Special topics: Convolutional Neural Networks

Week 9: Convolutional Graph NN

Lecturer: Nguyen Thi Ngoc Diep, Ph.D.

Email: ngocdiep@vnu.edu.vn

Slide & Code: https://github.com/chupibk/INT3414_22

1

Schedule

Week	Content	Class hour	Self-study hour
1	Introduction CNNs in Computer Vision	2	1
2	~~Foundations of CNNs~~ ~~Case study: Image classification problem~~ Basics of Neural networks Training with backpropagation Implementation with PyTorch	2	2-6
3	~~Training and tuning parameters~~ ~~Data augmentation – Data generator~~ Foundations of CNNs Transfer learning Mid-term assignment	2	2-6
4	Object detection	2	2-6
5	Segmentation	2	2-6
6, 7	Mid-term presentations	2	2-6
8, 9	Advanced topics using CNNs	2	2-6
10	Debug for Final project		
11, 12	Final project presentations	1	2-6
13	Class summarization	1-3	open

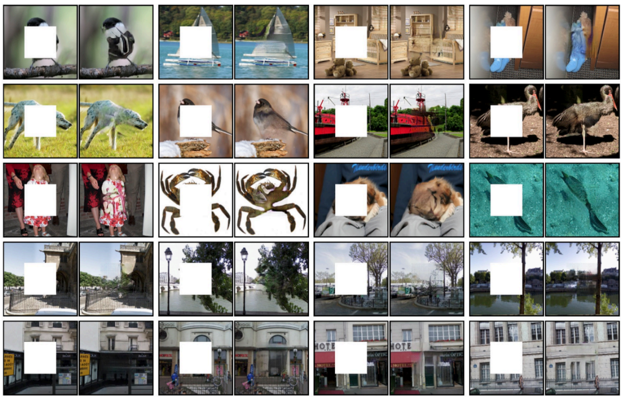
2

Recall week 8: Generative adversarial network

- Image synthesis/generation
- Super resolution
- Image to Image Translation
- Neural style transfer
- Text to image synthesis
- Image inpainting

3

Image Inpainting



Ref: Context Decoder

4

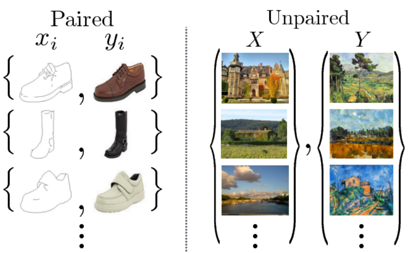
Text to image

Ref: StackGAN

5

Datasets and training

- Dataset:
  - Paired
  - Or, unpaired
- Training:
  - Generator
  - Discriminator



6

GAN example: StackGAN

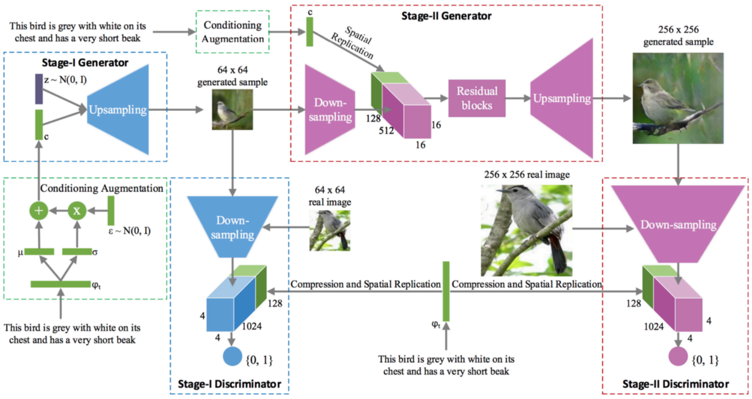


Figure 2. The architecture of the proposed StackGAN. The Stage-I generator draws a low resolution image by sketching rough shape and basic colors of the object from the given text and painting the background from a random noise vector. The Stage-II generator generates a high resolution image with photo-realistic details by conditioning on both the Stage-I result and the text again.

