《图卷积神经网络》

主讲： 彭老师

https://www.julyedu.com/

七月在线
JULYEDU.COM

图卷积神经网络

Graph Convolution Networks

GNN的基准化

Benchmarking Graph Neural Networks

/01 Graph Convolution Networks

图卷积神经网络

Dataset

- KarateClub：数据为无向图，来源于论文An Information Flow Model for Conflict and Fission in Small Groups；
- TUDataset：包括58个基础的分类数据集集合，数据都为无向图，如“IMDB-BINARY”，“PROTEINS”等，来源于TU Dortmund University
- Planetoid：引用网络数据集，包括“Cora”, “CiteSeer” and “PubMed”，数据都为无向图，来源于论文Revisiting Semi-Supervised Learning with Graph Embeddings。节点代表文档，边代表引用关系。
- CoraFull：完整的“Cora”引用网络数据集，数据为无向图，来源于论文Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking。节点代表文档，边代表引用关系。

Dataset

- Coauthor：共同作者网络数据集，包括“CS”和“Physics”，数据都为无向图，来源于论文Pitfalls of Graph Neural Network Evaluation。节点代表作者，若是共同作者则被边相连。学习任务是将作者映射到各自的研究领域中。

- Amazon：亚马逊网络数据集，包括“Computers”和“Photo”，数据都为无向图，来源于论文Pitfalls of Graph Neural Network Evaluation。节点代表货物，边代表两种货物经常被同时购买。学习任务是将货物映射到各自的种类里。

- PPI：蛋白质-蛋白质反应网络，数据为无向图，来源于论文Predicting multicellular function through multi-layer tissue networks

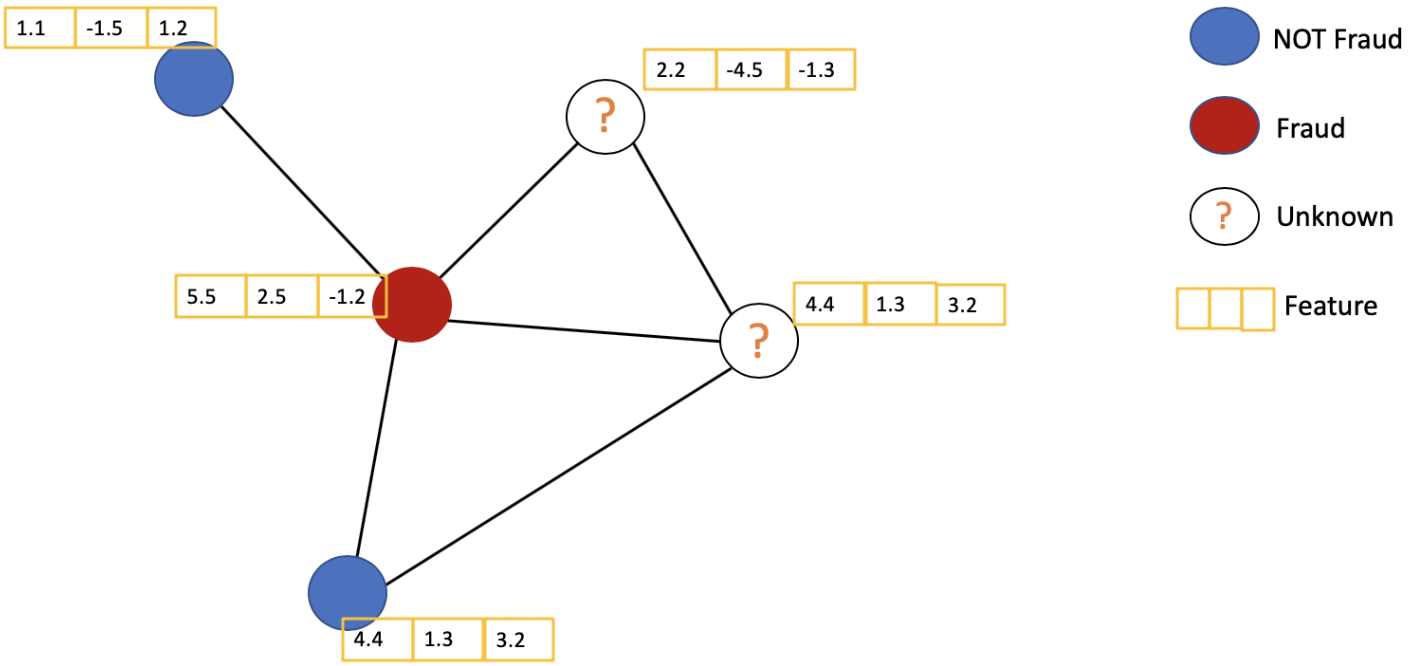
- Entities：关系实体网络，包括“AIFB”，“MUTAG”，“BGS”和“AM”，数据都为无向图，来源于论文Modeling Relational Data with Graph Convolutional Networks

- BitcoinOTC：数据为有向图，包括138个“who-trusts-whom”网络，来源于论文EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs，数据链接为Bitcoin OTC trust weighted signed network

Graph Convolution Networks

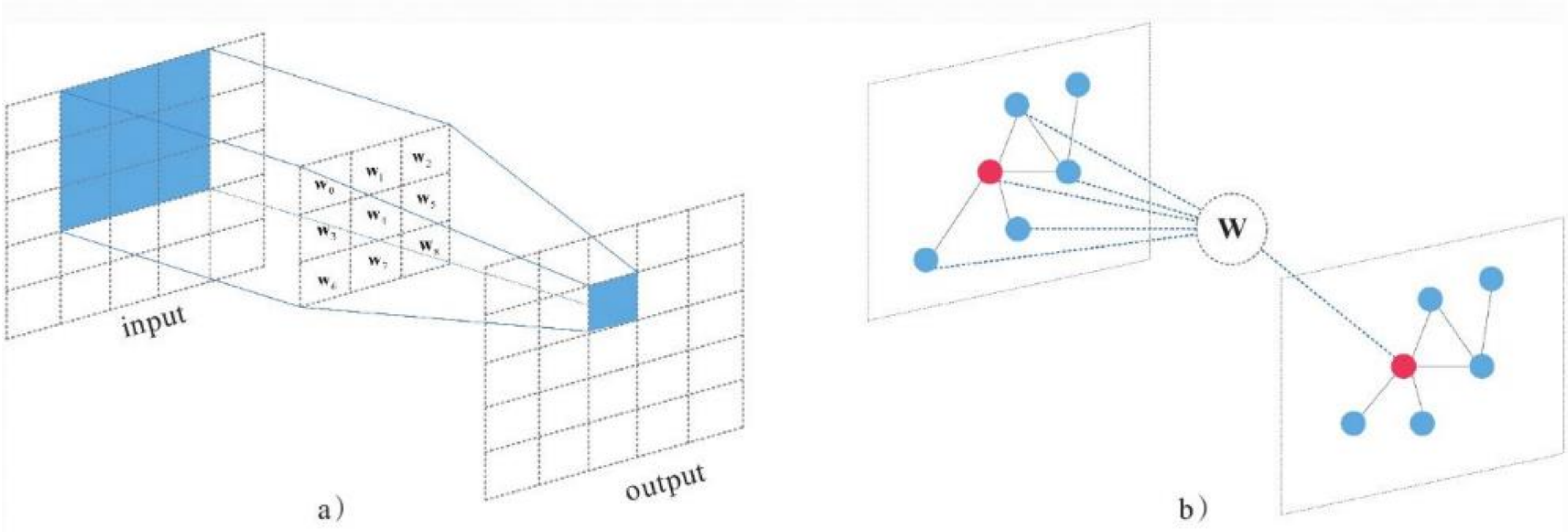
Graph Convolution Networks (**GCN**) is a type of convolutional neural network that can work directly on graphs and take advantage of their structural information.

