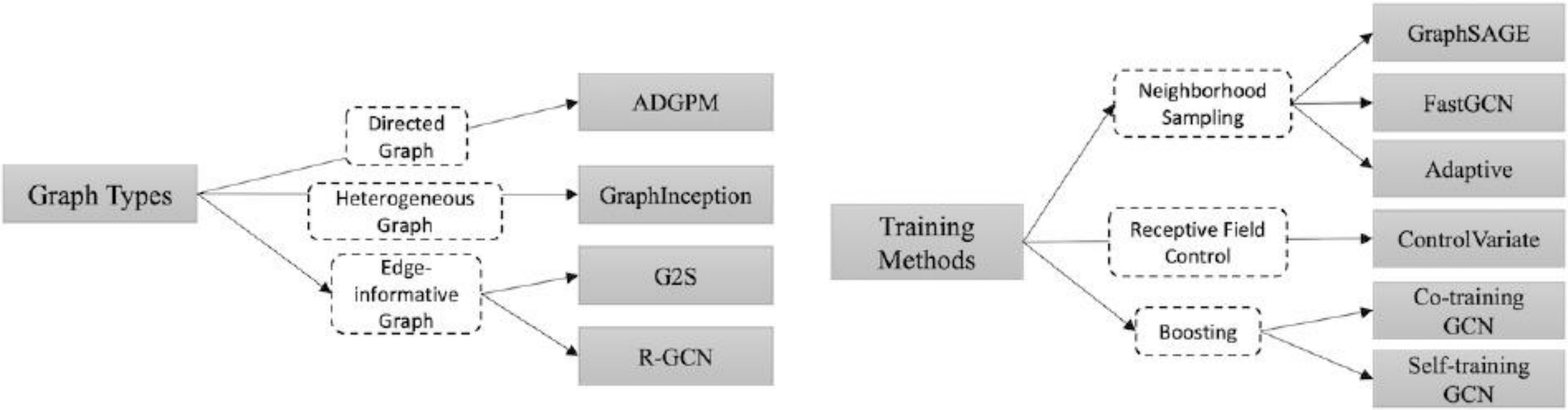
$$\mathbf{H} = F(\mathbf{H}, \mathbf{X}) \quad (3)$$

$$\mathbf{O} = G(\mathbf{H}, \mathbf{X}_N) \quad (4)$$

TABLE 1
Notations used in this paper.

Notations	Descriptions
$\mathbb{R}^m$	m-dimensional Euclidean space
a, $\mathbf{a}$, $\mathbf{A}$	Scalar, vector, matrix
$\mathbf{A}^T$	Matrix transpose
$\mathbf{I}_N$	Identity matrix of dimension N
$\mathbf{g}_\theta \star \mathbf{x}$	Convolution of $\mathbf{g}_\theta$ and $\mathbf{x}$
N	Number of nodes in the graph
N^v	Number of nodes in the graph
N^e	Number of edges in the graph
$\mathcal{N}_v$	Neighborhood set of node v
$\mathbf{a}_v^t$	Vector $\mathbf{a}$ of node v at time step t
$\mathbf{h}_v$	Hidden state of node v
$\mathbf{h}_v^t$	Hidden state of node v at time step t
$\mathbf{e}_{vw}$	Features of edge from node v to w
$\mathbf{e}_k$	Features of edge with label k
$\mathbf{o}_v^t$	Output of node v
$\mathbf{W}^i, \mathbf{U}^i, \mathbf{W}^o, \mathbf{U}^o, \ldots$	Matrices for computing $\mathbf{i}, \mathbf{o}, \ldots$
$\mathbf{b}^i, \mathbf{b}^o, \ldots$	Vectors for computing $\mathbf{i}, \mathbf{o}, \ldots$
σ	The logistic sigmoid function
ρ	An alternative non-linear function
tanh	The hyperbolic tangent function
LeakyReLU	The LeakyReLU function
$\odot$	Element-wise multiplication operation
$\|$	Vector concatenation

