L6-1: Graph convolutional network

We have introduced neural network basics, CNN, graph NN, generative networks (VAE, GAN, cGAN). Here we give more details of

1. Graph NN and kernelized affinity on graph (GCN, GAT)
2. The GAN loss and divergence between distributions (GAN, f-GAN)

Spectral graph convolution

- GCN, ChebNet
- Graph attention network (GAT)

- Graph convolution (spectral construction)

idea: convolution and Fourier transform

- Discrete Fourier transform (1D)

$x \in \mathbb{R}^n$, $\hat{x} = F^* x$,

F : $n \times n$ Fourier matrix

$F_{jk} = e^{2\pi i \frac{jk}{n}}$, $0 \le j, k \le n-1$

$\hat{x}(k) = \sum_{j=0}^{n-1} e^{-2\pi i \frac{jk}{n}} x(j)$, $k = 0, \cdots, n-1$

- 1D convolution (with periodic boundary padding)

$a \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, index $0, \cdots, n-1$,

$(a * x)_m = \sum_{j=0}^{n-1} a_{(m-j) \bmod n} \, x_j$

<u>Lemma</u> $a * x = Ax$, where A is a circulant

matrix with a being the first column,

$$A_{mj} = a(m-j \bmod n), \quad m,j = 0,\cdots,n-1$$

$$A = \begin{bmatrix} a_0 & a_{n-1} & & a_1 \\ a_1 & a_0 & & \vdots \\ \vdots & a_1 & \ddots & a_{n-1} \\ a_{n-1} & \vdots & & a_0 \end{bmatrix}_{n\times n} := \mathrm{Circ}(a)$$

Prop For $a \in \mathbb{R}^n$, $A = \mathrm{Circ}(a)$, then A is diagonalized under Fourier matrix, that is

$$A = U D_{\hat{a}} U^*, \quad U = \frac{1}{\sqrt{n}} F,$$

$D_{\hat{a}}$ is the diagonal matrix $(D_{\hat{a}})_{kk} = \hat{a}_k$, and U is unitary matrix.

Pf. First verify that U is unitary, $U^* U = I_n$. Then verify that

$$A U = U D_{\hat{a}},$$

which is equivalently

$$A F_{\cdot,k} = \hat{a}_k F_{\cdot,k}, \quad k = 0,\cdots,n-1. \quad \#$$

Rk The Prop gives that

$$a * x = A x = \frac{1}{n} F D_{\hat{a}} F^* x$$

$$\Rightarrow \quad (a*x)^\wedge = F^*(a*x) = D_{\hat{a}}(F^* x) = \hat{a} \odot \hat{x},$$

$$(u \odot v)_i = u_i v_i, \text{ Hadamard Product}$$

$$\boxed{(a * x)^\wedge = \hat{a} \odot \hat{x}}$$

- graph convolution

$G = (V, E)$, A adjacency matrix

W weighted affinity

$D_{ii} = \sum_j W_{ij}$, $D_{ii} > 0$, $i = 1, \cdots, n$

$$L \begin{cases} L_{un} = D - W \\ L_{rw} = I - D^{-1} W \\ L_{sym} = I - D^{-1/2} W D^{-1/2} \end{cases}$$

$$L_{sym} = \Phi \Lambda \Phi^T, \quad \Phi^T \Phi = I_n,$$

$x : V \to \mathbb{R}$, $x \in \mathbb{R}^n$, graph signal

Define general Fourier transform $\hat{x} := \Phi^T x$

For $g : V \to \mathbb{R}$, g possibly locally supported. we want to have

$$(g * x)^\wedge = \hat{g} \odot \hat{x}.$$

$$\Phi^T (g * x) = \hat{g} \odot (\Phi^T x)$$

$$\Rightarrow \quad g * x := \Phi D_{\hat{g}} \Phi^T x$$

$\hat{g} \in \mathbb{R}^n$, and if we furtherly require

$$\hat{g}_i = f(\lambda_i), \quad i = 1, \cdots, n, \quad g : \mathbb{R} \to \mathbb{R}$$

then $\Phi D_{\hat{g}} \Phi^T = \Phi f(\Lambda) \Phi^T = f(L)$, we then have

$$g * x = f(L)\, x$$

In GCN, $f(\cdot)$ is a linear function

- ChebNet and GCN

input $x(u, c)$, $u \in V$, $c \in [C_{in}]$

output $y(u, c')$, $u \in V$, $c' \in [C_{out}]$

$$\begin{cases} y(u, c') = \sigma\big(y_{pre}(u, c') + b(c') \big), & c' \in [C_{out}] \\ y_{pre}(\cdot, c') = \sum_{c=1}^{C_{in}} \sum_{l=0}^{L} \theta_l^{(c', c)} T_l(\tilde{L})\, x(\cdot, c) & \end{cases}$$

σ: activation function, ReLU, sigmoid, etc

b: bias

$$\tilde{L} = \alpha_1 I + \alpha_2 L_{sym}$$

the rescaled, recentered graph Laplacian

$T_l(x)$ Chebyshev polynomials, $l = 0, 1, \cdots$

GCN: $L = 1$, tie θ_0 and θ_1:

$$y_{pre}(\cdot, c') = \sum_{c \in [C_{in}]} \theta^{(c', c)} \left(\alpha_1' I + \alpha_2' A_{sym} \right) x(\cdot, c)$$

$$A_{sym} = D^{-1/2} A D^{-1/2}$$

α_1', α_2' : fixed constants

Matrix form

$$\underset{n \times C_{out}}{\boxed{Y_{pre}}} = \sum_{l=0}^{L} \underset{n \times n}{\boxed{T_l(\tilde{L})}} \; \underset{n \times C_{in}}{\boxed{X}} \; \underset{C_{in} \times C_{out}}{\boxed{\Theta_l}}$$

$$Y_{pre} = \sum_{l=0}^{L} T_l(\tilde{L}) \, X \, \Theta_l$$

- GAT (graph attention network)

$$Y_{pre} = \sum_{r=1}^{R} \underset{\substack{\uparrow \\ \text{attention graph affinity}}}{A^{(r)}} X \, \Theta_r ,$$

$$A^{(r)}_{uv} = \frac{\exp(c^{(r)}_{uv})}{\sum_{v', v' \sim u} \exp(c^{(r)}_{uv'})}$$ for $v \sim u$, v is 1-nb of u.

$$c^{(r)}_{uv} = \sigma\left(\left\langle \underline{a}^{(r)}, \begin{bmatrix} \underline{W}^{(r)} x_u \\ W^{(r)} x_v \end{bmatrix} \right\rangle \right)$$

$\{a^{(r)}, W^{(r)}\}_{r=1}^{R}$ trainable para

R: number of attention heads

In original GAT, $\Theta_r = W^{(r)} C^{(r)}$

$C^{(r)}$: fixed matrix to concatenate the output for $r = 1, \cdots, R$ into Y_{pre}

Generalized such that θ_r has extra trainable parameters, removing weight tie with $W^{(r)}$.