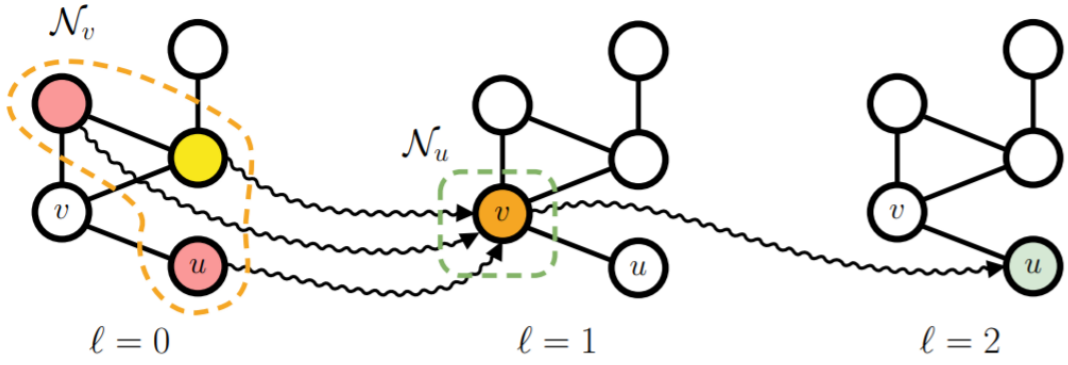
Графовые свёрточные нейронные сети

Александр Колодезный БПМИ192

Национальный исследовательский университет
«Высшая школа экономики» (Москва)

23 ноября 2021 г.

Graph Convolutional Network



Вид одного свёрточного слоя

$$\mathbf{h}_v^{\ell+1} = \phi^{\ell+1}\left(\mathbf{h}_v^{\ell},\ \Psi(\{\psi^{\ell+1}(\mathbf{h}_u^{\ell}) \mid u \in \mathcal{N}_v\})\right)$$

- Ψ — permutation-invariant функция
- ϕ^{l+1} и ψ^{l+1} — некоторый функция на *l*-ом слое

Graph Convolutional Network

$$\mathbf{h}_v^{\ell+1} = \sigma(\mathbf{W}^{\ell+1} \sum_{u \in \mathcal{N}(v)} \mathbf{L}_{uv} \mathbf{h}_u^{\ell}),$$

- σ — сигмоида
- W^{l+1} — обучаемая матрица
- L_{uv} — нормированный Лапласиан

Обработка рёбер

- Рёбра в графе могут иметь дополнительную информацию
- Можно брать различные веса для разных видов рёбер

$$\mathbf{h}_v^{\ell+1} = \phi^{\ell+1}\Big(\mathbf{h}_v^{\ell}, \sum_{c_k \in \mathcal{A}} \big(\Psi(\{\psi^{\ell+1}(\mathbf{h}_u^{\ell}) \mid u \in \mathcal{N}_v^{c_k}\}) * w_{c_k}\big)\Big),$$

- Более общий вид, если у рёбер есть свои признаки

$$\mathbf{h}_v^{\ell+1} = \phi^{\ell+1}\Big(\mathbf{h}_v^{\ell}, \ \Psi(\{e^{\ell+1}(\mathbf{a}_{uv})^T \psi^{\ell+1}(\mathbf{h}_u^{\ell}) \mid u \in \mathcal{N}_v\})\Big),$$

Attention

- Добавление дополнительных обучаемых весов на рёбра α_{uv}^{l+1}

$$\mathbf{h}_v^{\ell+1} = \phi^{\ell+1}\Big(\mathbf{h}_v^{\ell},\ \Psi(\{\alpha_{uv}^{\ell+1} * \psi^{\ell+1}(\mathbf{h}_u^{\ell}) \mid u \in \mathcal{N}_v\})\Big),$$