GCN solves the problem of **node classification** (such as documents) in a graph (such as a citation network), where labels are only available for a small subset of nodes (semi-supervised learning).



Comparison between CNN and GNN

The layer of a Graph Convolutional Neural Network (GCN) :

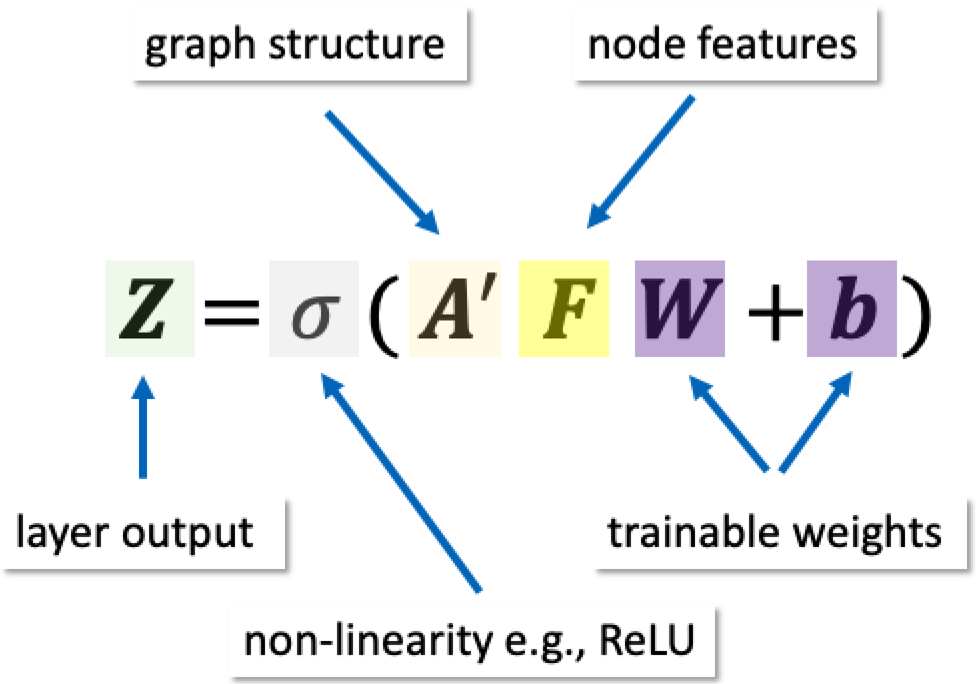


a)

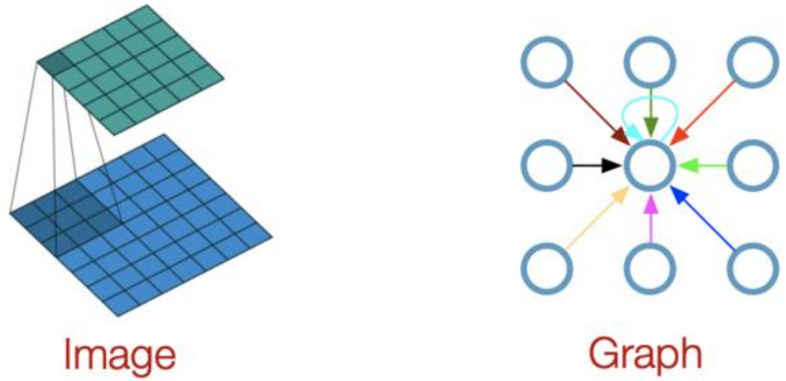
b)

Comparison between CNN and GNN

The layer of a Graph Convolutional Neural Network (GCN) :



Single CNN layer with 3x3 filter:



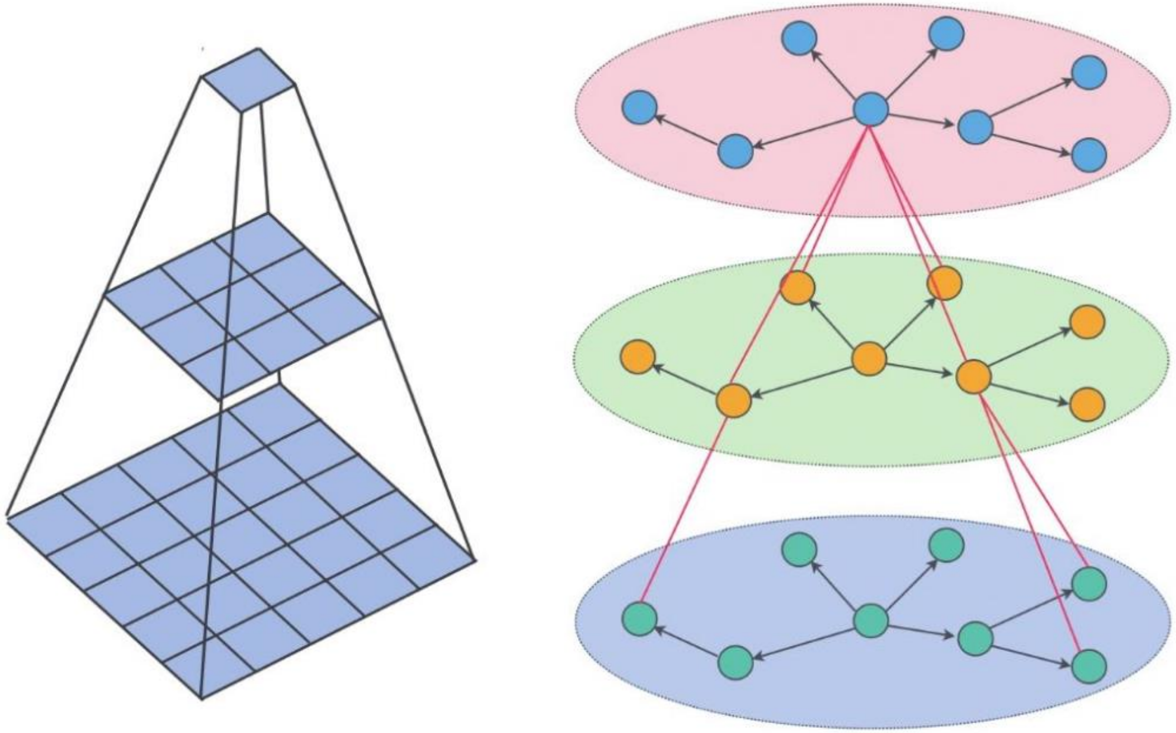
$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

