Лекция 5. Графови невронни мрежи.

DSCM023 „Прогнозиране чрез анализ на данни - III“

ИНСТИТУТ *за* СЪВРЕМЕННИ ФИЗИЧЕСКИ ИЗСЛЕДВАНИЯ

НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ

Стоян Мишев, Петър Христов

https://facebook.com/nbudatascience

План

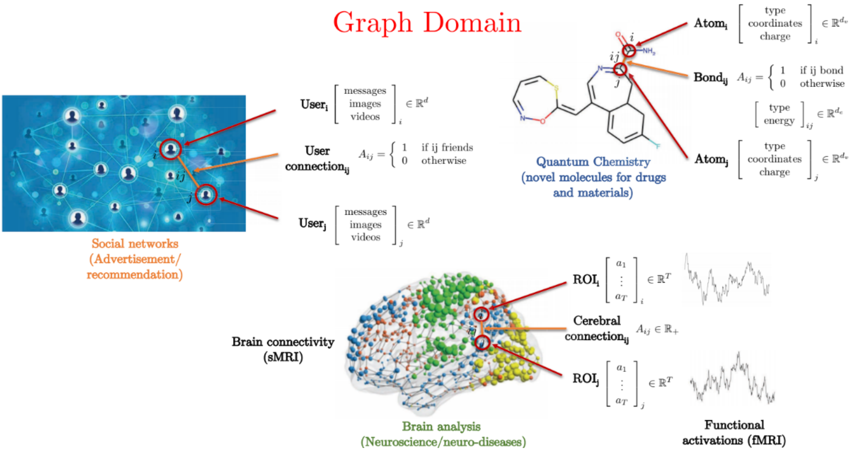
Графи и въпроси върху операцията конволюция

Спекрална теория на графи

Конволюция на графи in inverse space

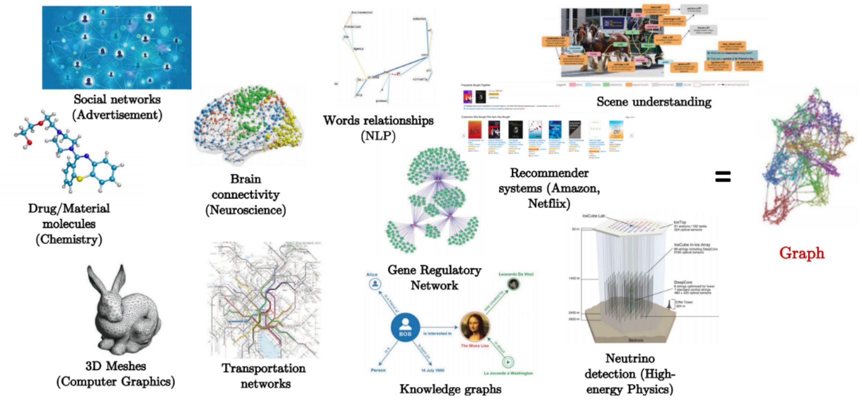
Графи и въпроси върху операцията конволюция