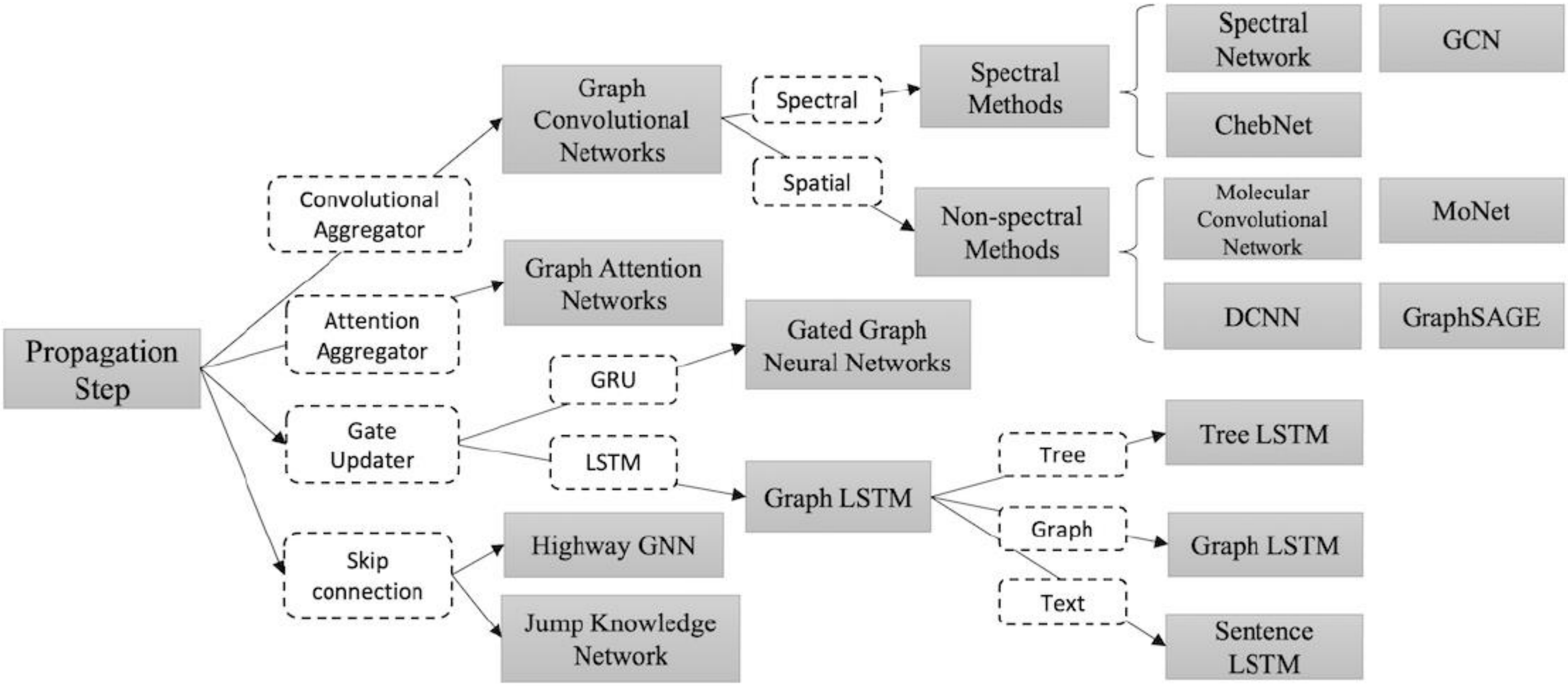
Model – Variants of GNN



(a) Graph Types

(b) Training Methods

Model – Variants of GNN



(c) Propagation Steps

Model – Variants of GNN

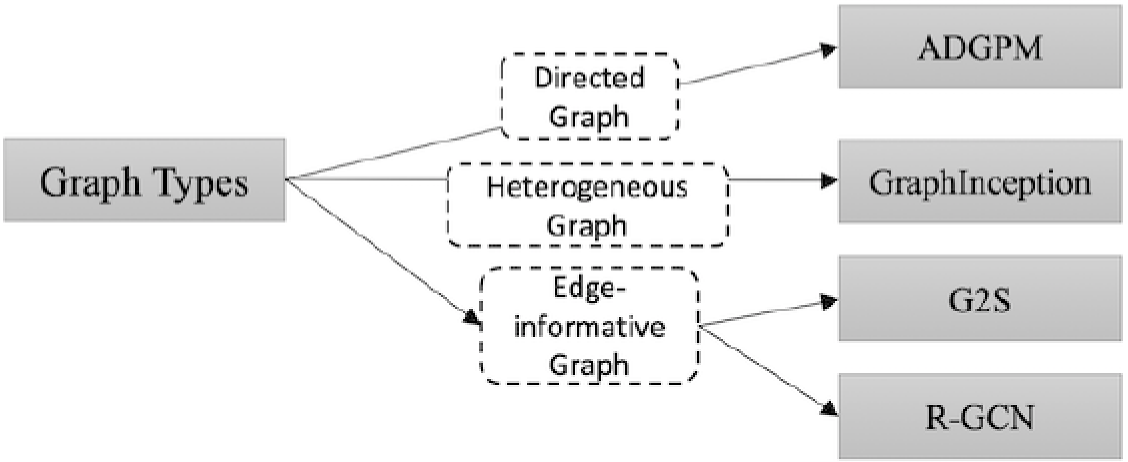
Directed Graph(ADGPM[2])

$$\mathbf{H}^t = \sigma(\mathbf{D}_p^{-1}\mathbf{A}_p\sigma(\mathbf{D}_c^{-1}\mathbf{A}_c\mathbf{H}^{t-1}\mathbf{W}_c)\mathbf{W}_p) \tag{7}$$

where $\mathbf{D}_p^{-1}\mathbf{A}_p$, $\mathbf{D}_c^{-1}\mathbf{A}_c$ are the normalized adjacency matrix for parents and children respectively.

Heterogeneous Graphs

Concept of metapath into the propagation on the heterogeneous graph.



(a) Graph Types

Model – Variants of GNN

Graphs with Edge Information

Convert the graph to a bipartite graph(G2S[3])

$$\mathbf{h}_v^t = \rho\left(\frac{1}{|\mathcal{N}_v|}\sum_{u\in\mathcal{N}_v}\mathbf{W}_r(\mathbf{r}_v^t \odot \mathbf{h}_u^{t-1}) + \mathbf{b}_r\right)$$

Adapt different weight matrices for the propagation on different kinds of edges.(r-GCN[4])

$$\mathbf{W}_r = \sum_1^B a_{rb}\mathbf{V}_b$$

Model – Variants of GNN

Name	Variant	Aggregator	Updater
Spectral Methods	ChebNet	$\mathbf{N}_k = \mathbf{T}_k(\tilde{\mathbf{L}})\mathbf{X}$	$\mathbf{H} = \sum_{k=0}^{K} \mathbf{N}_k \mathbf{\Theta}_k$