- Вычисляем сначала attention coefficient

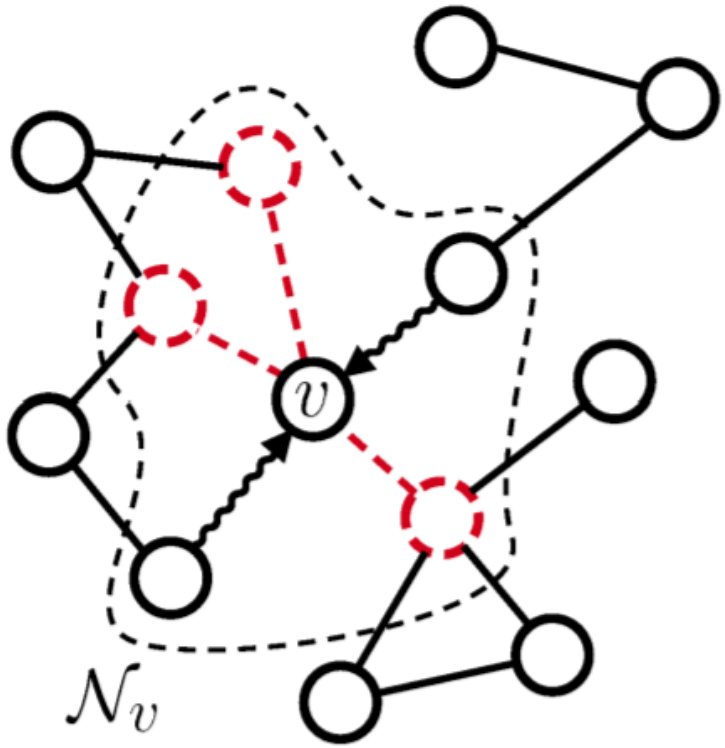
$$a(\mathbf{W}^{\ell}\,\mathbf{h}_u^{\ell}, \mathbf{W}^{\ell}\,\mathbf{h}_v^{\ell}) = \mathrm{LeakyReLU}((\mathbf{b}^{\ell})^T[\mathbf{W}^{\ell}\,\mathbf{h}_u^{\ell}, \mathbf{W}^{\ell}\,\mathbf{h}_v^{\ell}]),$$

- Считаем softmax

$$\alpha_{uv}^{\ell} = \frac{\exp(w_{uv}^{\ell})}{\sum_{u' \in \mathcal{N}_v} \exp(w_{u'v}^{\ell})}.$$

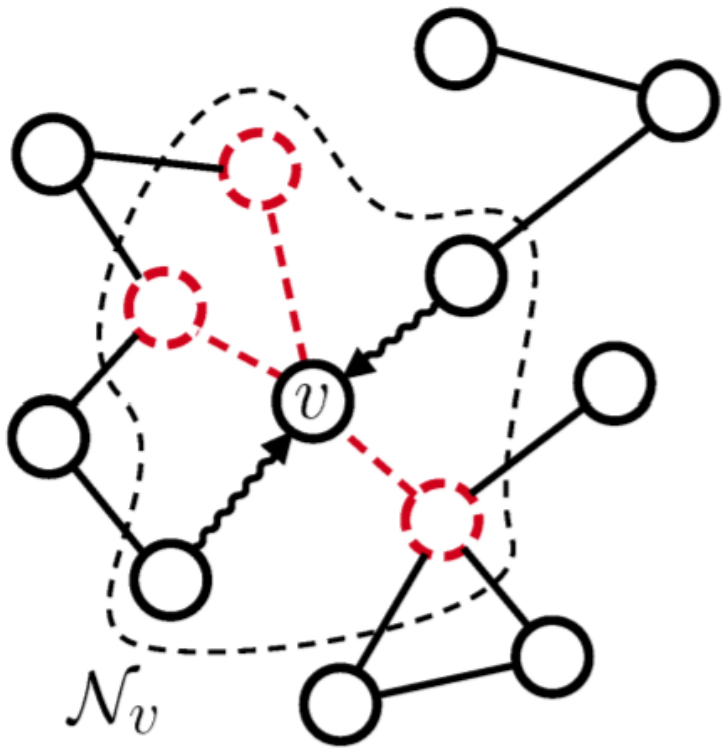
Sampling (GraphSAGE)

- Проблема — много вычислений на плотных графах
- Для каждой вершины на каждом слое выбираем случайное подмножество соседних вершин.
- Пересчитываем h_v только от h_u, которые выбрали.
- Приходится пересчитывать градиенты для всех вершин.