Comparison between CNN and GNN

Receptive Field：

Left: CNN

Right: GNN

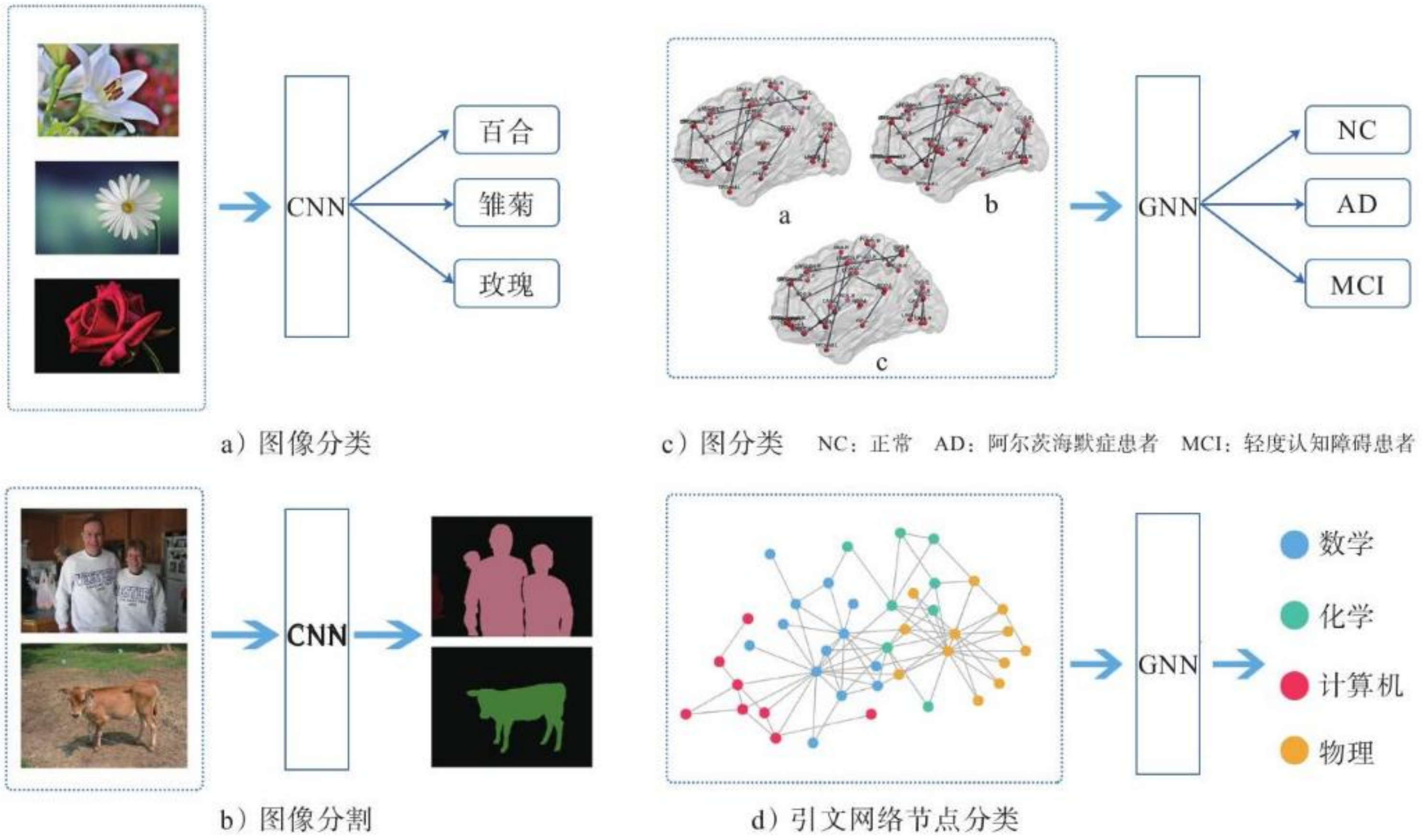


a)　　　　b)