	1^{st}-order model	$\mathbf{N}_0 = \mathbf{X}$ $\mathbf{N}_1 = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{X}$	$\mathbf{H} = \mathbf{N}_0\mathbf{\Theta}_0 + \mathbf{N}_1\mathbf{\Theta}_1$
	Single parameter	$\mathbf{N} = (\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{X}$	$\mathbf{H} = \mathbf{N}\mathbf{\Theta}$
	GCN	$\mathbf{N} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}$	$\mathbf{H} = \mathbf{N}\mathbf{\Theta}$
Non-spectral Methods	Convolutional networks in [33]	$\mathbf{h}_{\mathcal{N}_v}^t = \mathbf{h}_v^{t-1} + \sum_{k=1}^{\mathcal{N}_v} \mathbf{h}_k^{t-1}$	$\mathbf{h}_v^t = \sigma(\mathbf{h}_{\mathcal{N}_v}^t \mathbf{W}_L^{\mathcal{N}_v})$
	DCNN	Node classification: $\mathbf{N} = \mathbf{P}^*\mathbf{X}$ Graph classification: $\mathbf{N} = 1_N^T\mathbf{P}^*\mathbf{X}/N$	$\mathbf{H} = f\left(\mathbf{W}^c \odot \mathbf{N}\right)$
	GraphSAGE	$\mathbf{h}_{\mathcal{N}_v}^t = \mathrm{AGGREGATE}_t\left(\{\mathbf{h}_u^{t-1}, \forall u \in \mathcal{N}_v\}\right)$	$\mathbf{h}_v^t = \sigma\left(\mathbf{W}^t \cdot [\mathbf{h}_v^{t-1} \Vert \mathbf{h}_{\mathcal{N}_v}^t]\right)$