https://youtu.be/Iiv9R6BjxHM



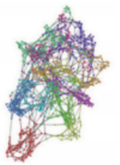
Graph Domain
User$_i$ $\begin{bmatrix} \text{messages} \\ \text{images} \\ \text{videos} \end{bmatrix}_i \in \mathbb{R}^d$
User connection$_{ij}$ $A_{ij} = \begin{cases} 1 & \text{if ij friends} \\ 0 & \text{otherwise} \end{cases}$
User$_j$ $\begin{bmatrix} \text{messages} \\ \text{images} \\ \text{videos} \end{bmatrix}_j \in \mathbb{R}^d$
i
ij
j
Social networks (Advertisement/ recommendation)
Atom$_i$ $\begin{bmatrix} \text{type} \\ \text{coordinates} \\ \text{charge} \end{bmatrix}_i \in \mathbb{R}^{d_v}$
Bond$_{ij}$ $A_{ij} = \begin{cases} 1 & \text{if ij bond} \\ 0 & \text{otherwise} \end{cases}$
$\begin{bmatrix} \text{type} \\ \text{energy} \end{bmatrix}_{ij} \in \mathbb{R}^{d_e}$
Atom$_j$ $\begin{bmatrix} \text{type} \\ \text{coordinates} \\ \text{charge} \end{bmatrix}_j \in \mathbb{R}^{d_v}$
Quantum Chemistry (novel molecules for drugs and materials)
ROI$_i$ $\begin{bmatrix} a_1 \\ \vdots \\ a_T \end{bmatrix}_i \in \mathbb{R}^T$
Cerebral connection$_{ij}$ $A_{ij} \in \mathbb{R}_+$
ROI$_j$ $\begin{bmatrix} a_1 \\ \vdots \\ a_T \end{bmatrix}_j \in \mathbb{R}^T$
Brain connectivity (sMRI)
Brain analysis (Neuroscience/neuro-diseases)
Functional activations (fMRI)

Graph Domain

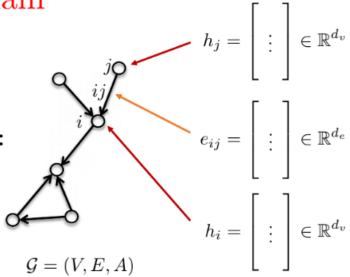


Graph Domain

- Graphs G are defined by :
  - Vertices V
  - Edges E
  - Adjacency matrix A
- Graph features :
  - Node features : $h_i,\ h_j$ (atom type)
  - Edge features : e_{ij} (bond type)
  - Graph features : g (molecule energy)



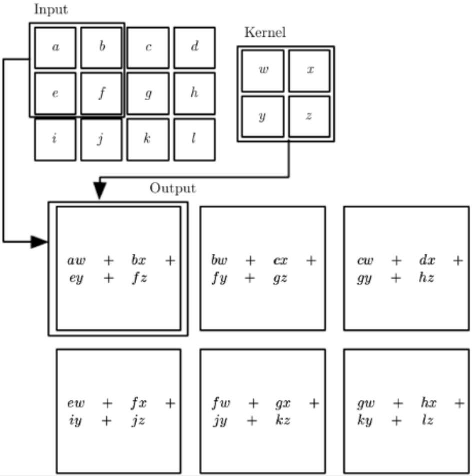
=



$$\underbrace{\begin{gathered}\mathcal{G} = (V, E, A) \\ V = \{1, .., n\}\end{gathered}}$$

$$g = \begin{bmatrix} \vdots \end{bmatrix} \in \mathbb{R}^{d_g}$$

Операцията „конволюция“



- Convolutional layer (for grids) :

$$\underset{n_1 \times n_2 \times d}{h^{\ell+1}} = \underset{3 \times 3 \times d}{w^{\ell}} * \underset{n_1 \times n_2 \times d}{h^{\ell}}$$

$*$

$=$

h^{ℓ}

Image/Hidden features

w^{ℓ}

Pattern/kernel (learned by backpropagation)

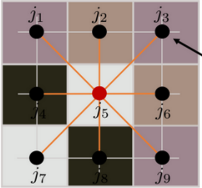
$h^{\ell+1}$

Convolution

- How to define convolution ?
  - Definition #1 : Convolution as template matching
  - $O(n)$ by parallelization and for compact support patterns

$$3 \times 3 \times d \qquad n_1 \times n_2 \times d$$

$$
\begin{aligned}
h_i^{\ell+1} &= w^\ell * h_i^\ell \\
&= \sum_{\underbrace{j \in \Omega}_{j \in \mathcal{N}_i}} \langle w_j^\ell, \underbrace{h_{i-j}^\ell}_{h_{i+j}^\ell = h_{ij}^\ell} \rangle \\
&= \sum_{j \in \mathcal{N}_i} \langle w_j^\ell, h_{ij}^\ell \rangle \\
&= \sum_{j \in \mathcal{N}_i} \langle \begin{bmatrix} \\ w_j^\ell \\ \\ \end{bmatrix}, \begin{bmatrix} \\ h_{ij}^\ell \\ \\ \end{bmatrix} \rangle
\end{aligned}
$$