Ref: https://arxiv.org/pdf/1612.03242v1.pdf

7

Tip: Model summary in PyTorch

8

Model summary

- Keras: model.summary()
- Pytorch: https://github.com/sksq96/pytorch-summary

  - `pip install torchsummary` or
  - `git clone https://github.com/sksq96/pytorch-summary`

```
from torchsummary import summary
summary(your_model, input_size=(channels, H, W))
```

  - Note that the `input_size` is required to make a forward pass through the network.

9

Tip: Data resource for *playing*

10

Setting up kaggle

https://towardsdatascience.com/setting-up-kaggle-in-google-colab-ebb281b61463

11

Graph Neural Networks

12

A Comprehensive Survey on Graph Neural Networks

Zonghan Wu, Shirui Pan, *Member, IEEE*, Fengwen Chen, Guodong Long, Chengqi Zhang, *Senior Member, IEEE*, Philip S. Yu, *Fellow, IEEE*

- Geometric deep learning:
  - Graphs
  - Manifolds
- Graph neural networks (GNNs)
  - → addressing the problem of network embedding
  - → or as building blocks for learning from relational data
  - Taxonomy:
    - Recurrent graph neural networks
    - Convolutional graph neural networks
    - Graph autoencoders
    - Spatial-temporal graph neural networks

13

Examples of application

- Social network (relationship graph)
- Citation graphs of scientific papers
- Interaction graphs between users and products
- In chemistry: graphs of molecules
- In computer vision: scene graph generation, point clouds classification, action recognition
- In natural language processing: semantic graph, inter-relation of documents or words to infer document labels
- Others: program verification, brain networks, event detection...

14

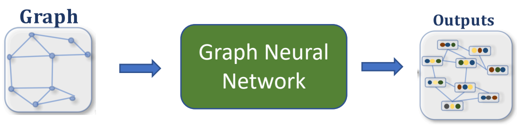
Benchmark datasets

Category	Data set	Source	# Graphs	# Nodes(Avg.)	# Edges (Avg.)	#Features	# Classes	Citation
Citation Networks	Cora	[117]	1	2708	5429	1433	7	[22], [23], [25], [41], [43], [44], [45] [49], [50], [51], [53], [56], [61], [62]
	Citeseer	[117]	1	3327	4732	3703	6	[22], [41], [43], [45], [50], [51], [53] [56], [61], [62]
	Pubmed	[117]	1	19717	44338	500	3	[18], [22], [25], [41], [43], [44], [45] [49], [51], [53], [55], [56], [61], [62] [70], [95]
	DBLP (v11)	[118]	1	4107340	36624464	-	-	[64], [70], [99]
Bio-chemical Graphs	PPI	[119]	24	56944	818716	50	121	[18], [42], [43], [48], [45], [50], [55] [56], [58], [64]
	NCI-1	[120]	4110	29.87	32.30	37	2	[25], [26], [46], [52], [57], [96], [98]
	MUTAG	[121]	188	17.93	19.79	7	2	[25], [26], [46], [52], [57], [96]
	D&D	[122]	1178	284.31	715.65	82	2	[26], [46], [52], [54], [96], [98]
	PROTEIN	[123]	1113	39.06	72.81	4	2	[26], [46], [52], [54], [57]
	PTC	[124]	344	25.5	-	19	2	[25], [26], [46], [52], [57]
	QM9	[125]	133885	-	-	-	-	[27], [69]
	Alchemy	[126]	119487	-	-	-	-	-
Social Networks	Reddit	[42]	1	232965	11606919	602	41	[42], [48], [49], [50], [51], [56]
	BlogCatalog	[127]	1	10312	333983	-	39	[18], [55], [60], [64]
Others	MNIST	[128]	70000	784	-	1	10	[19], [23], [21], [44], [96]
	METR-LA	[129]	1	207	1515	2	-	[48], [72], [76]
	Nell	[130]	1	65755	266144	61278	210	[22], [41], [50]