Model – Variants of GNN

Graph Attention Networks	GAT	$\alpha_{vk} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v \Vert \mathbf{W}\mathbf{h}_k]\right)\right)}{\sum_{j\in\mathcal{N}_v}\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v \Vert \mathbf{W}\mathbf{h}_j]))}$ $\mathbf{h}_{\mathcal{N}_v}^t = \sigma\left(\sum_{k\in\mathcal{N}_v}\alpha_{vk}\mathbf{W}\mathbf{h}_k\right)$ Multi-head concatenation: $\mathbf{h}_{\mathcal{N}_v}^t = \Big\Vert_{m=1}^{M}\sigma\left(\sum_{k\in\mathcal{N}_v}\alpha_{vk}^m\mathbf{W}^m\mathbf{h}_k\right)$ Multi-head average: $\mathbf{h}_{\mathcal{N}_v}^t = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{k\in\mathcal{N}_v}\alpha_{vk}^m\mathbf{W}^m\mathbf{h}_k\right)$	$\mathbf{h}_v^t = \mathbf{h}_{\mathcal{N}_v}^t$
Gated Graph Neural Networks	GGNN	$\mathbf{h}_{\mathcal{N}_v}^t = \sum_{k\in\mathcal{N}_v}\mathbf{h}_k^{t-1} + \mathbf{b}$	$\mathbf{z}_v^t = \sigma(\mathbf{W}^z\mathbf{h}_{\mathcal{N}_v}^t + \mathbf{U}^z\mathbf{h}_v^{t-1})$ $\mathbf{r}_v^t = \sigma(\mathbf{W}^r\mathbf{h}_{\mathcal{N}_v}^t + \mathbf{U}^r\mathbf{h}_v^{t-1})$ $\widetilde{\mathbf{h}_v^t} = \tanh(\mathbf{W}\mathbf{h}_{\mathcal{N}_v}^t + \mathbf{U}(\mathbf{r}_v^t \odot \mathbf{h}_v^{t-1}))$ $\mathbf{h}_v^t = (1-\mathbf{z}_v^t)\odot\mathbf{h}_v^{t-1} + \mathbf{z}_v^t\odot\widetilde{\mathbf{h}_v^t}$