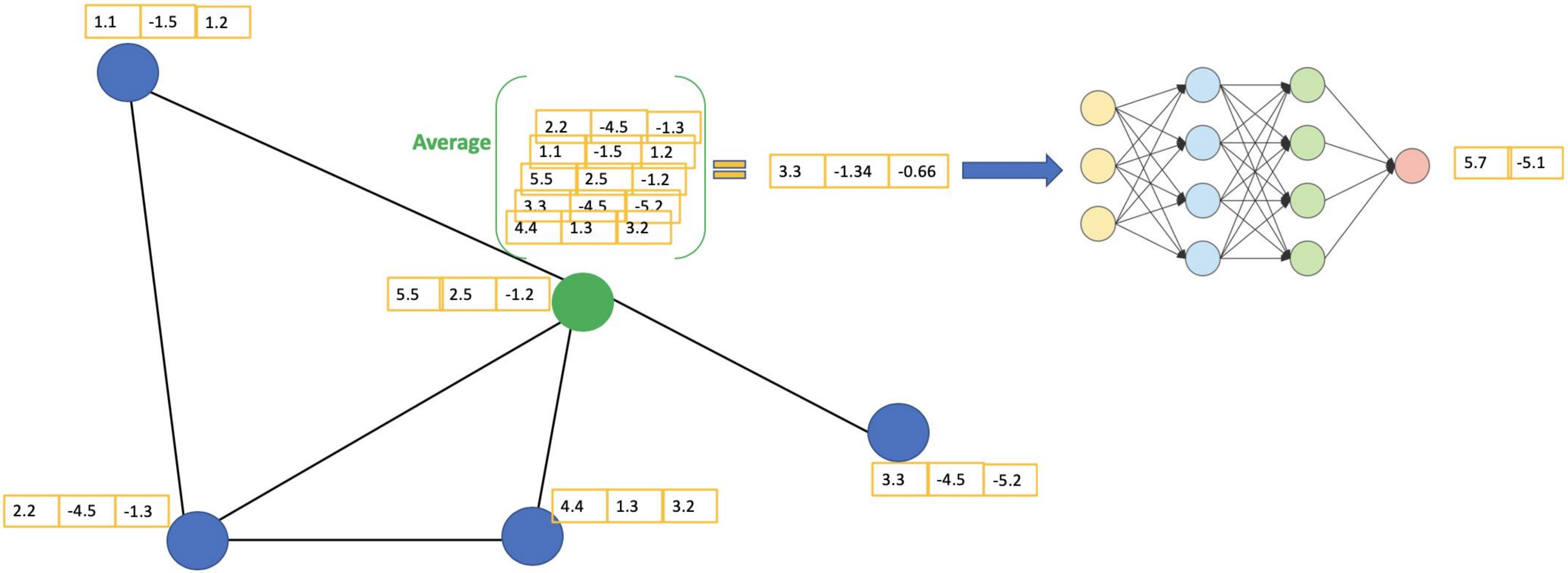
Comparison between CNN and GNN

Tasks：

Left: CNN
Right: GNN

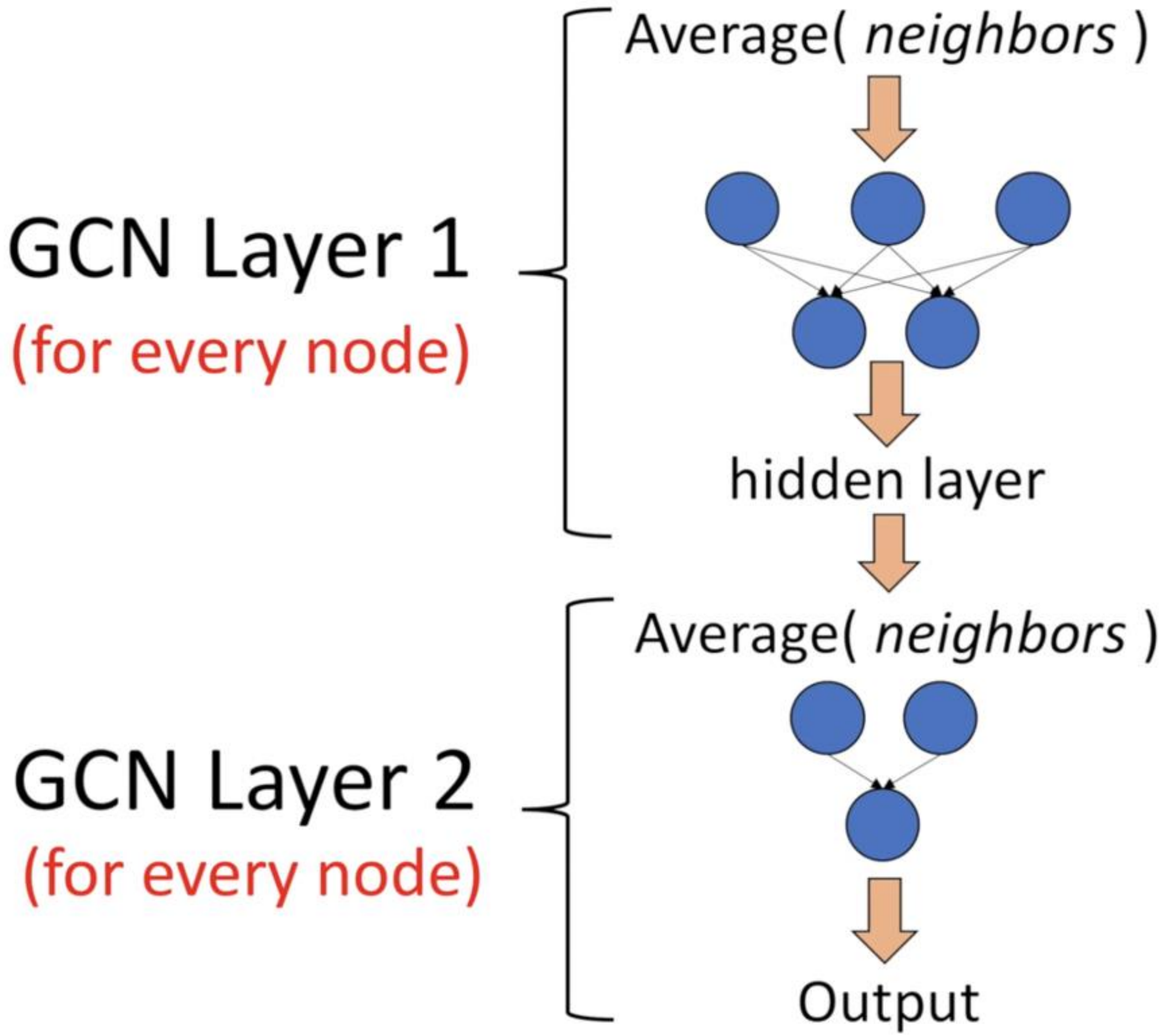


Main Idea of GCN

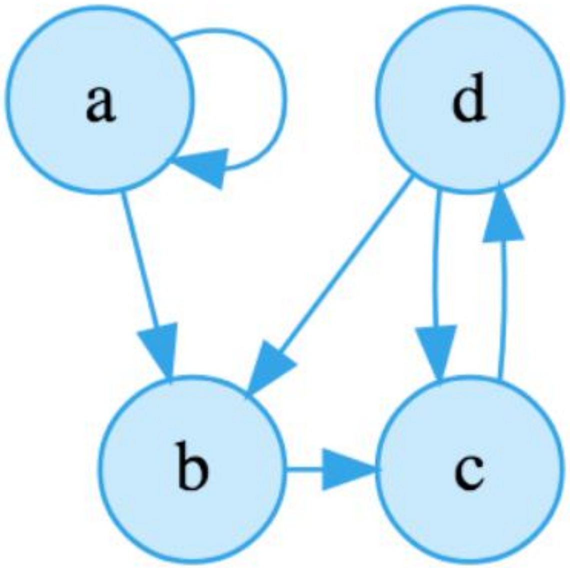


The feature values of the green node’s neighbors, including itself, then take the average. The result will be passed through a neural network to return a resulting vector.

Main Idea of GCN



Adjacency List



Adjacency List

```
a -> { a b }
b -> { c }
c -> { d }
d -> { b c }
```

Adjacency Matrix

Let G be a graph with "n" nodes that are assumed to be ordered from v_1 to v_n.
The $n \times n$ matrix A is called an **adjacency matrix**, which satisfies:

$$\begin{cases} a_{ij} = 1 \text{ if there exists a path from node } i \text{ to node } j; \\ a_{ij} = 0 \text{ otherwise.} \end{cases}$$

