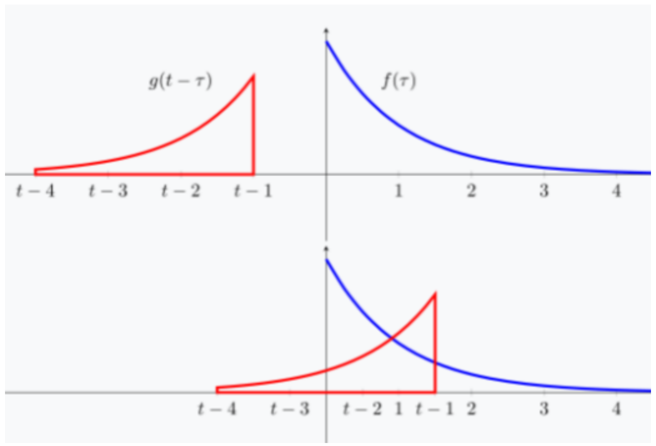
Summary Questions of the lecture

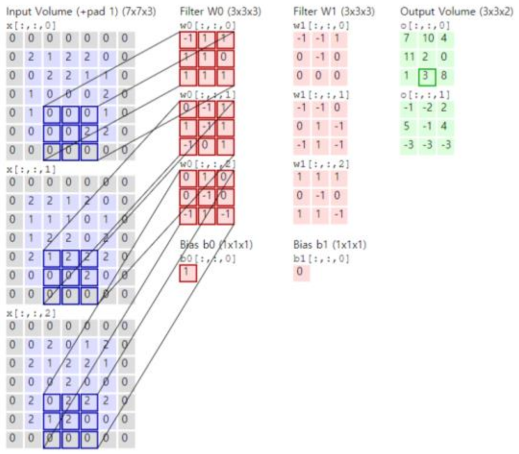
- Explain the concept of convolution sum and its meaning in the spatial and temporal applications.

→ Convolution is an operation on two functions, which is defined as the sum of the product of the two functions after one is reversed and shifted by the time index. The convolution result represents the degree of cross correlation between the two signals in view of local-in-time or local-in-space.

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[n-m]g[m]$$





Summary Questions of the lecture

- What problems in the conventional Laplacian smoothness applications does the GCN try to solve?

→ GCN tries to mitigate the inefficiency arising from redundancy by high dimension of input data space via feature embedding in GCN and the dependency of the fixed graph structure via attentional aggregation or diffusion, etc. in graph.

Summary Questions of the lecture

- Explain the two main operations in GCN.

→ One is graph filtering for node feature embedding (node feature representation learning) and the other is graph pooling to obtain a small graph from an original large graph.

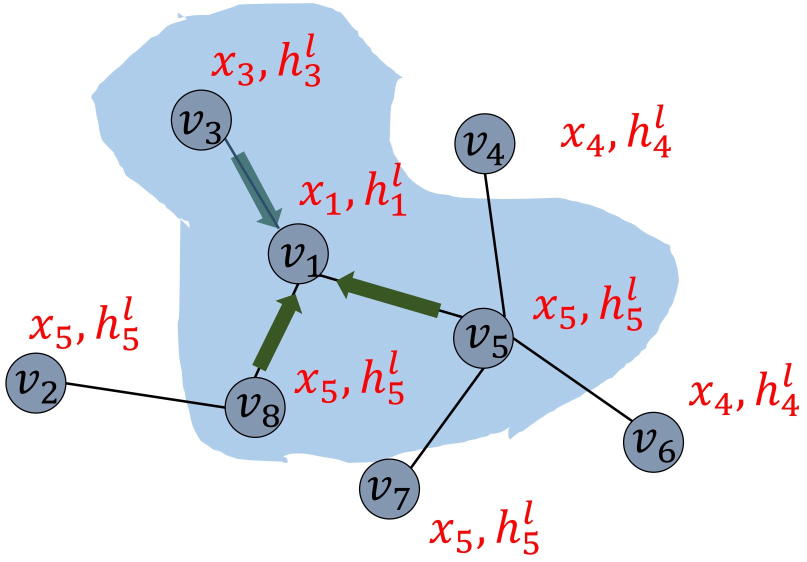
Outline of Lecture (3)

- Graph Convolution Networks (GCN)
  - What are issues on GCN
  - Graph Filtering in GCN
  - Graph Pooling in GCN
- Original GNN (Scarselli et al. 2005)
- Spectral GCN
  - Spectral Filtering
  - Graph Spectral Filtering in GCN
  - Spectral Graph CNN (Bruna et al. ICLR 2014)
  - GraphSage (Hamilton et al. NIPS 2017)
- Spatial GCN
  - GCN (Kipf & Welling. ICLR 2017)
  - GAT (Veličković et al. ICLR 2018)
  - MPNN (Glimer et al. ICML 2017)
- Link Analysis
  - PageRank
  - Diffusion
- Propagation using graph diffusion
  - Predict Then Propagate [ICLR'19]
  - Graph Diffusion-Embedding Networks [CVPR'19]
- Making a new graph
  - Diffusion Improves Graph Learning [NIPS'19]
  - Graph Learning-Convolutional Nets. [CVPR'19]