Model – Variants of GNN

Graph LSTM	Tree LSTM (Child sum)	$\mathbf{h}_{\mathcal{N}_v}^t = \sum_{k\in\mathcal{N}_v} \mathbf{h}_k^{t-1}$	$\mathbf{i}_v^t = \sigma(\mathbf{W}^i\mathbf{x}_v^t + \mathbf{U}^i\mathbf{h}_{\mathcal{N}_v}^t + \mathbf{b}^i)$ $\mathbf{f}_{vk}^t = \sigma\left(\mathbf{W}^f\mathbf{x}_v^t + \mathbf{U}^f\mathbf{h}_k^{t-1} + \mathbf{b}^f\right)$ $\mathbf{o}_v^t = \sigma(\mathbf{W}^o\mathbf{x}_v^t + \mathbf{U}^o\mathbf{h}_{\mathcal{N}_v}^t + \mathbf{b}^o)$ $\mathbf{u}_v^t = \tanh(\mathbf{W}^u\mathbf{x}_v^t + \mathbf{U}^u\mathbf{h}_{\mathcal{N}_v}^t + \mathbf{b}^u)$ $\mathbf{c}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{k\in\mathcal{N}_v} \mathbf{f}_{vk}^t \odot \mathbf{c}_k^{t-1}$ $\mathbf{h}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t)$
	Tree LSTM (N-ary)	$\mathbf{h}_{\mathcal{N}_v}^{ti} = \sum_{l=1}^K \mathbf{U}_l^i\mathbf{h}_{vl}^{t-1}$ $\mathbf{h}_{\mathcal{N}_v k}^{tf} = \sum_{l=1}^K \mathbf{U}_{kl}^f\mathbf{h}_{vl}^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{to} = \sum_{l=1}^K \mathbf{U}_l^o\mathbf{h}_{vl}^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{tu} = \sum_{l=1}^K \mathbf{U}_l^u\mathbf{h}_{vl}^{t-1}$	$\mathbf{i}_v^t = \sigma(\mathbf{W}^i\mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{ti} + \mathbf{b}^i)$ $\mathbf{f}_{vk}^t = \sigma(\mathbf{W}^f\mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v k}^{tf} + \mathbf{b}^f)$ $\mathbf{o}_v^t = \sigma(\mathbf{W}^o\mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{to} + \mathbf{b}^o)$ $\mathbf{u}_v^t = \tanh(\mathbf{W}^u\mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{tu} + \mathbf{b}^u)$ $\mathbf{c}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{l=1}^K \mathbf{f}_{vl}^t \odot \mathbf{c}_{vl}^{t-1}$ $\mathbf{h}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t)$
	Graph LSTM in [34]	$\mathbf{h}_{\mathcal{N}_v}^{ti} = \sum_{k\in\mathcal{N}_v} \mathbf{U}_{m(v,k)}^i\mathbf{h}_k^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{to} = \sum_{k\in\mathcal{N}_v} \mathbf{U}_{m(v,k)}^o\mathbf{h}_k^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{tu} = \sum_{k\in\mathcal{N}_v} \mathbf{U}_{m(v,k)}^u\mathbf{h}_k^{t-1}$	$\mathbf{i}_v^t = \sigma(\mathbf{W}^i\mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{ti} + \mathbf{b}^i)$ $\mathbf{f}_{vk}^t = \sigma(\mathbf{W}^f\mathbf{x}_v^t + \mathbf{U}_{m(v,k)}^f\mathbf{h}_k^{t-1} + \mathbf{b}^f)$ $\mathbf{o}_v^t = \sigma(\mathbf{W}^o\mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{to} + \mathbf{b}^o)$ $\mathbf{u}_v^t = \tanh(\mathbf{W}^u\mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{tu} + \mathbf{b}^u)$ $\mathbf{c}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{k\in\mathcal{N}_v} \mathbf{f}_{vk}^t \odot \mathbf{c}_k^{t-1}$ $\mathbf{h}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t)$

Model Example - Convolutions

$$\mathbf{g}_\theta \star \mathbf{x} = \mathbf{U}\mathbf{g}_\theta(\Lambda)\mathbf{U}^T\mathbf{x}$$

$$\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{U}\Lambda\mathbf{U}^T$$

For ChebNet[5]:

$$\mathbf{g}_\theta \star \mathbf{x} \approx \sum_{k=0}^{K} \theta_k \mathbf{T}_k(\tilde{\mathbf{L}})\mathbf{x}$$

For GCN[6]:

$$\mathbf{g}_\theta \star \mathbf{x} \approx \theta \left(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\right)\mathbf{x}$$

For Non-spectral methods, they define convolutions directly on the graph, operating on spatially close neighbors

Model Example – Attention[7]

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

$$\mathbf{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\mathbf{h}_j\right)$$