Node j_3 is always located at the top-right corner of the pattern.

$$\begin{bmatrix} \\ w_{j_3}^\ell \\ \\ \end{bmatrix} \in \mathbb{R}^d$$

Template features at j_3

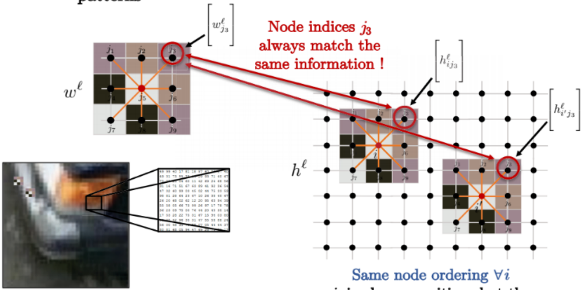
All nodes of the template w^l are always ordered/positioned the same way !

ℓ

Convolution

- How to define convolution ?
  - Definition #1 : Convolution as template matching
  - $O(n)$ by parallelization for compact support patterns

$$
\begin{aligned}
h_i^{\ell+1} &= w^\ell * h_i^\ell \\
&= \sum_{j \in \mathcal{N}_i} \langle w_j^\ell, h_{ij}^\ell \rangle \\
&= \sum_{j \in \mathcal{N}_i} \left\langle \begin{bmatrix} w_j^\ell \end{bmatrix}, \begin{bmatrix} h_{ij}^\ell \end{bmatrix} \right\rangle
\end{aligned}
$$



Same node ordering $\forall i$
j_3 is always positioned at the top-right corner of the template.

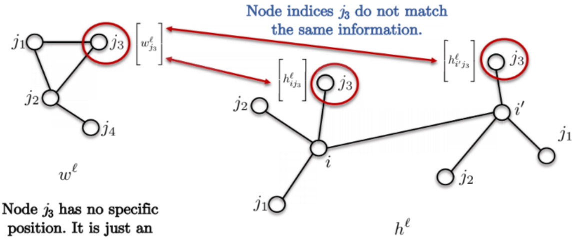
$$\left\langle \begin{bmatrix} w_{j_3}^\ell \end{bmatrix}, \begin{bmatrix} h_{ij_3}^\ell \end{bmatrix} \right\rangle$$

$$\left\langle \begin{bmatrix} w_{j_3}^\ell \end{bmatrix}, \begin{bmatrix} h_{i'j_3}^\ell \end{bmatrix} \right\rangle$$

These matching scores are always for the top-right corner between the template and the image patches.

Graph Convolution

- Can we extend template matching for graphs ?
  - Main issues :
    - No node ordering : How to match template features with data features when nodes have no given position (index is not a position) ?



Node j_3 has no specific position. It is just an (arbitrary) index.

No node ordering on graphs :
The correspondence of nodes is lost on graphs.
There is no up, down, right and left on graphs.

$$\langle \left[w^\ell_{j_3} \right], \left[h^\ell_{ij_3} \right] \rangle$$

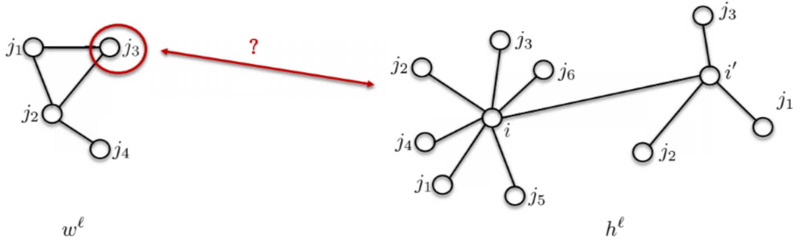
$$\langle \left[w^\ell_{j_3} \right], \left[h^\ell_{i'j_3} \right] \rangle$$