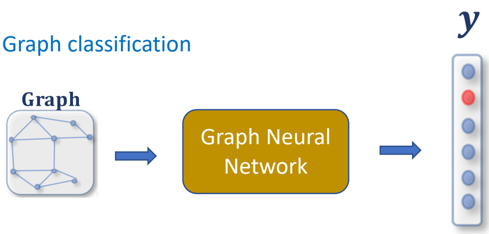
15

What is to learn

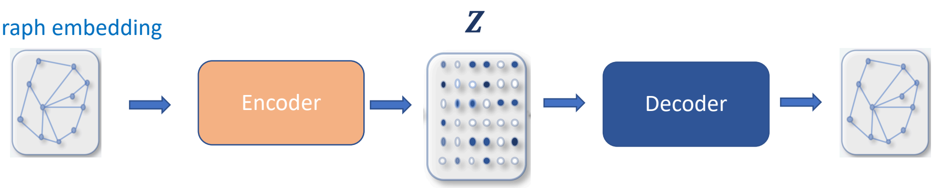
Node (or edge) classification



Graph classification



Graph embedding



Latent node representation

16

Different graph analytics tasks

- Mechanisms:
  - Node level: outputs relate to node regression and node classification tasks
    - With a multi-perceptron or a softmax layer as the output layer, GNNs are able to perform node-level tasks in an end-to-end manner
  - Edge-level: outputs relate to the edge classification and link prediction tasks
    - With two nodes' hidden representations from GNNs as inputs, a similarity function or a neural network can be utilized to predict the label/connection strength of an edge
  - Graph-level: outputs relate to the graph classification task.
    - are often combined with pooling and readout operations.

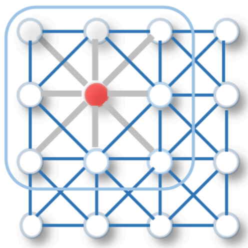
17

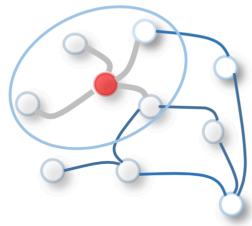
Convolutional Graph Neural Networks

ConvGNN

18

2D convolution vs Graph convolution





- A pixel = a node
- Neighborhood = filter size

→ neighbors are ordered and fixed in size

- 2D conv: takes the weighted average of pixel values along with the neighbors.

- Node
- Neighborhood = adjacency

→ Neighbors are nordered and variable in size

- Conv: take the average value of the node features along with the neighbors

19

Graph definition

Definition 1 (Graph): A graph is represented as $G = (V, E)$ where V is the set of vertices or nodes (we will use nodes throughout the paper), and E is the set of edges. Let $v_i \in V$ to denote a node and $e_{ij} = (v_i, v_j) \in E$ to denote an edge pointing from v_j to v_i. The neighborhood of a node v is defined as $N(v) = \{u \in V | (v, u) \in E\}$. The adjacency matrix $\mathbf{A}$ is a $n \times n$ matrix with $A_{ij} = 1$ if $e_{ij} \in E$ and $A_{ij} = 0$ if $e_{ij} \notin E$. A graph may have node attributes $\mathbf{X}$ [1], where $\mathbf{X} \in \mathbf{R}^{n \times d}$ is a node feature matrix with $\mathbf{x}_v \in \mathbf{R}^d$ representing the feature vector of a node v. Meanwhile, a graph may have edge attributes $\mathbf{X}^e$, where $\mathbf{X}^e \in \mathbf{R}^{m \times c}$ is an edge feature matrix with $\mathbf{x}^e_{v,u} \in \mathbf{R}^c$ representing the feature vector of an edge (v, u).

TABLE I: Commonly used notations.