For multi-head attention:

$$\mathbf{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j\right)$$

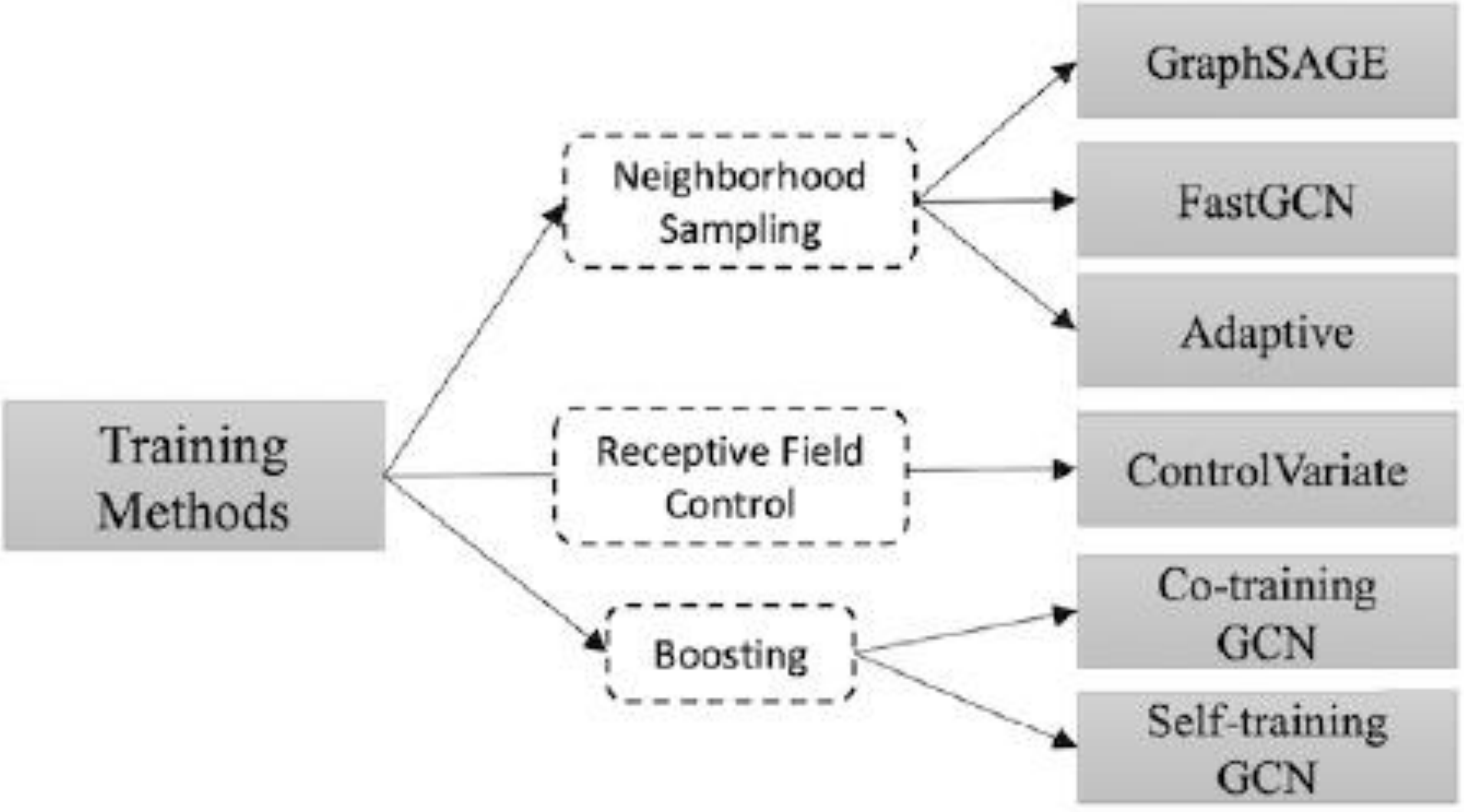
$$\mathbf{h}'_i = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j\right)$$

Model Example – Gate Method

During propagation, gate method diminishes the restrictions in the former GNN models and improve the long-term propagation of information across the graph structure.

