Глубинное обучение в анализе графовых данных

5. Передача информации

Высшая Школа Экономики 22/23

в предыдущих сериях…

Свойства декодеров

- симметрия (асимметрия)
- инверсионность
- композитность

$$\mathrm{DEC}(\mathbf{z}_u, \tau, \mathbf{z}_v) = \mathrm{DEC}(\mathbf{z}_v, \tau, \mathbf{z}_u)$$

$$-\|\mathbf{z}_u + \mathbf{r}_\tau - \mathbf{z}_v\| = -\|\mathbf{z}_v + \mathbf{r}_\tau - \mathbf{z}_r\|$$

$$\Rightarrow$$

$$-\mathbf{r}_\tau = \mathbf{r}_\tau$$

$$\Rightarrow$$

$$\mathbf{r}_\tau = 0.$$

PageRank

Алгоритм ранжирования страниц в интернете

С помощью метода power iteration решаем задачу вида r = Gr

При этом в “гугл” матрице имеется dumping factor \beta, который позволяет решить проблему циклов в графе, и также тупики предобрабатываются заранее

Передача информации

Вспомним об эмбеддингах

Эмбеддинги вершин позволяют получать удобные для обучения представления вершин

Раньше мы смотрели эмбеддинг вида ENC(v) = Z[i] (просто брали эмбеддинг вершины из эмбеддингового пространства)

Подход имел ряд проблем - тяжело считается, получая новые вершины эмбеддинга для них не найти, не используются признаки внутри вершины

Эмбеддинги и GNN

$$\mathrm{ENC}(v) = \mathrm{MLP}$$

Глубинные графовые энкодеры

Общая схема очень похожа на классические DL подходы

INPUT: граф

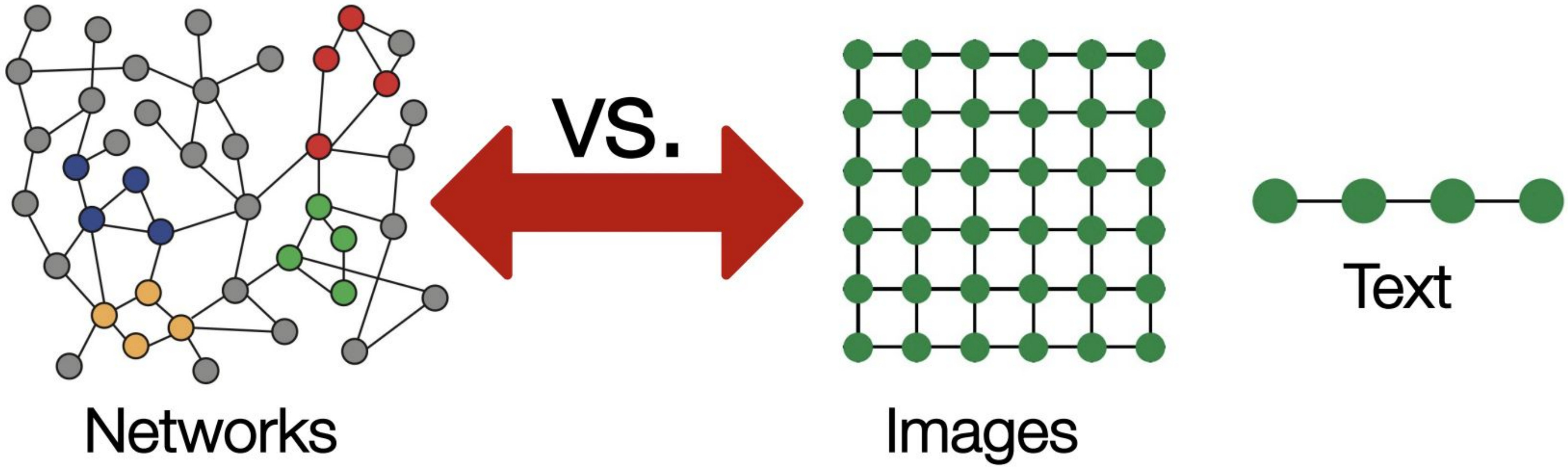
Проходит через свертки, функции активации, регуляризационные юниты, по итогу получаются эмбеддинги вершин

Таким подходом можно работать еще и подграфами (и графами)

Задачи

- Классификация вершин
- Предсказание связей
- Детекция сообществ
- Похожесть сетей

Основная проблема



Классический DL

$$f(\boldsymbol{x}) = W_2(W_1\boldsymbol{x})$$
$$h(x) = W_1\boldsymbol{x}$$
$$g(z) = W_2 z$$

- Решаем задачу $\min_{\Theta} \mathcal{L}(\boldsymbol{y}, f(\boldsymbol{x}))$
- f - линейный слой, сеть, графовая сеть
- Сэмплируем минибатч x
- Делаем forward pass (считаем L)
- Делаем backward pass ($\nabla_w \mathcal{L}$) используя chain rule
- С помощью стохастического градиентного спуска оптимизируем