$A_{ij} = 1$ if there is a link from node ***i*** to node ***j***

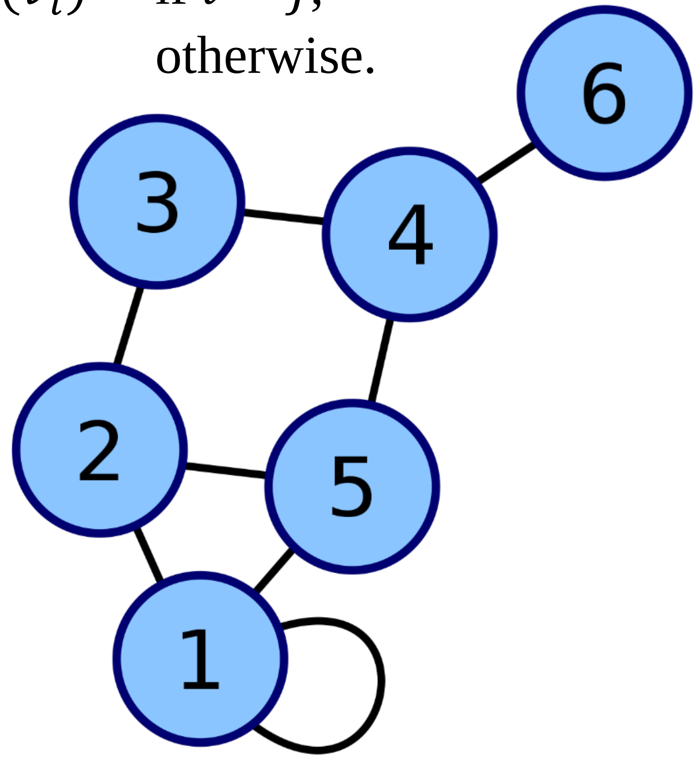
$A_{ij} = 0$ otherwise

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Degree Matrix

Let G be a graph with "n" nodes that are assumed to be ordered from v_1 to v_n.
The $n \times n$ matrix D is called a **degree matrix**, which satisfies:

$$\begin{cases} d_{ij} = deg(v_i) & \text{if } i = j; \\ a_{ij} = 0 & \text{otherwise.} \end{cases}$$



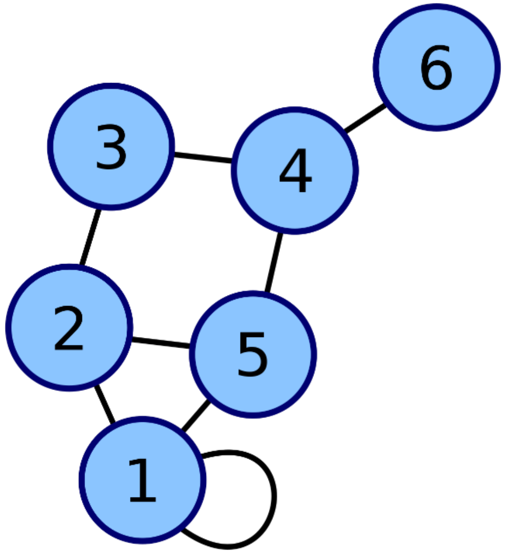
$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Degree Matrix

Where the degree $deg(v_i)$ of a vertex counts the number of times an edge terminates at that vertex.

In an undirected graph, this means that each **loop** increases the degree of a vertex by **two**.

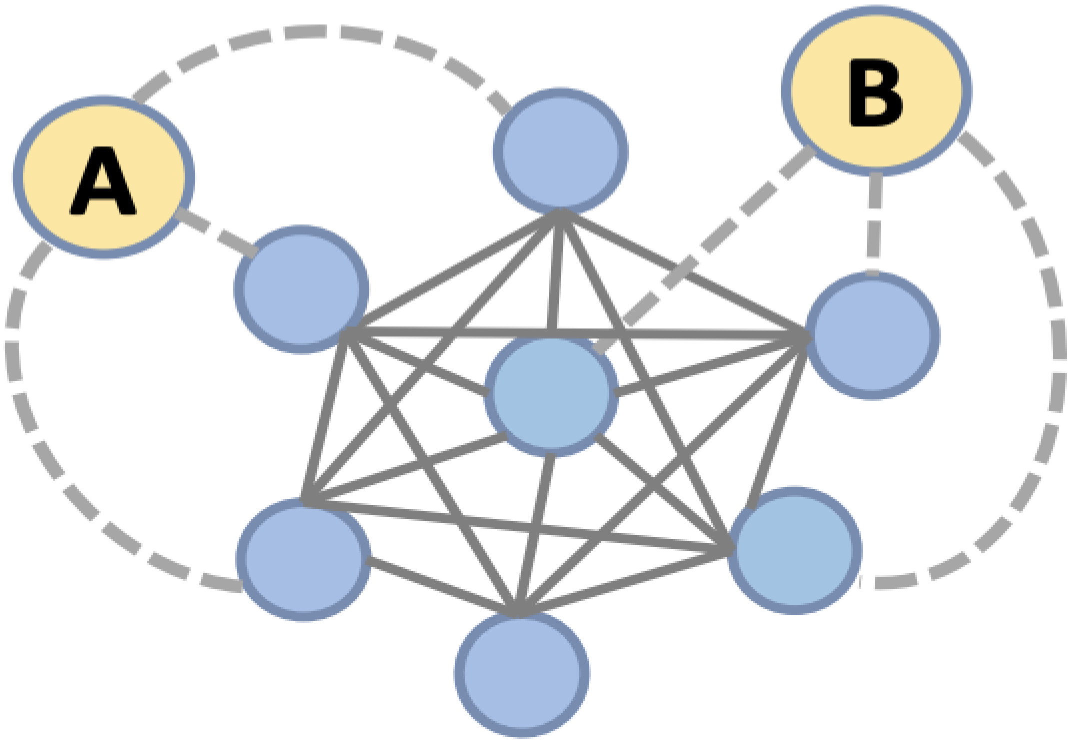
In a directed graph, the term degree may refer either to indegree (the number of incoming edges at each vertex) or outdegree (the number of outgoing edges at each vertex).



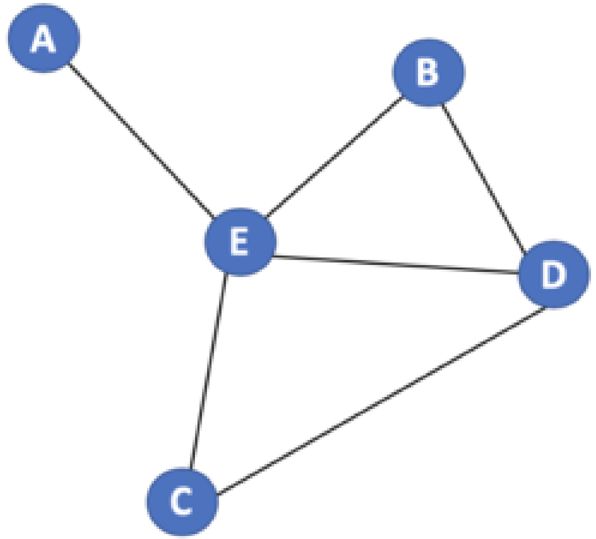
$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Massage Passing

What is the relationship between A and B?