$$
\begin{aligned}
\mathbf{a}_v^t &= \mathbf{A}_v^T [\mathbf{h}_1^{t-1} \dots \mathbf{h}_N^{t-1}]^T + \mathbf{b} \\
\mathbf{z}_v^t &= \sigma\left(\mathbf{W}^z \mathbf{a}_v^t + \mathbf{U}^z \mathbf{h}_v^{t-1}\right) \\
\mathbf{r}_v^t &= \sigma\left(\mathbf{W}^r \mathbf{a}_v^t + \mathbf{U}^r \mathbf{h}_v^{t-1}\right) \\
\widetilde{\mathbf{h}_v^t} &= \tanh\left(\mathbf{W} \mathbf{a}_v^t + \mathbf{U}\left(\mathbf{r}_v^t \odot \mathbf{h}_v^{t-1}\right)\right) \\
\mathbf{h}_v^t &= \left(1 - \mathbf{z}_v^t\right) \odot \mathbf{h}_v^{t-1} + \mathbf{z}_v^t \odot \widetilde{\mathbf{h}_v^t}
\end{aligned}
$$

Model – Training Methods



(b) Training Methods

Model – General Frameworks

MPNN(Message Passing Neural Networks)[9]

Containing two phases, message passing and readout.

$$\mathrm{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} M_t\left(\mathrm{h}_v^t, \mathrm{h}_w^t, \mathrm{e}_{vw}\right)$$

$$\mathrm{h}_v^{t+1} = U_t\left(\mathrm{h}_v^t, \mathrm{m}_v^{t+1}\right)$$

$$\hat{\mathrm{y}} = R(\{\mathrm{h}_v^T | v \in G\})$$

$$M_t\left(\mathrm{h}_v^t, \mathrm{h}_w^t, \mathrm{e}_{vw}\right) = \mathbf{A}_{\mathbf{e}_{vw}} \mathrm{h}_w^t$$

$$U_t = GRU\left(\mathrm{h}_v^t, \mathrm{m}_v^{t+1}\right)$$

$$R = \sum_{v \in V} \sigma\left(i(\mathrm{h}_v^T, \mathrm{h}_v^0)\right) \odot \left(j(\mathrm{h}_v^T)\right)$$

Model – General Frameworks

Non-local Neural Networks[10]

Have proposed the Non-local Neural Networks (NLNN) for capturing long-range dependencies with deep neural networks.

$$\mathrm{h}'_i = \frac{1}{\mathcal{C}(\mathrm{h})} \sum_{\forall j} f(\mathrm{h}_i, \mathrm{h}_j) g(\mathrm{h}_j)$$

We should figure out f and g function, including:
- Gaussian
- Embed Gaussian
- Dot Product
- Concatenation

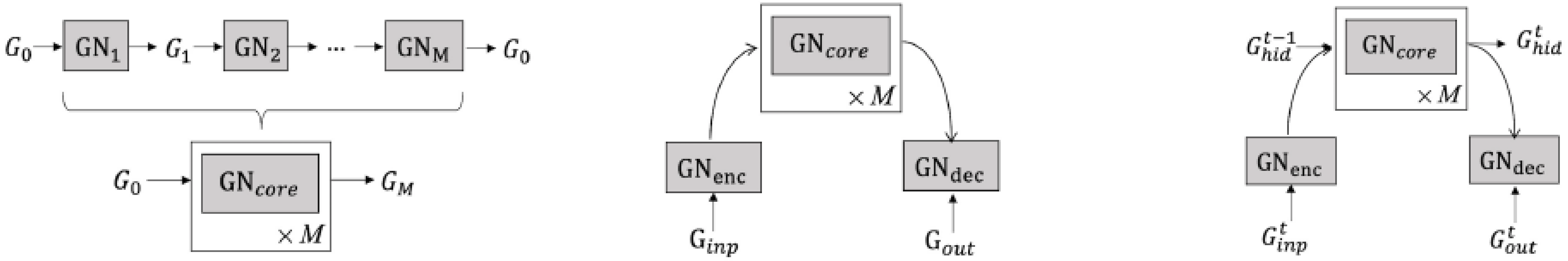
Model – General Frameworks

Graph Networks[11]

Graph Network (GN) framework generalizes and extends various graph neural network, MPNN and NLNN approaches.