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}g}{\mathrm{d}h} \cdot \frac{\mathrm{d}h}{\mathrm{d}x}$$

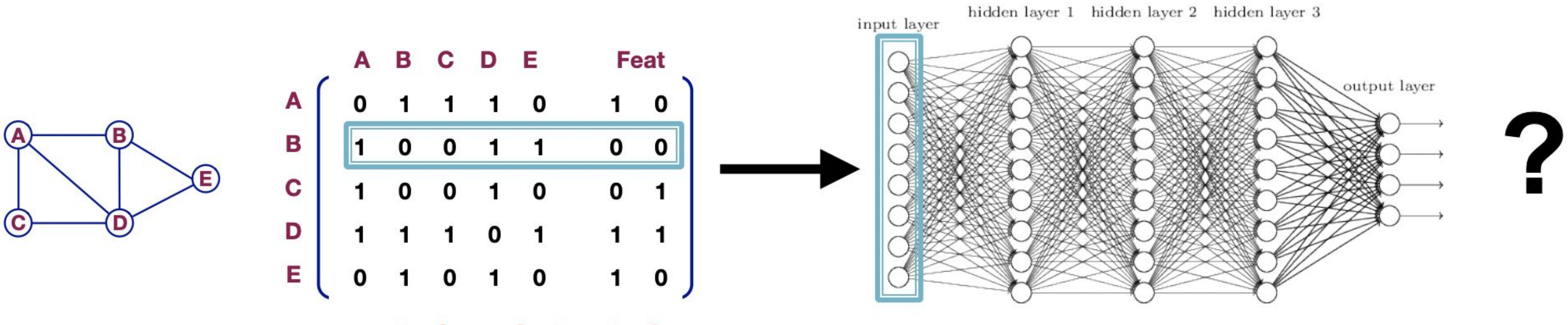
$$\boldsymbol{x}^{(l+1)} = \sigma(W_l \boldsymbol{x}^{(l)} + b^l)$$

О графах

- V - вершины
- A - бинарная матрица смежности
- X \in R ^{m x |V|} - матрица признаков вершин
- v - вершина в V, N(v) - соседство вершины v

Наивный подход:

Наивный подход



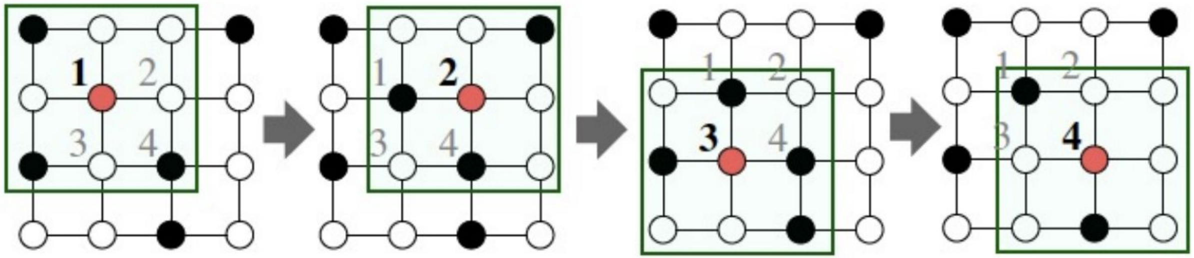
Проблемы наивного подхода

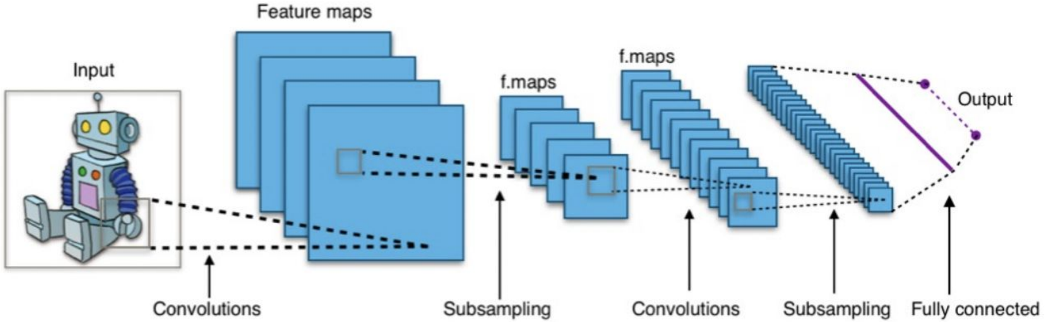
- Долго
- Зависит от порядка вершин
- На разных графах работать не будет