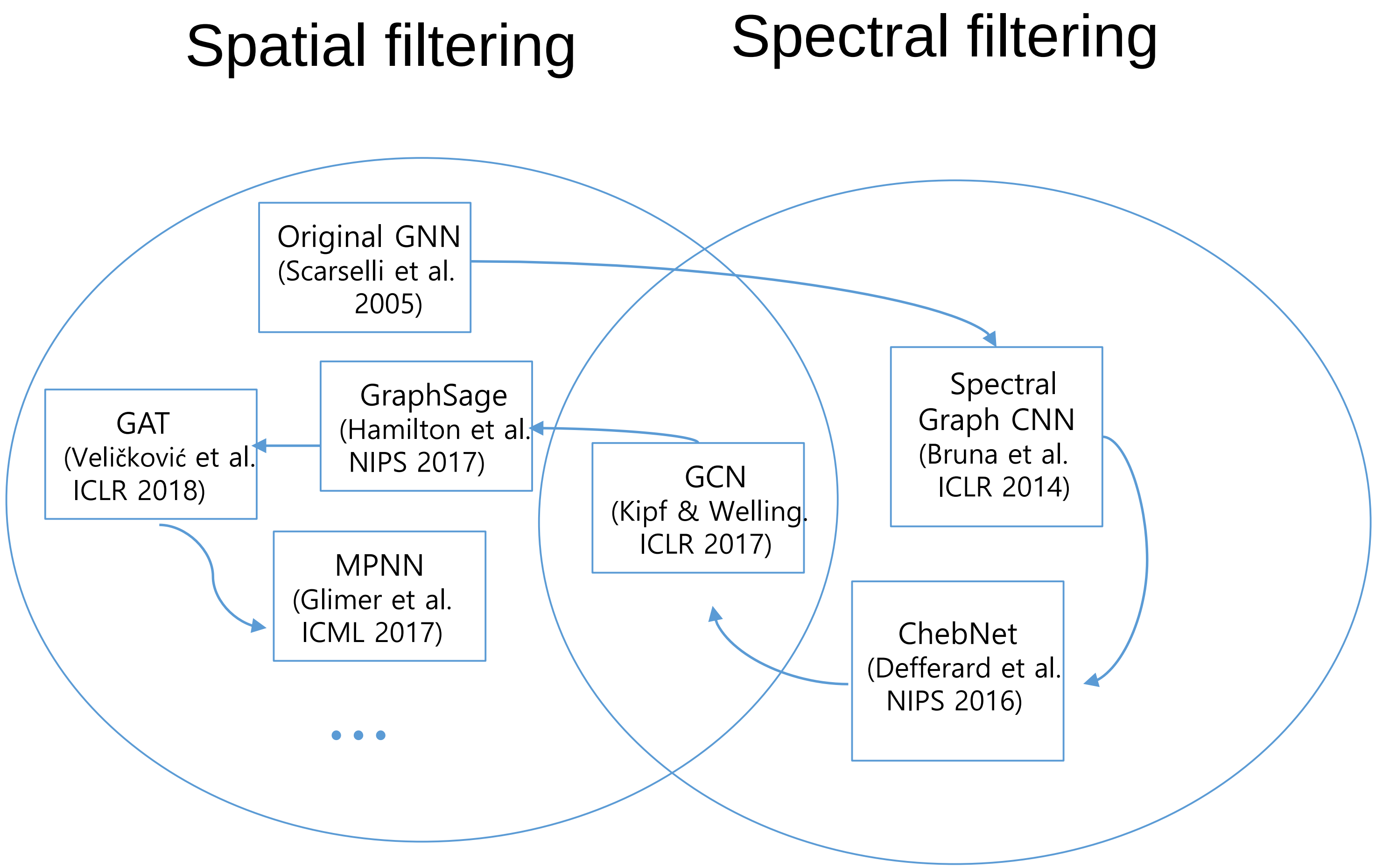
$GCN(\mathcal{G})$

Geometric deep learning via graph convolution … (continue)

$$\boldsymbol{h}_i^{(l+1)} = \sum_{v_j \in N(v_i)} \boldsymbol{f}\left(\boldsymbol{x}_i, \boldsymbol{w}_{ij}, \boldsymbol{h}_j^{(l)}, \boldsymbol{x}_j\right)$$