Massage Passing



Graph G

	A	B	C	D	E
A	0	0	0	0	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	1	1	0	1
E	1	1	1	1	0

Adjacency matrix A

	A	B	C	D	E
A	1	0	0	0	0
B	0	2	0	0	0
C	0	0	2	0	0
D	0	0	0	3	0
E	0	0	0	0	4

Degree matrix D

A	-1.1	3.2	4.2
B	0.4	5.1	-1.2
C	1.2	1.3	2.1
D	1.4	-1.2	2.5
E	1.4	2.5	4.5

Feature vector X

Massage Passing： Update Adjacency Matrix

$$\tilde{A} = A + \lambda I$$

	A	B	C	D	E
A	0	0	0	0	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	1	1	0	1
E	1	1	1	1	0

Adjacency matrix A

+

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Identity matrix I

=

	A	B	C	D	E
A	1	0	0	0	1
B	0	1	0	1	1
C	0	0	1	1	1
D	0	1	1	1	1
E	1	1	1	1	1

New Adjacency matrix $\tilde{A}$

Massage Passing： Update Degree Matrix

	A	B	C	D	E
A	1	0	0	0	1
B	0	1	0	1	1
C	0	0	1	1	1
D	0	1	1	1	1
E	1	1	1	1	1

New adjacency matrix $\tilde{A}$

→

2	0	0	0	0
0	3	0	0	0
0	0	3	0	0
0	0	0	4	0
0	0	0	0	5

New degree matrix $\tilde{D}$

→

1/2	0	0	0	0
0	1/3	0	0	0
0	0	1/3	0	0
0	0	0	1/4	0
0	0	0	0	1/5

$\tilde{D}^{-1}$

Massage Passing: Scale Factor

We consider $(\widetilde{D}^{-1}\tilde{A})X$, so $\widetilde{D}^{-1}$ be the **scale factor** of $\tilde{A}$.
From this perspective, each row i of $\tilde{A}$ will be scaled by $\widetilde{D}_{ii}$.

1/2	0	0	0	0
0	1/3	0	0	0
0	0	1/3	0	0
0	0	0	1/4	0
0	0	0	0	1/5

$\widetilde{D}^{-1}$

	A	B	C	D	E
A	1	0	0	0	1
B	0	1	0	1	1
C	0	0	1	1	1
D	0	1	1	1	1
E	1	1	1	1	1

New adjacency matrix $\tilde{A}$

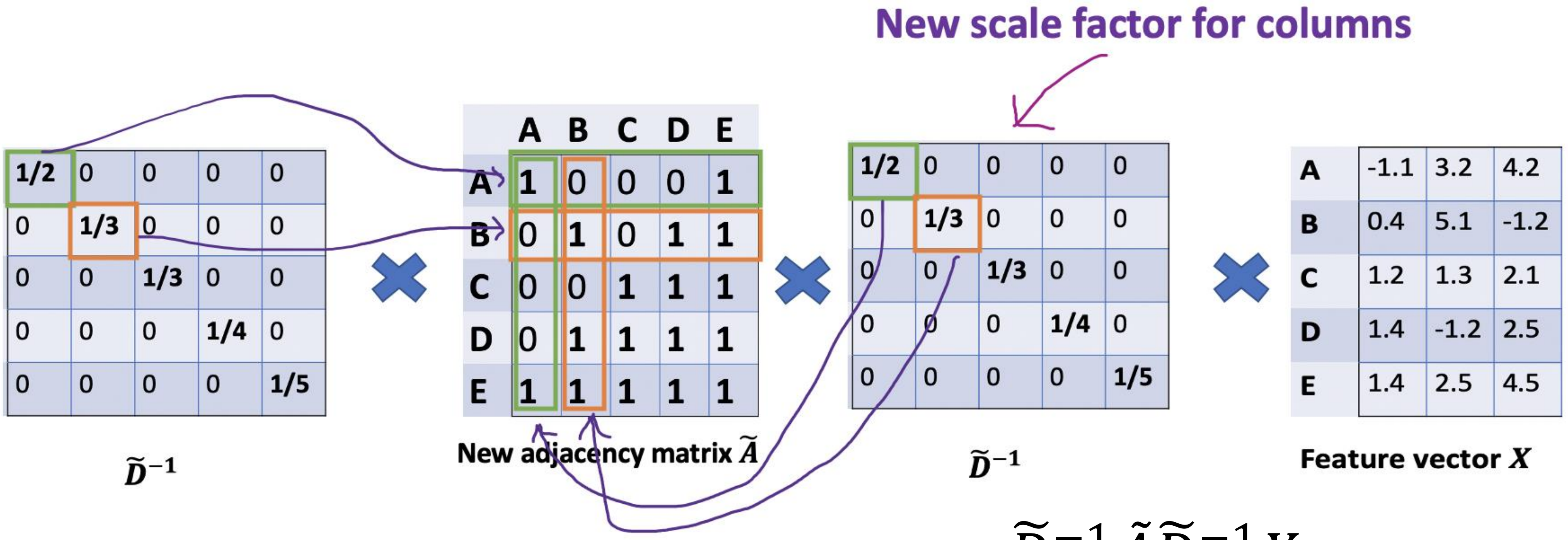
A	-1.1	3.2	4.2
B	0.4	5.1	-1.2
C	1.2	1.3	2.1
D	1.4	-1.2	2.5
E	1.4	2.5	4.5

Feature vector X

$$(\widetilde{D}^{-1}\tilde{A})X$$

Massage Passing: Scale Factor

From this perspective, each row i of $\tilde{A}$ will be **scaled** by $\widetilde{D}_{ii}$.
$\tilde{A}$ is a symmetric matrix, it means row i is the same value as column i.
If we scale each row i of $\tilde{A}$ by $\widetilde{D}_{ii}$, intuitively, we have a feeling that we should do the *same* for its corresponding **column**.

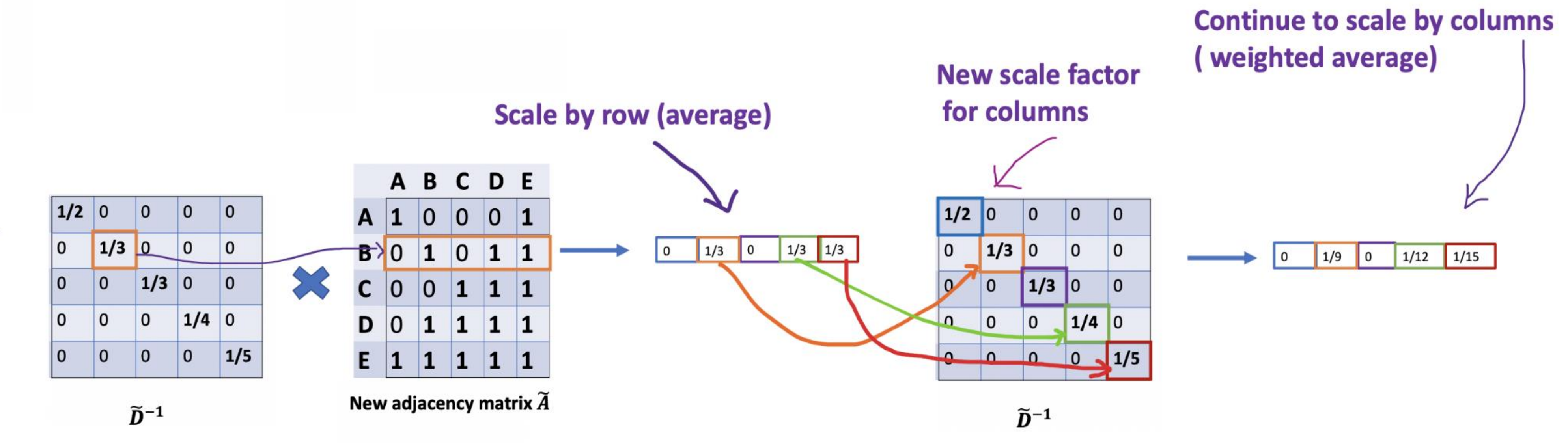


$$\widetilde{D}^{-1}\tilde{A}\widetilde{D}^{-1}X$$

Massage Passing: Scale Factor

The new scaler gives us the **“weighted” average**.