These matching scores have no meaning as they do not compare the same information.

$$h_i^{\ell+1} = w^\ell *_{\mathcal{G}} h_i^\ell$$
$n \times d$
$$= \cancel{\sum_{j \in \mathcal{N}_i} \langle w_j^\ell, h_{ij}^\ell \rangle}$$

Graph Convolution

- Can we extend template matching for graphs ?
  - Main issues :
    - No node ordering : How to match template features with data features ?
    - Heterogeneous neighborhood : How to deal with different neighborhood sizes ?



Graph Convolution

- How to define convolution ?
  - Definition #1 : Template matching
  - Definition #2 : Convolution theorem
    - Fourier transform of the convolution of two functions is the pointwise product of their Fourier transforms

$$\mathcal{F}(w * h) = \mathcal{F}(w) \odot \mathcal{F}(h) \quad \Rightarrow \quad w * h = \mathcal{F}^{-1}(\mathcal{F}(w) \odot \mathcal{F}(h))$$

    - Generic Fourier transform has $\mathrm{O}(n^2)$ complexity, but if the domain is a grid then complexity can be reduced to $\mathrm{O}(n\log n)$ with FFT[1].

    - Can we extend the Convolution theorem to graphs ?
      - How to define Fourier transform for graphs ?
      - How to compute fast spectral convolutions in $\mathrm{O}(n)$ time for compact kernels ?

$$w *_{\mathcal{G}} h \stackrel{?}{=} \mathcal{F}_{\mathcal{G}}^{-1}(\mathcal{F}_{\mathcal{G}}(w) \odot \mathcal{F}_{\mathcal{G}}(h))$$

[1] JW Cooley, JW Tukey, An algorithm for the machine calculation of complex Fourier series, 1965

Трансформация на Фурие (напомняне).

Права и обратна трансформация:

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt, \qquad h(t) = \int_{-\infty}^{\infty} H(f) e^{-2\pi i f t} df$$

If ...	then ...
$h(t)$ is real	$H(-f) = [H(f)]^*$
$h(t)$ is imaginary	$H(-f) = -[H(f)]^*$
$h(t)$ is even	$H(-f) = H(f)$ [i.e., $H(f)$ is even]
$h(t)$ is odd	$H(-f) = -H(f)$ [i.e., $H(f)$ is odd]
$h(t)$ is real and even	$H(f)$ is real and even
$h(t)$ is real and odd	$H(f)$ is imaginary and odd
$h(t)$ is imaginary and even	$H(f)$ is imaginary and even
$h(t)$ is imaginary and odd	$H(f)$ is real and odd

$$F(g * f) = F(g).F(f), \qquad g * f = F^{-1}(F(g).F(f))$$

Права и обратна трансформация:

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt, \qquad h(t) = \int_{-\infty}^{\infty} H(f) e^{-2\pi i f t} df$$

If...	then...
$h(t)$ is real	$H(-f) = [H(f)]^*$
$h(t)$ is imaginary	$H(-f) = -[H(f)]^*$
$h(t)$ is even	$H(-f) = H(f)$ [i.e., $H(f)$ is even]
$h(t)$ is odd	$H(-f) = -H(f)$ [i.e., $H(f)$ is odd]
$h(t)$ is real and even	$H(f)$ is real and even
$h(t)$ is real and odd	$H(f)$ is imaginary and odd
$h(t)$ is imaginary and even	$H(f)$ is imaginary and even
$h(t)$ is imaginary and odd	$H(f)$ is real and odd

$$F(g * f) = F(g).F(f), \qquad g * f = F^{-1}(F(g).F(f))$$

За дискретни данни h_k:

$$H_n = \sum_{k=0}^{N-1} h_k e^{2\pi i k n/N}, h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n e^{-2\pi i k n/N}, \sum_{k=0}^{N-1} |h_k|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |H_n|^2$$

Колко опрации са нужни за дискретното преобразуване на Фурие $H_n = \sum_{k=0}^{N-1} W^{nk} h_k$, където $W \equiv e^{2\pi i/N}$? Може би $\sim N^2$...