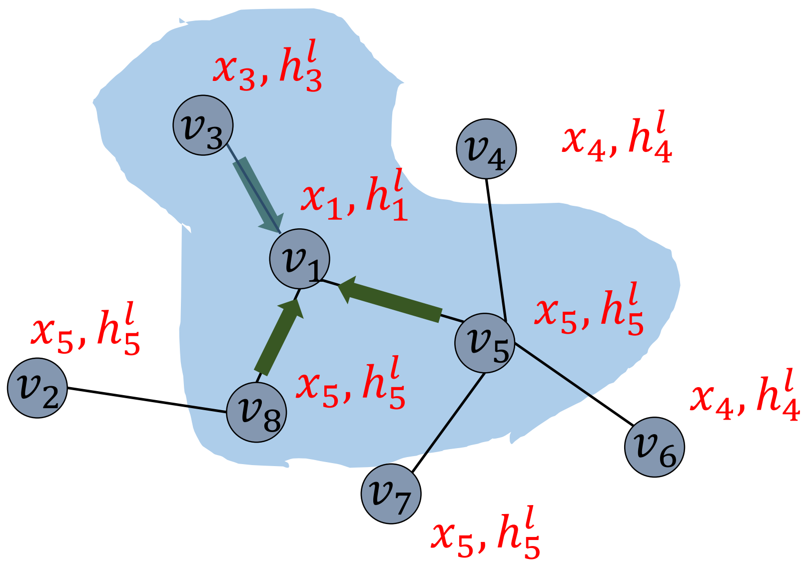


GCN: Types of Operations for Graph Filtering



GCN: Graph Filtering

Spatial filtering



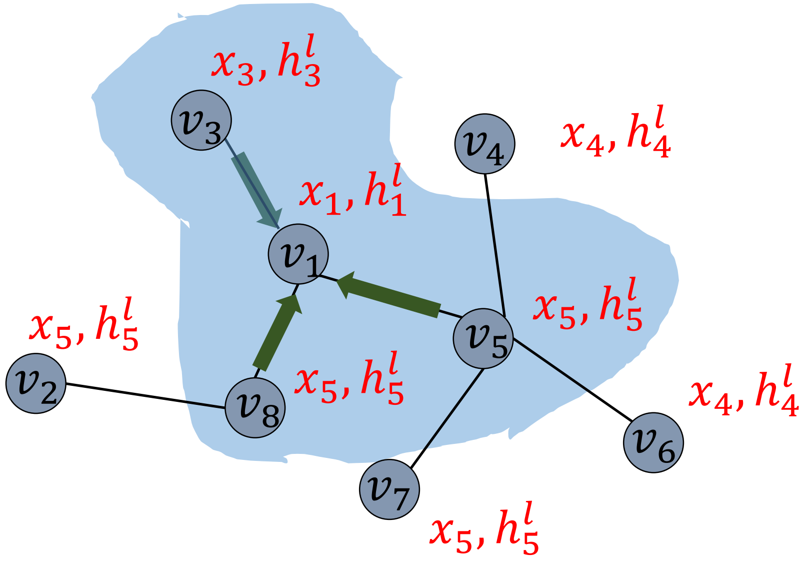
$\boldsymbol{h}_i$: hidden features

$\boldsymbol{x}_i$: input features

Original GNN: Graph neural networks for ranking web pages, IEEE *Web Intelligence* (Scarselli et al. 2005)

GCN: Graph Filtering

Spatial filtering



$\boldsymbol{h}_i$: hidden features

$\boldsymbol{x}_i$: input features

$$\boldsymbol{h}_i^{(l+1)} = \sum_{v_j \in N(v_i)} \boldsymbol{f}\left(\boldsymbol{x}_i, \boldsymbol{h}_j^{(l)}, \boldsymbol{x}_j\right), \qquad \forall\, v_i \in V.$$

$N(v_i)$: neighbors of the node v_i.

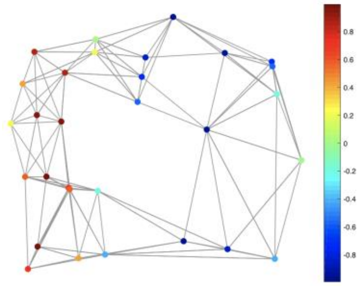
$\boldsymbol{f}(\cdot)$: feedforward neural network.

GNN

R: Graph Fourier Transform

A signal $\boldsymbol{f}$ can be written as graph Fourier series:

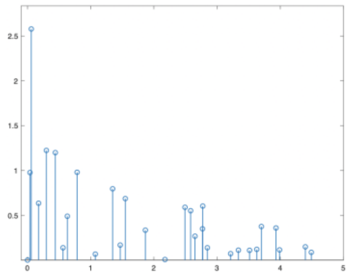
$$\boldsymbol{f} = \sum_{\boldsymbol{i}} \hat{f}_i \boldsymbol{u}_i \qquad \boldsymbol{f}^T \boldsymbol{u}_k = \sum_{\boldsymbol{i}} \hat{f}_i \boldsymbol{u}_i^T \boldsymbol{u}_k = \hat{f}_k \;= \boldsymbol{u}_k^T \boldsymbol{f} \qquad \hat{\boldsymbol{f}} = \begin{bmatrix} \hat{f}_{\boldsymbol{1}} \\ \dots \\ \hat{f}_{\boldsymbol{N}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_1^T \\ \dots \\ \boldsymbol{u}_N^T \end{bmatrix} \boldsymbol{f}$$



$\hat{\boldsymbol{f}} = \boldsymbol{U}^T \boldsymbol{f}$

Decompose signal f

Reconstruct signal $\boldsymbol{f}$



Spatial domain: $\boldsymbol{f}$

Spectral domain: $\hat{\boldsymbol{f}}$

Design GCN in spatial domain

Design GCN in spectral domain

The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine, 2013* (1867회 인용)

GCN: Graph Filtering

Spectral filtering

Recall: a signal $\boldsymbol{f}$ can be written as graph Fourier series:

GFT: $\hat{\boldsymbol{f}} = \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{f}, \hat{f}_i = \boldsymbol{u}_i^T\boldsymbol{f}$ **IGFT:** $\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}} = \sum_i \hat{f}_i\boldsymbol{u}_i$

Filter a graph signal $\boldsymbol{f}$:

$\boldsymbol{f}$

GCN: Graph Filtering

Spectral filtering

Recall: a signal $\boldsymbol{f}$ can be written as graph Fourier series:

GFT: $\hat{\boldsymbol{f}} = \boldsymbol{U}^T \boldsymbol{f}, \hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$

IGFT: $\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}} = \sum_i \hat{f}_i \boldsymbol{u}_i$

Filter a graph signal $\boldsymbol{f}$:

$$\boxed{\boldsymbol{f}} \xrightarrow[\text{Decompose}]{\textbf{GFT}} \boxed{\hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}}$$

Coefficients

GCN: Graph Filtering

Spectral filtering

Recall: a signal $\boldsymbol{f}$ can be written as graph Fourier series:

GFT: $\hat{\boldsymbol{f}} = \boldsymbol{U}^T \boldsymbol{f}, \hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$ **IGFT:** $\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}} = \sum_i \hat{f}_i \boldsymbol{u}_i$

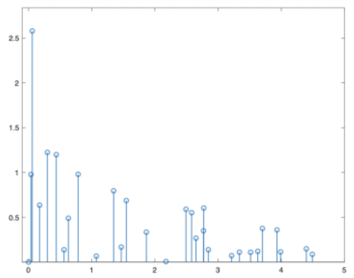
Filter a graph signal $\boldsymbol{f}$:

$\boldsymbol{f}$ $\xrightarrow[\text{Decompose}]{\textbf{GFT}}$ $\hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$ $\xrightarrow[\text{Filter}]{\hat{g}(\lambda_i)}$ $\hat{g}(\lambda_i) \cdot \boldsymbol{u}_i^T \boldsymbol{f}$

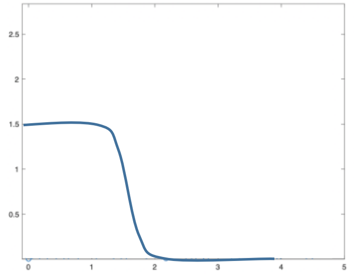
Coefficients

Filtered coefficients

$\hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$



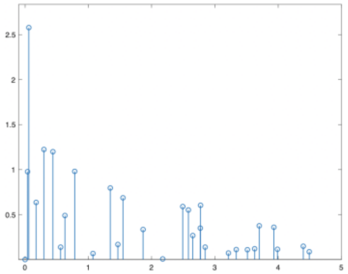
$\hat{g}(\lambda_i)$



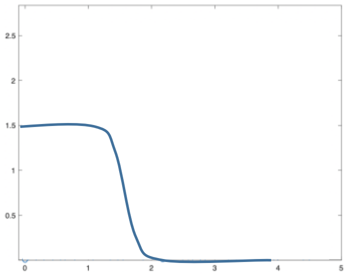
Filter $\hat{g}(\lambda_i)$: Modulating the frequency

GCN: Graph Filtering

$\hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$



$\hat{g}(\lambda_i)$



Spectral filtering

Recall: a signal $\boldsymbol{f}$ can be written as graph Fourier series:

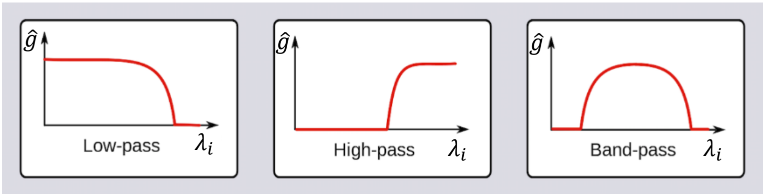
GFT: $\hat{\boldsymbol{f}} = \boldsymbol{U}^T \boldsymbol{f}, \hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$ **IGFT:** $\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}} = \sum_i \hat{f}_i \boldsymbol{u}_i$

Filter a graph signal $\boldsymbol{f}$:

$$\boldsymbol{f} \xrightarrow[\text{Decompose}]{\text{GFT}} \underset{\text{Coefficients}}{\hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}} \xrightarrow[\text{Filter}]{\hat{g}(\lambda_i)} \underset{\text{Filtered coefficients}}{\hat{g}(\lambda_i) \cdot \boldsymbol{u}_i^T \boldsymbol{f}} \xrightarrow[\text{Reconstruct}]{\text{IGFT}} \underset{\text{Filtered signal}}{\sum_i \hat{g}(\lambda_i) \boldsymbol{u}_i^T \boldsymbol{f}\, \boldsymbol{u}_i}$$

Filter $\hat{g}(\lambda_i)$: Modulating the frequency

Example:



GCN: Graph Filtering

Spectral filtering

$\boldsymbol{f}$ $\xrightarrow[\text{Decompose}]{\text{GFT}}$ $\hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$ $\xrightarrow[\text{Filter}]{\hat{g}(\lambda_i)}$ $\hat{g}(\lambda_i) \cdot \boldsymbol{u}_i^T \boldsymbol{f}$ $\xrightarrow[\text{Reconstruct}]{\text{IGFT}}$ $\sum_i \hat{g}(\lambda_i) \boldsymbol{u}_i^T \boldsymbol{f}\, \boldsymbol{u}_i$

Coefficients — Filtered coefficients — Filtered signal

$$\boldsymbol{U}\hat{g}(\Lambda)\boldsymbol{U}^T\boldsymbol{f} = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \cdots \quad \boldsymbol{u}_N] \begin{bmatrix} \hat{g}(\lambda_1) & 0 & 0 & 0 \\ 0 & \hat{g}(\lambda_2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \hat{g}(\lambda_N) \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^T \\ \boldsymbol{u}_2^T \\ \vdots \\ \boldsymbol{u}_N^T \end{bmatrix} \boldsymbol{f}$$

$$= [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \cdots \quad \boldsymbol{u}_N] \begin{bmatrix} \hat{g}(\lambda_1) & 0 & 0 & 0 \\ 0 & \hat{g}(\lambda_2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \hat{g}(\lambda_N) \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1^T\boldsymbol{f} \\ \boldsymbol{u}_2^T\boldsymbol{f} \\ \vdots \\ \boldsymbol{u}_N^T\boldsymbol{f} \end{bmatrix}$$

$$= [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \cdots \quad \boldsymbol{u}_N] \begin{bmatrix} \hat{g}(\lambda_1)\boldsymbol{u}_1^T\boldsymbol{f} \\ \hat{g}(\lambda_2)\boldsymbol{u}_2^T\boldsymbol{f} \\ \vdots \\ \hat{g}(\lambda_N)\boldsymbol{u}_N^T\boldsymbol{f} \end{bmatrix} = \sum_i \hat{g}(\lambda_i)\boldsymbol{u}_i^T\boldsymbol{f}\,\boldsymbol{u}_i$$

GCN: Graph Filtering

Spectral filtering

Recall: a signal $\boldsymbol{f}$ can be written as graph Fourier series:

GFT: $\hat{\boldsymbol{f}} = \boldsymbol{U}^T\boldsymbol{f}, \hat{f}_i = \boldsymbol{u}_i^T\boldsymbol{f}$ **IGFT:** $\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}} = \sum_i \hat{f}_i \boldsymbol{u}_i$

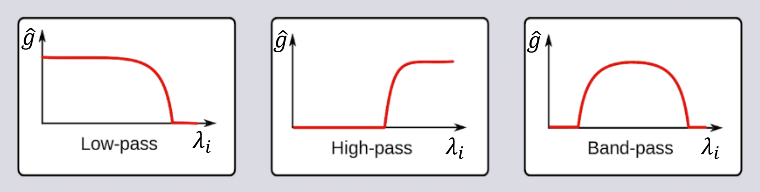
$$\boxed{\sum_i \hat{g}(\lambda_i)\boldsymbol{u}_i^T\boldsymbol{f}\, \boldsymbol{u}_i}$$

Filter a graph signal $\boldsymbol{f}$:

$\boldsymbol{f}$ —GFT (Decompose)→ $\hat{\boldsymbol{f}} = \boldsymbol{U}^T\boldsymbol{f}$ (Coefficients) —$\hat{g}(\Lambda)$ (Filter)→ $\hat{g}(\Lambda)\boldsymbol{U}^T\boldsymbol{f}$ (Filtered coefficients) —IGFT (Reconstruct)→ $\boldsymbol{U}\hat{g}(\Lambda)\boldsymbol{U}^T\boldsymbol{f}$ (Filtered signal)

$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

Example:



GCN: Graph Filtering

Spectral filtering

Recall: a signal $\boldsymbol{f}$ can be written as graph Fourier series:

GFT: $\hat{\boldsymbol{f}} = \boldsymbol{U}^T \boldsymbol{f}, \hat{f}_i = \boldsymbol{u}_i^T \boldsymbol{f}$ **IGFT:** $\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}} = \sum_i \hat{f}_i \boldsymbol{u}_i$

Filter a graph signal $\boldsymbol{f}$:

$$\boxed{\sum_i \hat{g}(\lambda_i)\boldsymbol{u}_i^T \boldsymbol{f}\, \boldsymbol{u}_i}$$

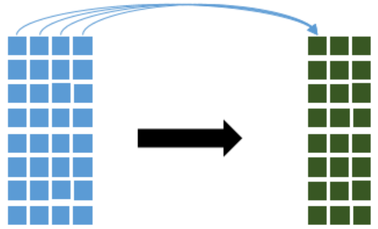
$$\boldsymbol{f} \xrightarrow[\text{Decompose}]{\text{GFT}} \underbrace{\hat{\boldsymbol{f}} = \boldsymbol{U}^T \boldsymbol{f}}_{\text{Coefficients}} \xrightarrow[\text{Filter}]{\hat{g}(\Lambda)} \underbrace{\hat{g}(\Lambda)\boldsymbol{U}^T \boldsymbol{f}}_{\text{Filtered coefficients}} \xrightarrow[\text{Reconstruct}]{\text{IGFT}} \underbrace{\boxed{\boldsymbol{U}\hat{g}(\Lambda)\boldsymbol{U}^T}}_{\boxed{\hat{g}(\boldsymbol{L})}}\boldsymbol{f}$$