The idea of this weighted average is that we assume low-degree nodes would have bigger impacts on their neighbors, whereas high-degree nodes generate lower impacts as they scatter their influence at too many neighbors.



$$\tilde{D}^{-1}\tilde{A}\tilde{D}^{-1}X$$

Massage Passing: Scale Factor

When using two scalers ($\widetilde{D}_{ii}$ and $\widetilde{D}_{jj}$), we normalize twice: one for row, the other for column.

It will make sense if we rebalance by modifying $\widetilde{D}_{ii}\widetilde{D}_{jj}$ to $(\widetilde{D}_{ii}\widetilde{D}_{jj})^{1/2}$

2	0	0	0	0
0	3	0	0	0
0	0	3	0	0
0	0	0	4	0
0	0	0	0	5

$\widetilde{D}$

1/2	0	0	0	0
0	1/3	0	0	0
0	0	1/3	0	0
0	0	0	1/4	0
0	0	0	0	1/5

$\widetilde{D}^{-1}$

$1/\sqrt{2}$	0	0	0	0
0	$1/\sqrt{3}$	0	0	0
0	0	$1/\sqrt{3}$	0	0
0	0	0	1/2	0
0	0	0	0	$1/\sqrt{5}$

$\widetilde{D}^{-1/2}$

$$\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{\frac{1}{2}}X$$

Massage Passing: Update Degree Matrix

Quick summary so far:

$\tilde{A}X$: sum of all neighbors' feature vectors, including itself.

$\tilde{D}^{-1}\tilde{A}X$: average of all neighbors' feature vectors (including itself). Adjacency matrix is scaled by rows

$\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}X$: average of all neighbors' feature vectors (including itself). The adjacency matrix is scaled by both rows and columns. By doing this, we get the weighted average preferring on low-degree nodes.

Massage Passing： Update Degree Matrix

Ok, now let's put things together.

Let's call $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ just for a clear view. With 2-layer GCN, we have the form of our forward model as below.

Feature vector matrix (***N***x***C***)

$$Z = f(X, A) = \mathrm{softmax}\Big(\hat{A}\ \boxed{\mathrm{ReLU}\big(\hat{A}XW^{(0)}\big)}\ W^{(1)}\Big)$$

Trainable weights (***H***x***F***)

Trainable weights (***C***x***H***)

First layer

Scaled adjacency matrix (***N***x***N***)

Recall that ***N*** is #nodes, ***C*** is #dimensions of feature vectors. We also have ***H*** is #nodes in the hidden layer, and ***F*** is the dimensions of resulting vectors.

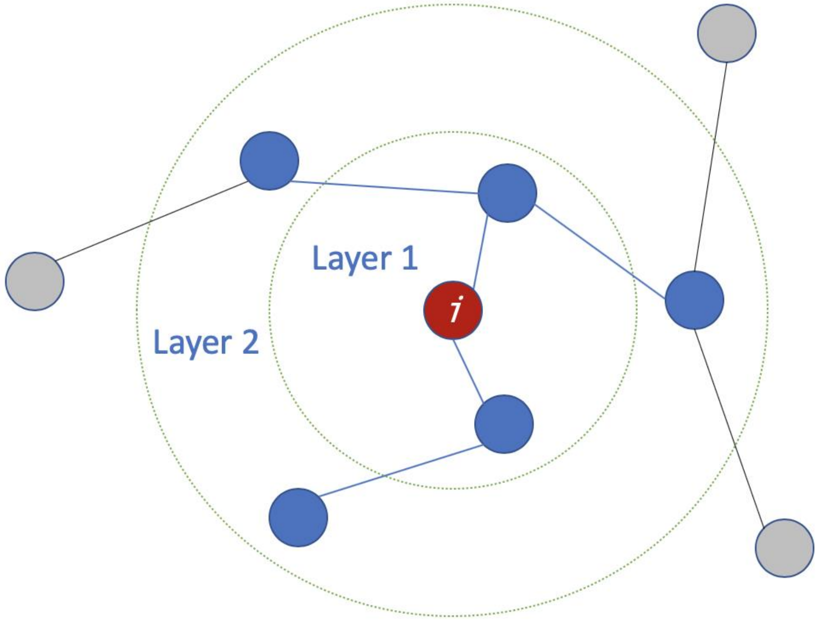
Loss Function

The Loss function is simply calculated by the **cross-entropy error** over all labeled examples.

Y_l is the set of node indices that have labels.

$$\mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

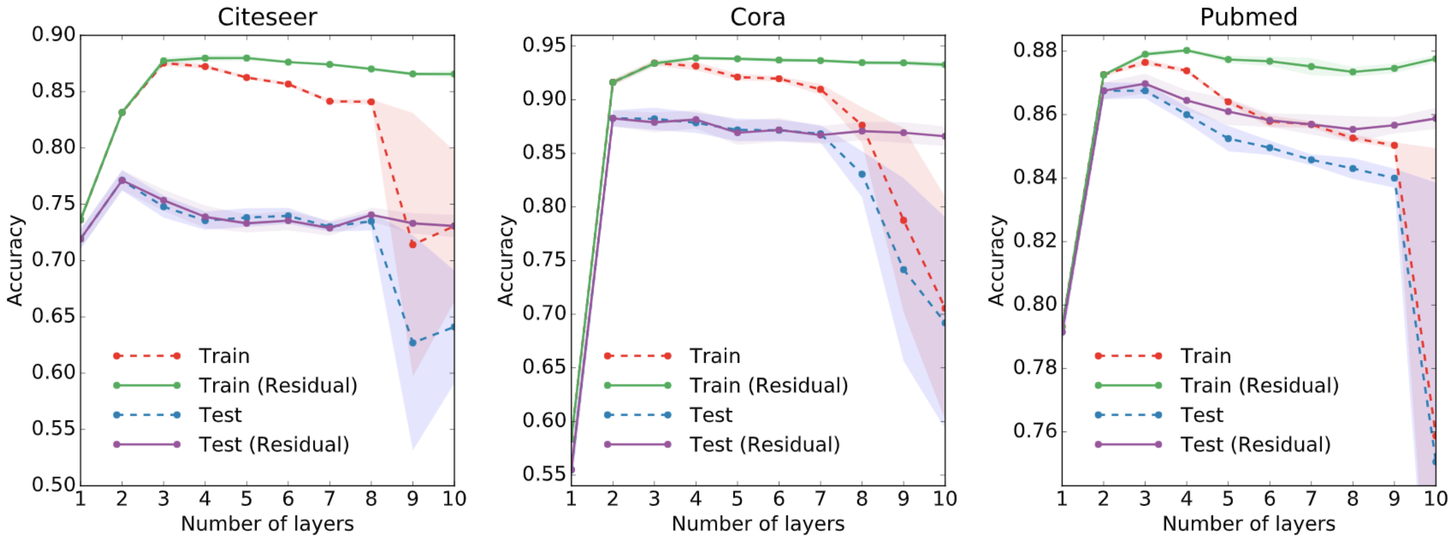
Layers



The number of layers is the farthest distance that node features can travel.