Notations	Descriptions
$\lvert \cdot \rvert$	The length of a set.
$\odot$	Element-wise product.
G	A graph.
V	The set of nodes in a graph.
v	A node $v \in V$.
E	The set of edges in a graph.
e_{ij}	An edge $e_{ij} \in E$.
$N(v)$	The neighbors of a node v.
$\mathbf{A}$	The graph adjacency matrix.
$\mathbf{A}^T$	The transpose of the matrix $\mathbf{A}$.
$\mathbf{A}^n, n \in Z$	The n^{th} power of $\mathbf{A}$.
$[\mathbf{A}, \mathbf{B}]$	The concatenation of $\mathbf{A}$ and $\mathbf{B}$.
$\mathbf{D}$	The degree matrix of $\mathbf{A}$. $\mathbf{D}_{ii} = \sum_{j=1}^{n} \mathbf{A}_{ij}$.
n	The number of nodes, $n = \lvert V \rvert$.
m	The number of edges, $m = \lvert E \rvert$.
d	The dimension of a node feature vector.
b	The dimension of a hidden node feature vector.
c	The dimension of an edge feature vector.
$\mathbf{X} \in \mathbf{R}^{n \times d}$	The feature matrix of a graph.
$\mathbf{x} \in \mathbf{R}^n$	The feature vector of a graph in the case of $d = 1$.
$\mathbf{x}_v \in \mathbf{R}^d$	The feature vector of the node v.
$\mathbf{X}^e \in \mathbf{R}^{m \times c}$	The edge feature matrix of a graph.
$\mathbf{x}^e_{(v,u)} \in \mathbf{R}^c$	The edge feature vector of the edge (v, u).
$\mathbf{X}^{(t)} \in \mathbf{R}^{n \times d}$	The node feature matrix of a graph at the time step t.
$\mathbf{H} \in \mathbf{R}^{n \times b}$	The node hidden feature matrix.
$\mathbf{h}_v \in \mathbf{R}^b$	The hidden feature vector of node v.
k	The layer index
t	The time step/iteration index
$\sigma(\cdot)$	The sigmoid activation function.
$\sigma_h(\cdot)$	The tangent hyperbolic activation function.
$\mathbf{W}, \mathbf{\Theta}, w, \theta$	Learnable model parameters.

20

Architecture – NN4G

A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.

T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. of ICLR*, 2017.

Next layer node states:

$$\mathbf{H}^{(k)} = f(\mathbf{X}\mathbf{W}^{(k)} + \sum_{i=1}^{k-1} \mathbf{A}\mathbf{H}^{(k-1)}\mathbf{\Theta}^{(k)})$$

Update rules vary between models!

Node hidden feature matrix

Feature matrix

Adjacency matrix

Activation function

Learnable model parameters

21

Graph pooling modules

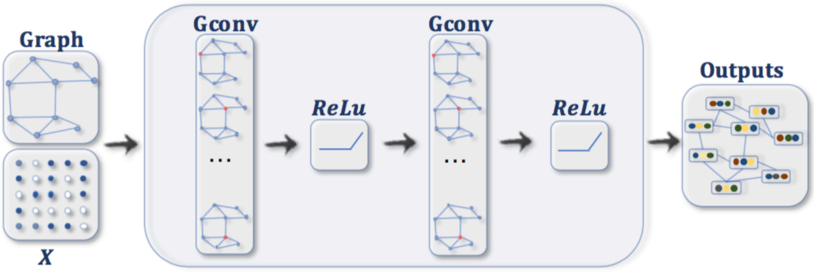
- Objectives:
  - To reduce the size of parameters by down-sampling the nodes to generate smaller representations
    - Avoiding overfitting, permutation invariance, and computational complexity
  - To generate graph-level representation based on node representation
    - Readout phase

$$\mathbf{h}_G = mean/max/sum(\mathbf{h}_1^{(K)}, \mathbf{h}_2^{(K)}, ..., \mathbf{h}_n^{(K)})$$