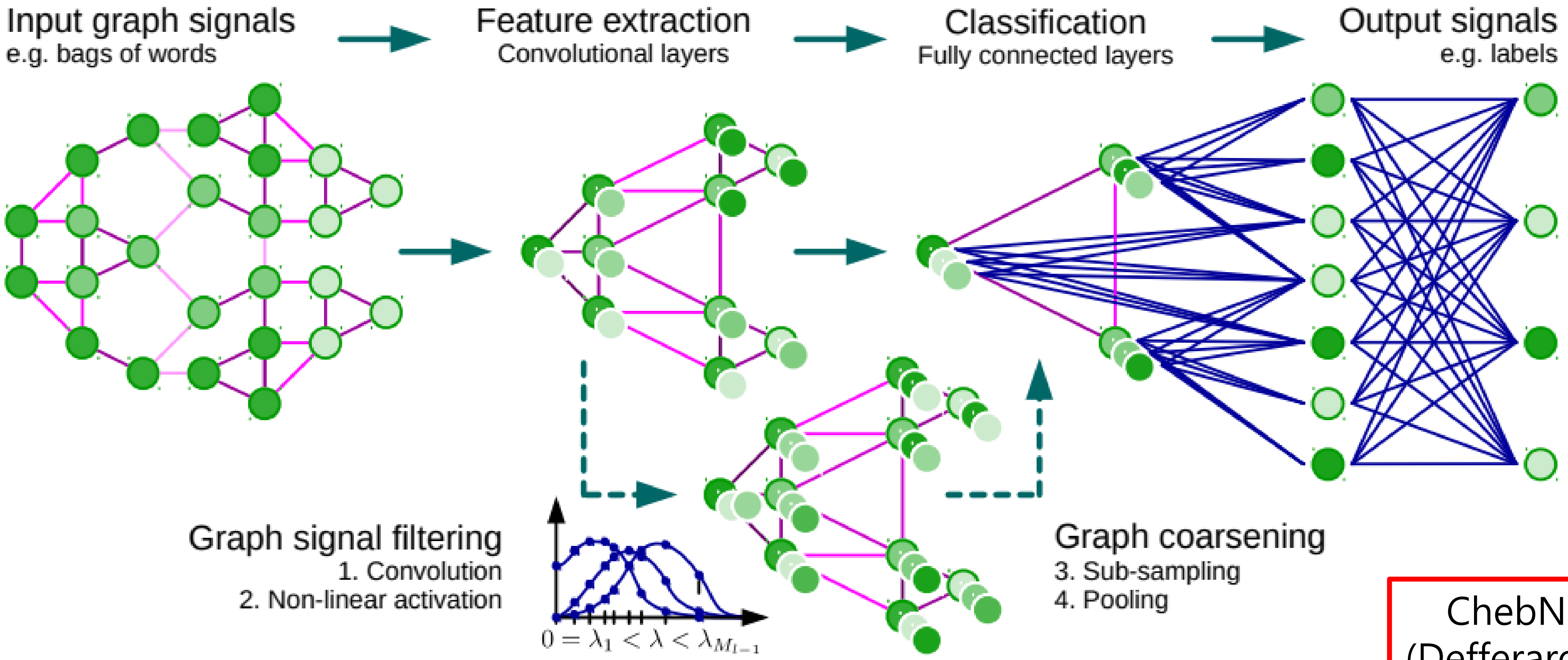
$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

$$\boldsymbol{f} \xrightarrow[\text{Filtering}]{} \hat{g}(\boldsymbol{L})\boldsymbol{f}$$

GCN: Graph Filtering

How to design the spectral filter for GCN?