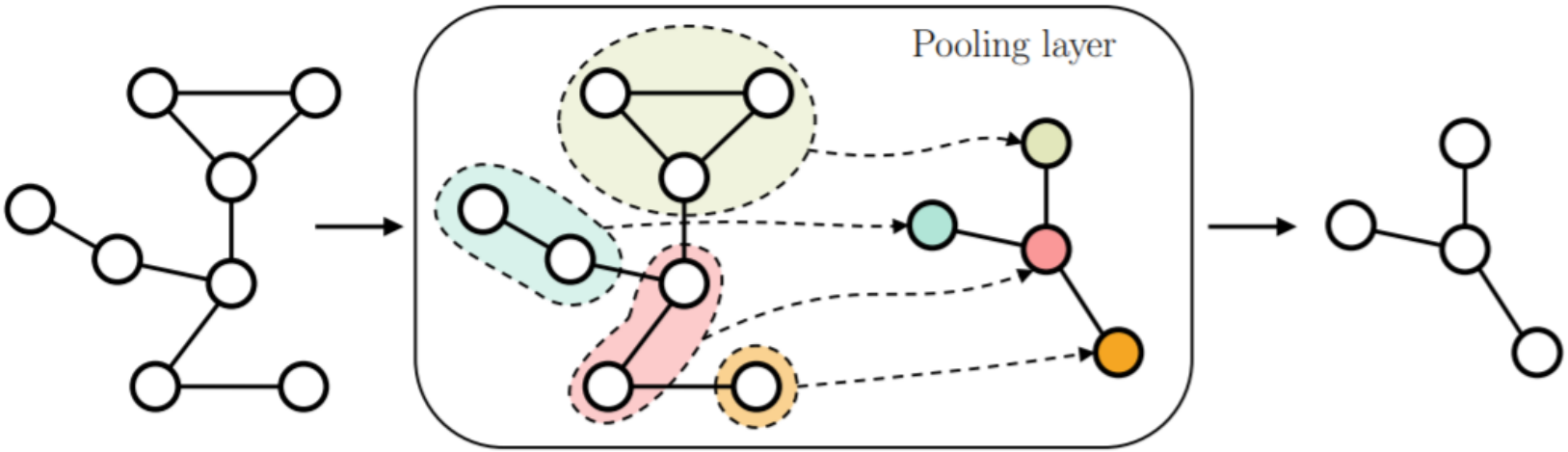


Sampling (FastGCN)

- Выбираем случайное подмножество вершин графа
- Оставляем в графе только выбранные вершины и рёбра между ними
- Рассматриваем при обучении модель только выбранные вершины.



Pooling



- Сжатие кластеров похожих вершин в одну
- Уменьшает размеры графа
- Уменьшает вычислительную стоимость

Differentiable Pooling

- Обучаем для каждой вершины попадание в кластера

$$\mathbf{S}^{\ell+1} = \mathrm{softmax}(\mathrm{GNN}(\mathbf{A}^{\ell}, \mathbf{H}^{\ell})),$$

- Пересчитываются embedding для новых вершин, и новая матрица смежности

$$\mathbf{H}^{\ell+1} = {\mathbf{S}^{\ell+1}}^{T}\mathbf{H}^{\ell} \qquad \text{and} \qquad \mathbf{A}^{\ell+1} = {\mathbf{S}^{\ell+1}}^{T}\mathbf{A}^{\ell}\mathbf{S}^{\ell+1}.$$