Design Principles:

- Flexible representations
- Configurable within-block structure.
- Composable multi-block architectures



(a) Sequential GN blocks (b) Encode-process-decode (c) Recurrent GN blocks

Fig. 3. Examples of architectures composed by GN blocks. (a) The sequential processing architecture; (b) The encode-process-decode architecture; (c) The recurrent architecture.

Applications

Area	Application	Algorithm	Deep Learning Model
Text	Text classification	GCN	Graph Convolutional Network
		GAT	Graph Attention Network
		DGCNN	Graph Convolutional Network
		Text GCN	Graph Convolutional Network
		Sentence LSTM	Graph LSTM
	Sequence Labeling (POS, NER)	Sentence LSTM	Graph LSTM
	Sentiment classification	Tree LSTM	Graph LSTM
	Semantic role labeling	Syntactic GCN	Graph Convolutional Network
	Neural machine translation	Syntactic GCN	Graph Convolutional Network
		GGNN	Gated Graph Neural Network
	Relation extraction	Tree LSTM	Graph LSTM
		Graph LSTM	Graph LSTM
		GCN	Graph Convolutional Network
	Event extraction	Syntactic GCN	Graph Convolutional Network
	AMR to text generation	Sentence LSTM	Graph LSTM
		GGNN	Gated Graph Neural Network
	Multi-hop reading comprehension	Sentence LSTM	Graph LSTM
	Relational reasoning	RN	MLP
		Recurrent RN	Recurrent Neural Network
		IN	Graph Neural Network

Applications

Image	Social Relationship Understanding	GRM	Gated Graph Neural Network
	Image classification	GCN	Graph Convolutional Network
		GGNN	Gated Graph Neural Network
		ADGPM	Graph Convolutional Network
		GSNN	Gated Graph Neural Network
	Visual Question Answering	GGNN	Gated Graph Neural Network
	Object Detection	RN	Graph Attention Network
	Interaction Detection	GPNN	Graph Neural Network
		Structural-RNN	Graph Neural Network
	Region Classification	GCNN	Graph CNN
	Semantic Segmentation	Graph LSTM	Graph LSTM
		GGNN	Gated Graph Neural Network
		DGCNN	Graph CNN
		3DGNN	Graph Neural Network
Science	Physics Systems	IN	Graph Neural Network
		VIN	Graph Neural Network
		GN	Graph Networks
	Molecular Fingerprints	NGF	Graph Convolutional Network
		GCN	Graph Convolutional Network
	Protein Interface Prediction	GCN	Graph Convolutional Network
	Side Effects Prediction	Decagon	Graph Convolutional Network
	Disease Classification	PPIN	Graph Convolutional Network
Knowledge Graph	KB Completion	GNN	Graph Neural Network
	KG Alignment	GCN	Graph Convolutional Network

Applications

Combinatorial Optimization	structure2vec	Graph Convolutional Network
	GNN	Graph Neural Network
	GCN	Graph Convolutional Network
	AM	Graph Attention Network
Graph Generation	NetGAN	Long short-term memory
	GraphRNN	Rucurrent Neural Network
	Regularizing VAE	Variational Autoencoder
	GCPN	Graph Convolutional Network
	MolGAN	Relational-GCN

Mainly focus on the application on images and texts, and some generative methods.

Open Problems

- Shallow Structure
- Dynamic Graphs
- Non-Structural Scenarios
- Scalability

Resources

GNN paper links:

https://github.com/thunlp/GNNPapers

Survey Papers:

https://arxiv.org/pdf/1812.08434.pdf
https://arxiv.org/pdf/1901.00596.pdf
https://arxiv.org/pdf/1812.04202.pdf