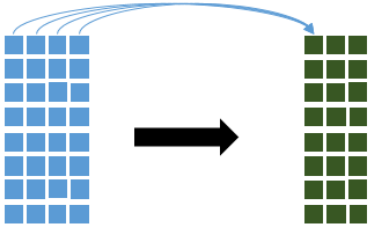


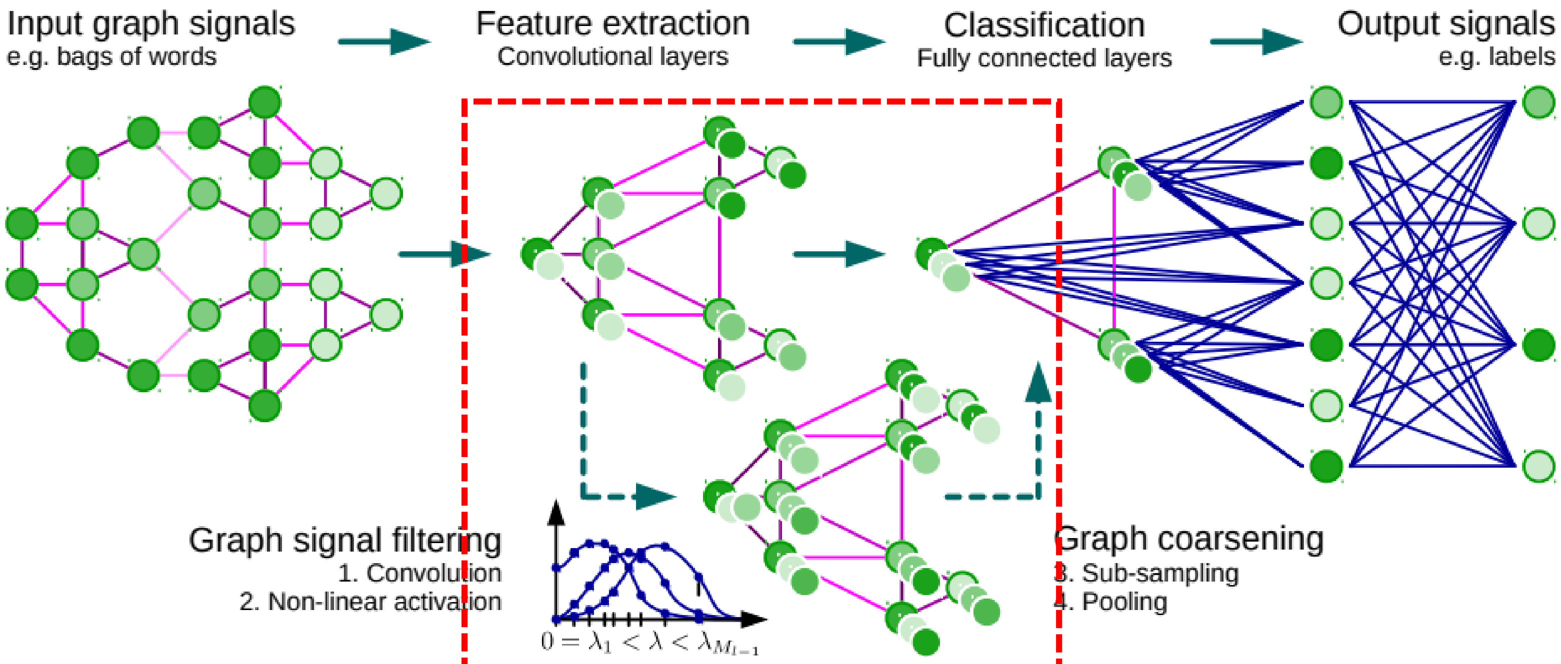
ChebNet
(Defferard et al.
NIPS 2016)

GCN: Graph Filtering

How to design the spectral filter for GCN?

Data-driven! Learn $\boxed{\hat{g}(\Lambda)}$ from data!



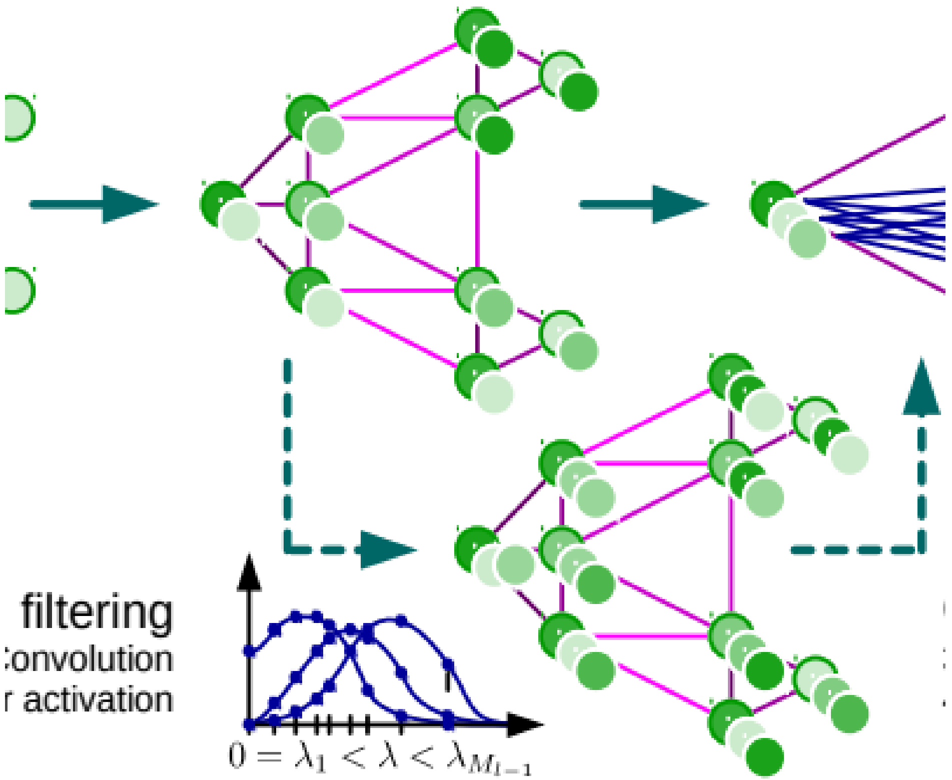


GCN: Graph Filtering

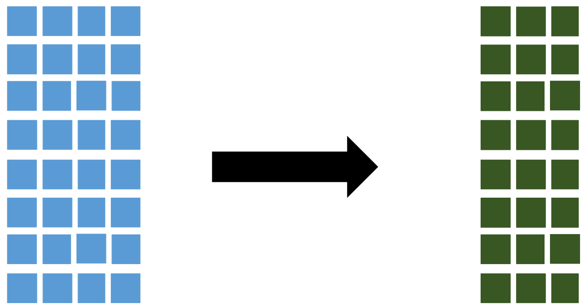
How to design the spectral filter for GCN?