- Новая матрица смежности оказывается полной

Top-k Pooling

- Для каждой вершины посчитаем attention как проекция его эмбединга на обучаемый вектор p

$$s^{\ell+1} = \frac{\mathbf{H}^{\ell} p^{\ell+1}}{\|p^{\ell+1}\|}.$$

- Выбираем k вершин с наибольшим полученным attention
- Оставляем в графе только найденные вершины
- Расширение метода Self-attention pooling

$$s^{\ell+1} = \sigma(\mathrm{GCN}(\mathbf{A}^{\ell}, \mathbf{H}^{\ell})).$$

Edge Pooling

- Считаем attention для рёбер

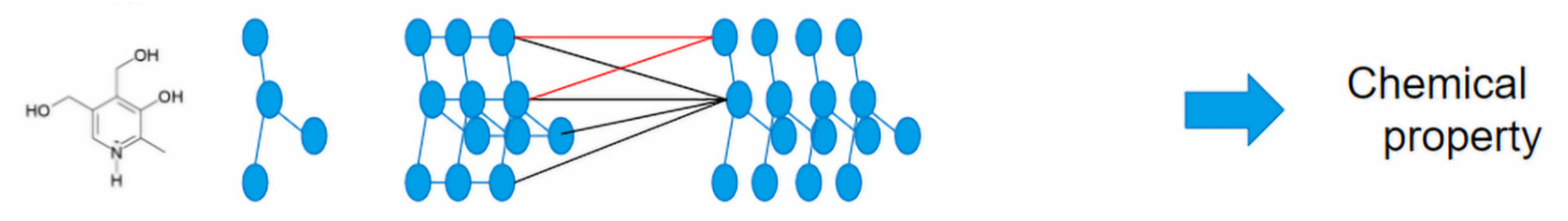
$$s^{\ell+1}((v,u) \in \mathcal{E}_g) = \sigma(\mathbf{w}^T[\mathbf{h}_v^\ell, \mathbf{h}_u^\ell] + \mathbf{b}).$$

- Сжимаем две вершины, соединённые этим ребром в одну
- Повторяем итерационно

Топологические pooling-и

- GRACLUS — алгоритм основанный на спектральной кластеризации
- Non-negative Factorization Matrix Pooling — pooling основанный на NFM факторизации матрицы смежности.