Колко опрации са нужни за дискретното преобразуване на Фурие $H_n = \sum_{k=0}^{N-1} W^{nk} h_k$, където $W \equiv e^{2\pi i/N}$? Може би $\sim N^2$... Оказва се, че това може да стане с $\sim N \log_2 N$ операции. Danielson и Lanczos (1942) показват, че дискретното преобразование на Фурие с дължина N може да се запише като **сума от две** дискретни преобразования на Фурие, всяко с дължина $N/2$. Едното е преобразуванието на стойностите, който стоят на четни позиции, а другото - на нечетните:

$$F_k = \sum_{j=0}^{N-1} W^{ij} f_j = ... = F_k^e + W^k F_k^o$$

Спекрална теория на графи

ENCYCLOPEDIA OF MATHEMATICS AND ITS APPLICATIONS

Eigenspaces of graphs

D. Cvetković
University of Belgrade

P. Rowlinson
University of Stirling

S. Simić
University of Belgrade

CAMBRIDGE
UNIVERSITY PRESS

Conference Board of the Mathematical Sciences

CBMS

Regional Conference Series in Mathematics

Number 92

Spectral Graph Theory

Fan R. K. Chung

Published for the
Conference Board of the Mathematical Sciences
by the
American Mathematical Society
Providence, Rhode Island
with support from the
National Science Foundation

Graph Laplacian

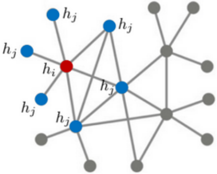
- Core operator in Spectral Graph Theory

$$\mathcal{G} = (V, E, \underset{n\times n}{A}) \quad \rightarrow \quad \underset{n\times n}{\Delta} = \mathrm{I} - D^{-1/2} A D^{-1/2} \qquad \text{Normalized Laplacian}$$

$$\text{where } \underset{n\times n}{D} = \mathrm{diag}(\sum_{j\neq i} A_{ij})$$

- Interpretation :
  - Measure of smoothness : Difference between local value h_i and its neighborhood average values h_j.

$$(\Delta h)_i = h_i - \sum_{j\in\mathcal{N}_i} \frac{1}{\sqrt{d_i d_j}} A_{ij} h_j$$



-

$$\Delta \equiv \nabla^2 = \sum_i \partial_{x_i}^2$$

-

$$\Delta \equiv \nabla^2 = \sum_i \partial_{x_i}^2$$

- Фурие трансформация на граф се нарича разлагането на матрицата на Лаплас за този граф по собствени вектори (създаващи, т.нар. Фурие базис на граф).

Fourier Functions

- Eigen-decomposition of graph Laplacian :

Lap Eigenvalues/
Spectrum

Lap Eigenvectors/
Fourier functions

$$\underset{n \times n}{\Delta} = \Phi^T \Lambda \Phi$$

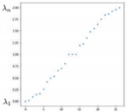
Fourier functions form an orthonormal basis

$$\text{where } \underset{n \times n}{\Phi} = [\phi_1, ..., \phi_n] = \begin{bmatrix} | & & | \\ \phi_1 & \ldots & \phi_n \\ | & & | \end{bmatrix} \text{ and } \Phi^T\Phi = \mathrm{I},\ \langle \phi_k, \phi_{k'} \rangle = \delta_{kk'}$$

$$\text{where } \underset{n \times n}{\Lambda} = \mathrm{diag}(\lambda_1, ..., \lambda_n) = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} \text{ and } 0 \leq \lambda_1 \leq ... \leq \lambda_n = \lambda_{\max} \leq 2$$

$$\text{and } \underset{n \times 1}{\Delta \phi_k} = \lambda_k \phi_k, \quad k = 1, ..., n$$