Data-driven! Learn $\hat{g}(\Lambda)$ from data!

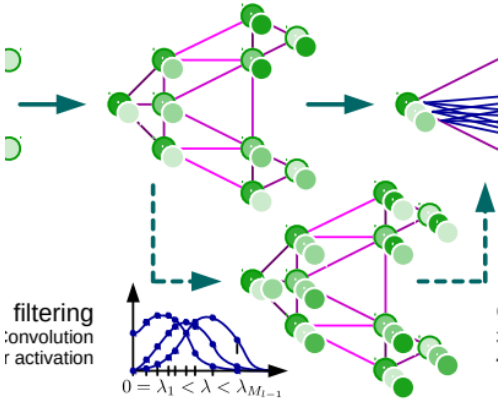
How to deal with multi-channel signals?



$$\boldsymbol{F}_{in} \in \mathbb{R}^{N \times d_1} \rightarrow \boldsymbol{F}_{out} \in \mathbb{R}^{N \times d_2}.$$



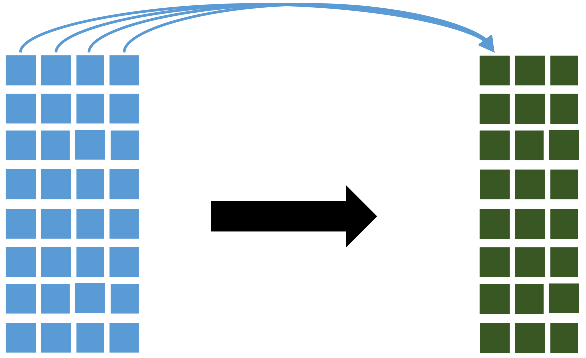
GCN: Graph Filtering



How to design the spectral filter for GCN?

Data-driven! Learn $\hat{g}(\Lambda)$ from data!

How to deal with multi-channel signals?



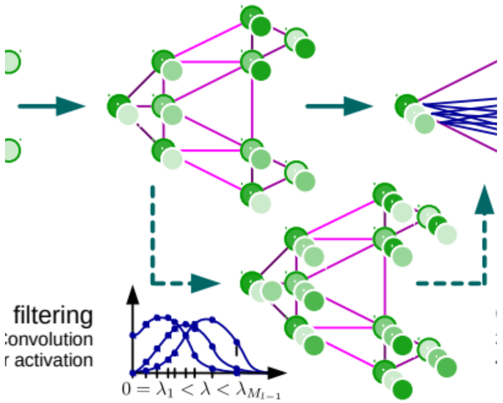
$$\boldsymbol{F}_I \in \mathbb{R}^{N\times d_1} \rightarrow \boldsymbol{F}_O \in \mathbb{R}^{N\times d_2}.$$

Each input channel contributes to each output channel

$$\boldsymbol{F}_O[:,n] = \sum_{m=1}^{d_1} \boxed{\hat{g}_{nm}(\boldsymbol{L})}\boldsymbol{F}_I[:,m] \quad n = 1, \dots d_2$$

Filter for input channel

GCN: Graph Filtering

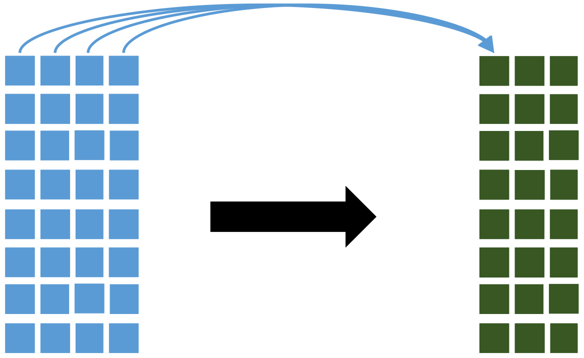


How to design the spectral filter for GCN?

Data-driven! Learn $\hat{g}(\Lambda)$ from data!

How to deal with multi-channel signals?

$$\boldsymbol{F}_I \in \mathbb{R}^{N\times d_1} \rightarrow \boldsymbol{F}_O \in \mathbb{R}^{N\times d_2}.$$



Each input channel contributes to each output channel

$$\boldsymbol{F}_O[:,n] = \sum_{m=1}^{d_1} \boxed{\hat{g}_{nm}(\boldsymbol{L})}\boldsymbol{F}_I[:,m] \quad n = 1,\ldots d_2$$

Filter for input channel

Learn $d_2 \times d_1$ filter

Summary Questions of the lecture

- Explain the spectral filtering of a graph signal
- How to design the spectral filter for GCN?
- What is the channel in a graph signal?
- How to deal with multi-channel signals in GCN?