- Pooling method:
  - Mean/max/sum
  - SortPooling → rearranging nodes to a meaningful order (DGCNN)
  - DiffPooling → considering structural information of graph
  - SAGPool→ consider both node & graph topology + self-attention

22

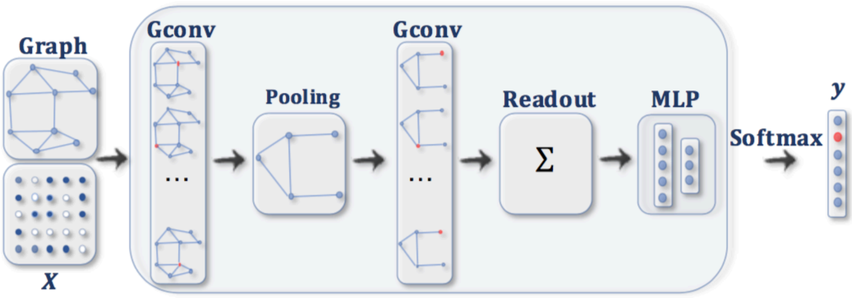
Node-level learning



(a) A ConvGNN with multiple graph convolutional layers. A graph convolutional layer encapsulates each node's hidden representation by aggregating feature information from its neighbors. After feature aggregation, a non-linear transformation is applied to the resulted outputs. By stacking multiple layers, the final hidden representation of each node receives messages from a further neighborhood.

23

Graph classification



(b) A ConvGNN with pooling and readout layers for graph classification [21]. A graph convolutional layer is followed by a pooling layer to coarsen a graph into sub-graphs so that node representations on coarsened graphs represent higher graph-level representations. A readout layer summarizes the final graph representation by taking the sum/mean of hidden representations of sub-graphs.

24

Some ConvGNNs

NN4G (2009) [24]	Spatial-based ConvGNN	A, X	-	sum/mean	$O(m)$
DCNN (2016) [25]	Spatial-based ConvGNN	A, X	-	mean	$O(n^2)$
PATCHY-SAN (2016) [26]	Spatial-based ConvGNN	A, X, X^e	-	sum	-
MPNN (2017) [27]	Spatial-based ConvGNN	A, X, X^e	-	attention sum/set2set	$O(m)$
GraphSage (2017) [42]	Spatial-based ConvGNN	A, X	-	-	-
GAT (2017) [43]	Spatial-based ConvGNN	A, X	-	-	$O(m)$
MoNet (2017) [44]	Spatial-based ConvGNN	A, X	-	-	$O(m)$
LGCN (2018) [45]	Spatial-based ConvGNN	A, X	-	-	-
PGC-DGCNN (2018) [46]	Spatial-based ConvGNN	A, X	sort pooling	attention sum	$O(n^3)$
CGMM (2018) [47]	Spatial-based ConvGNN	A, X, X^e	-	sum	-
GAAN (2018) [48]	Spatial-based ConvGNN	A, X	-	-	$O(m)$
FastGCN (2018) [49]	Spatial-based ConvGNN	A, X	-	-	-
StoGCN (2018) [50]	Spatial-based ConvGNN	A, X	-	-	-
Huang et al. (2018) [51]	Spatial-based ConvGNN	A, X	-	-	-
DGCNN (2018) [52]	Spatial-based ConvGNN	A, X	sort pooling	-	$O(m)$
DiffPool (2018) [54]	Spatial-based ConvGNN	A, X	differential pooling	mean	$O(n^2)$
GeniePath (2019) [55]	Spatial-based ConvGNN	A, X	-	-	$O(m)$
DGI (2019) [56]	Spatial-based ConvGNN	A, X	-	-	$O(m)$
GIN (2019) [57]	Spatial-based ConvGNN	A, X	-	sum	$O(m)$
ClusterGCN (2019) [58]	Spatial-based ConvGNN	A, X	-	-	-

25

Open-source implementations

- PyTorch Geometric
  - https://github.com/rusty1s/pytorch_geometric
- Deep Graph Library
  - https://www.dgl.ai/

26