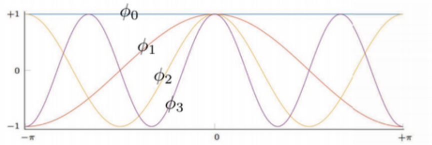
$n = 27$



Spectrum

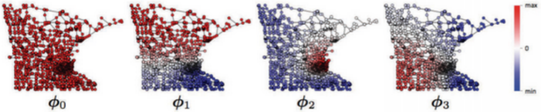
Fourier Functions

- Grid/Euclidean domain :



First eigenvectors of 1D Euclidean Laplacian = standard Fourier basis

- Graph domain :



First Laplacian eigenvectors of a graph

Fourier functions related to graph geometry

(s.a. communities, hubs, etc)

Spectral graph clustering[1]

[1] Von Luxburg, A tutorial on spectral clustering, 2007

Fourier Transform

- Fourier series : Decompose function h with Fourier functions[1] :

$$\underset{n\times 1}{h} = \sum_{k=1}^{n} \underbrace{\langle \phi_k, h \rangle}_{\substack{\mathcal{F}(h)_k = \hat{h}_k = \phi_k^T h \\ \text{scalar}}} \underset{n\times 1}{\phi_k}$$

$$= \sum_{k=1}^{n} \hat{h}_k \phi_k$$

$$= \underbrace{\Phi \hat{h}}_{\mathcal{F}^{-1}(\hat{h})}$$

Fourier transforms are one line of code (linear operations)

$$\underset{n\times 1}{\mathcal{F}(h)} = \Phi^T h = \hat{h}$$

Fourier Transform/ coefficients of Fourier Series

$$\underset{n\times 1}{\mathcal{F}^{-1}(\hat{h})} = \Phi \hat{h}$$

Inverse Fourier Transform

$$= \Phi \Phi^T h = h \text{ as } \mathcal{F}^{-1} \circ \mathcal{F} = \Phi \Phi^T = \mathrm{I}$$

Orthonormal basis/ Invertible transformation

[1] K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, 2011

Конволюция на графи in inverse space

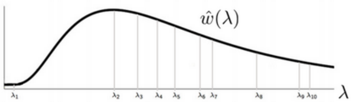
Convolution Theorem

- **Fourier transform of the convolution of two functions is the pointwise product of their Fourier transforms :**

$$
\begin{aligned}
\underset{n\times 1}{w * h} &= \underbrace{\mathcal{F}^{-1}}_{\Phi}\left(\underbrace{\mathcal{F}(w)}_{\Phi^T w = \hat{w}} \odot \underbrace{\mathcal{F}(h)}_{\Phi^T h} \right) \\
&= \underset{n\times n}{\Phi} \left(\underset{n\times 1}{\hat{w}} \odot \underset{n\times 1}{\Phi^T h} \right) \\
&= \Phi \left(\underset{n\times n}{\hat{w}(\Lambda)} \; \underset{n\times 1}{\Phi^T h} \right) \\
&= \Phi \; \hat{w}(\Lambda) \; \Phi^T h \\
&= \hat{w}(\underbrace{\Phi \Lambda \Phi^T}_{\Delta}) \; h \\
&= \underset{n\times n}{\hat{w}(\Delta)} \underset{n\times 1}{h}
\end{aligned}
$$

$$
\underset{n\times 1}{\hat{w}} = \begin{bmatrix} \hat{w}(\lambda_1) \\ \vdots \\ \hat{w}(\lambda_n) \end{bmatrix}
$$

$$
\underset{n\times n}{\hat{w}(\Lambda)} = \mathrm{diag}(\hat{w}) = \begin{bmatrix} \hat{w}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & \hat{w}(\lambda_n) \end{bmatrix}
$$



Expensive computation $O(n^2)$
No FFT

Spectral function/filter

Notebook

`https://github.com/xbresson/spectral_graph_convnets/blob/master/02_graph_convnet_lenet5_mnist_pytorch.ipynb`