Each node can only get the information from its neighbors with 1-layer GCN.

The gathering information process takes place independently, at the same time for all the nodes.



/02 Benchmarking Graph Neural Networks

GNN的基准化

Benchmarking Graph Neural Networks

Issues:
GNN has been increasingly difficult to gauge the effectiveness of new GNNs and compare models in the **absence** of a standardized benchmark with consistent experimental settings and large datasets.

Solution:
The authors propose a reproducible **GNN benchmarking framework**, with the facility for researchers to add new datasets and models conveniently.

Applications:
The benchmarking framework can be applied to novel medium-scale graph datasets from mathematical modeling, computer vision, chemistry and combinatorial problems to establish key operations when designing effective GNNs.

More:
Precisely, graph convolutions, anisotropic diffusion, residual connections and normalization layers are universal building blocks for developing robust and scalable GNNs.

Benchmarking Graph Neural Networks

Questions Proposed：

- ✓ How to build powerful GNNs has become central?
- ✓ How to study and quantify the impact of theoretical developments for GNNs?

Benchmarking Graph Neural Networks

Contributions:：

I. We release an **open benchmark infrastructure** for GNNs, hosted on GitHub based on PyTorch and DGL libraries;

II. We aim to go beyond the popular but small CORA and TU datasets by **introducing medium-scale datasets** *with 12k-70k graphs of variable sizes 9-500 nodes*. Proposed datasets are from mathematical modeling (Stochastic Block Models), computer vision (super-pixels), combinatorial optimization (Traveling Salesman Problem) and chemistry (molecules' solubility).

III. We identify important **building blocks** of GNNs with the proposed benchmark infrastructure. Graph convolutions, anisotropic diffusion, residual connections, and normalization layers stick out as most useful to design efficient GNNs.

Benchmarking Graph Neural Networks

Contributions: :

IV. We fix a parameter budget for all models and analyze performance trends to identify the important **GNN mechanisms**.

V. The numerical results are entirely **reproducible**. We make it simple to reproduce the reported results by running scripts. Besides, the installation and execution of the benchmark infrastructure are explained in detail in the GitHub repository.

https://github.com/graphdeeplearning/benchmarking-gnns

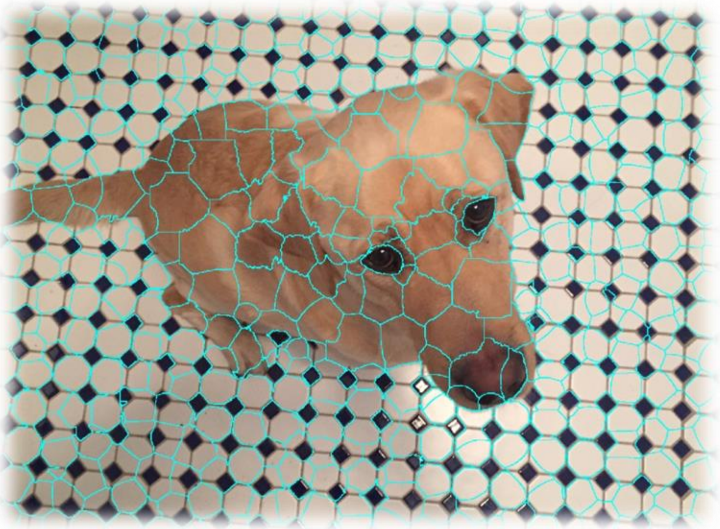
Superpixels

A superpixel can be defined as a **group** of pixels that share common characteristics (like pixel intensity).

- Superpixels carry more information than pixels.

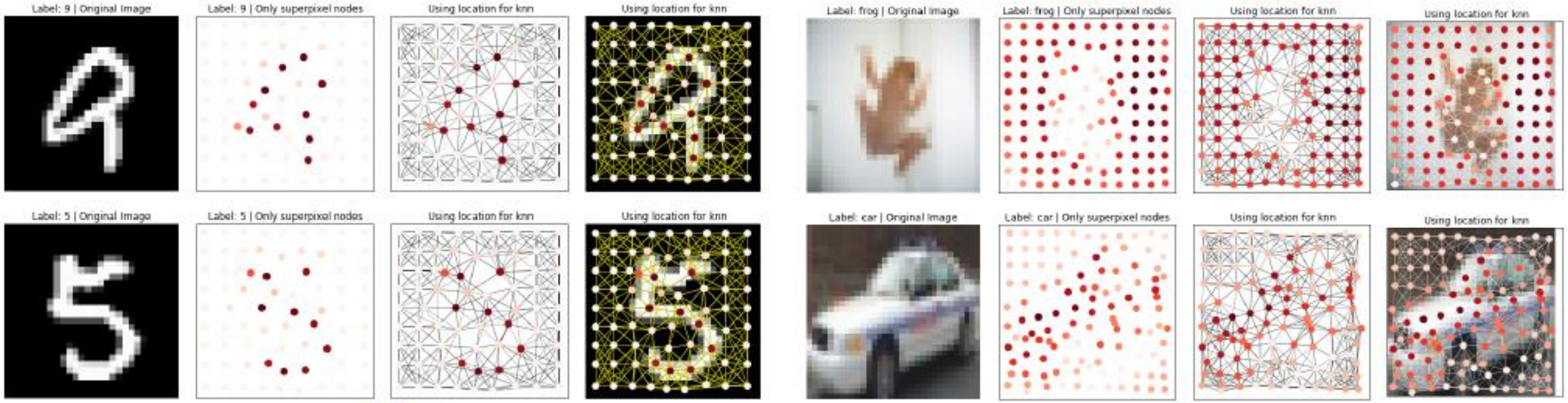
- Superpixels have a perceptual meaning since pixels belonging to a given superpixel share similar visual properties.

- Superpixels provide a convenient and compact representation of images that can be very useful for computationally demanding problems.



https://zhouyifan.net/2020/01/30/DIP-image-segmentation-project/

Superpixels



(a) MNIST (b) CIFAR10

Sample images and their superpixel graphs. The graphs of SLIC superpixels (at most 75 nodes for MNIST and 150 nodes for CIFAR10) are 8-nearest neighbor graphs in the Euclidean space and node colors denote the mean pixel intensities.

微信扫一扫关注我们

THANKS

Speaker name and title

https://www.julyedu.com

七月在线
JULYEDU.COM