Graph embedding

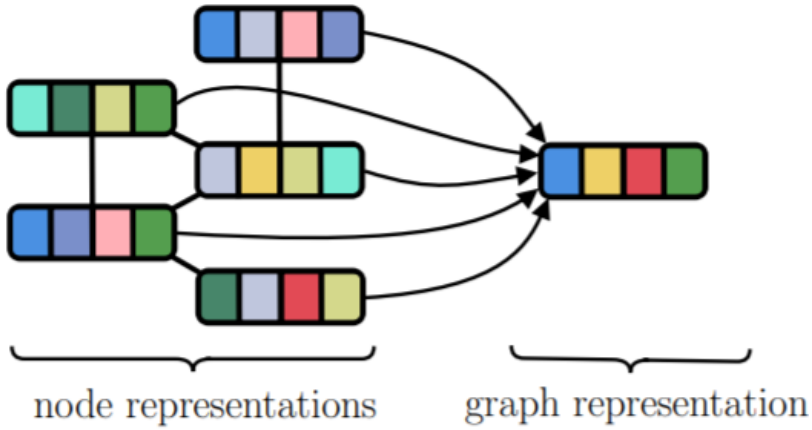


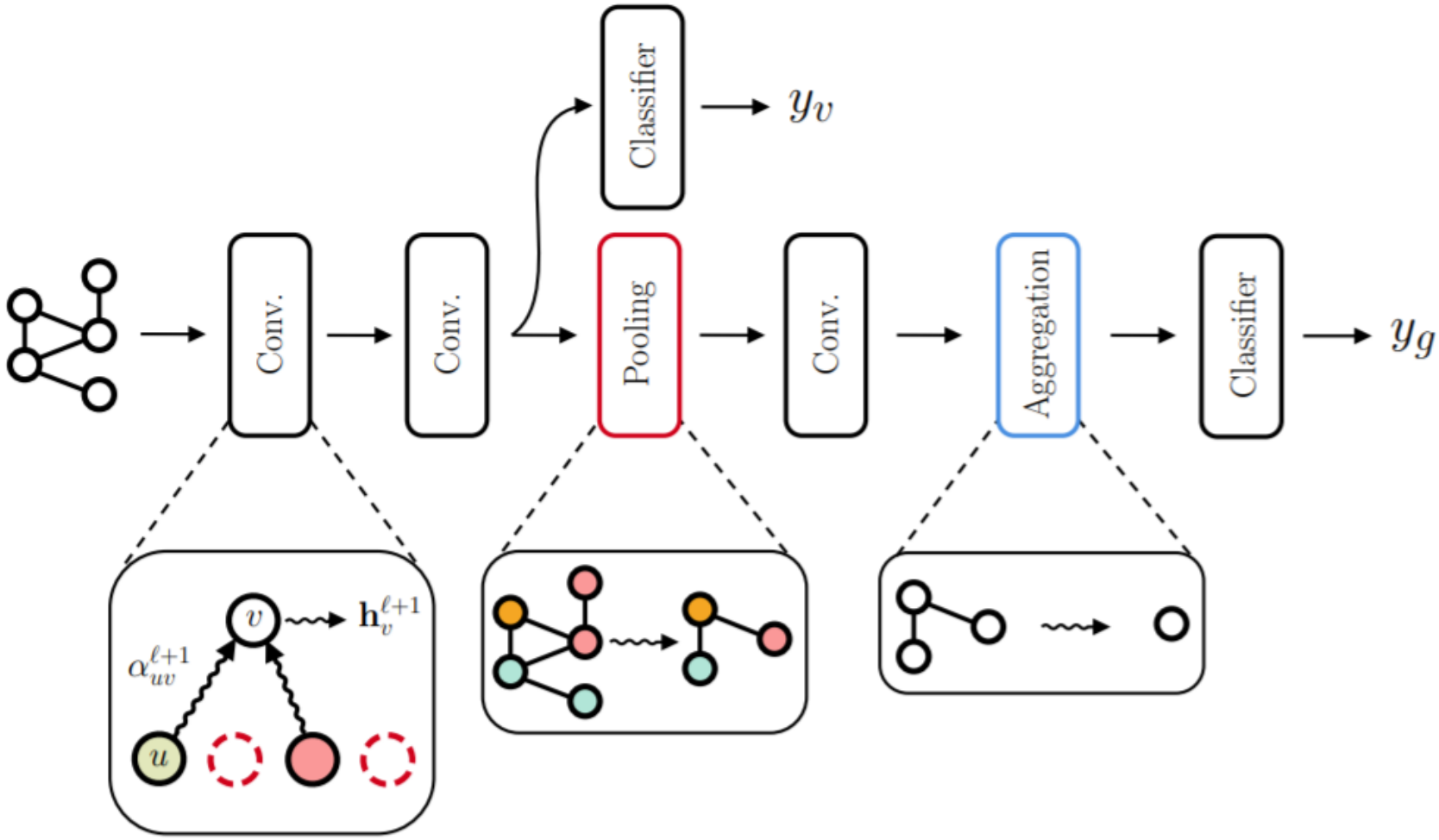
- Хотим построить embedding для всего графа
- Нужно агрегировать embedding для нод
- Можно делать polling графа, пока не останется одна вершина

Graph embedding

$$\mathbf{h}_g^\ell = \Psi\Big(\{f(\mathbf{h}_v^\ell) \mid v \in \mathcal{V}_g\}\Big),$$

- Выбрать Ψ как сумму, максимум или минимум, а f — тождественную функцию.
- f как нейронная сеть




Classifier
y_v
Conv.
Conv.
Pooling
Conv.
Aggregation
Classifier
y_g
v
$\mathbf{h}_v^{\ell+1}$
$\alpha_{uv}^{\ell+1}$
u

Список литературы

- https://arxiv.org/pdf/1912.12693.pdf
- https://arxiv.org/pdf/1609.02907.pdf
- https://arxiv.org/pdf/1812.04202.pdf
- https://arxiv.org/pdf/1801.10247.pdf
- https://arxiv.org/pdf/1706.02216.pdf