Интуиция

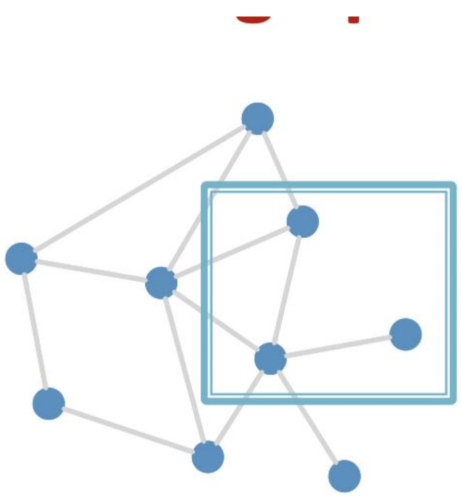
Цель - генерализовать какие-то топологические мотивы с помощью некоторого аналога “сверток”

Обычные свертки

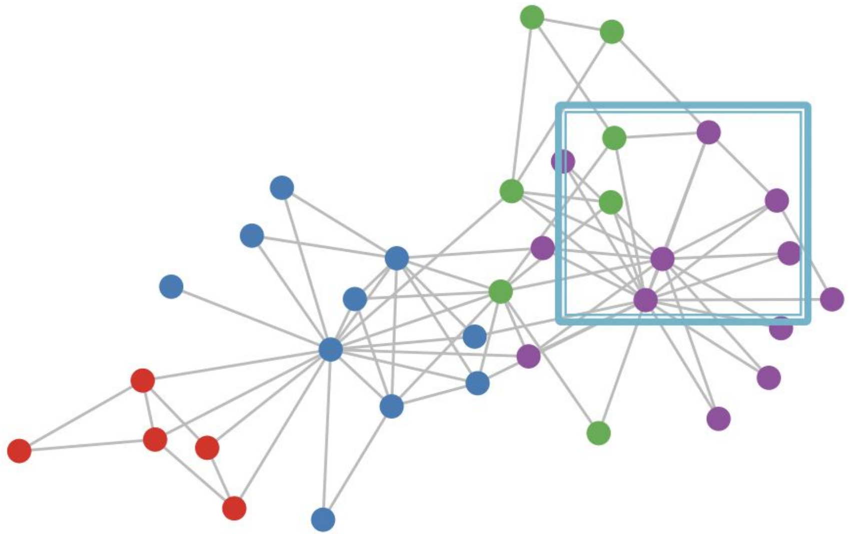




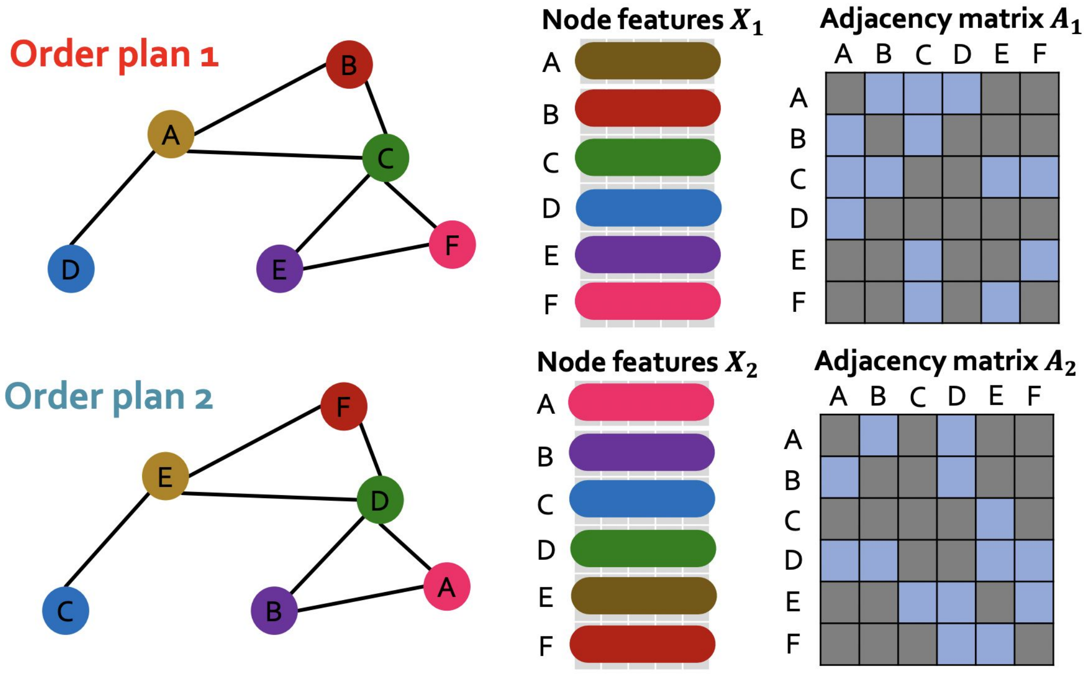
Проблема с применение обычных сверток



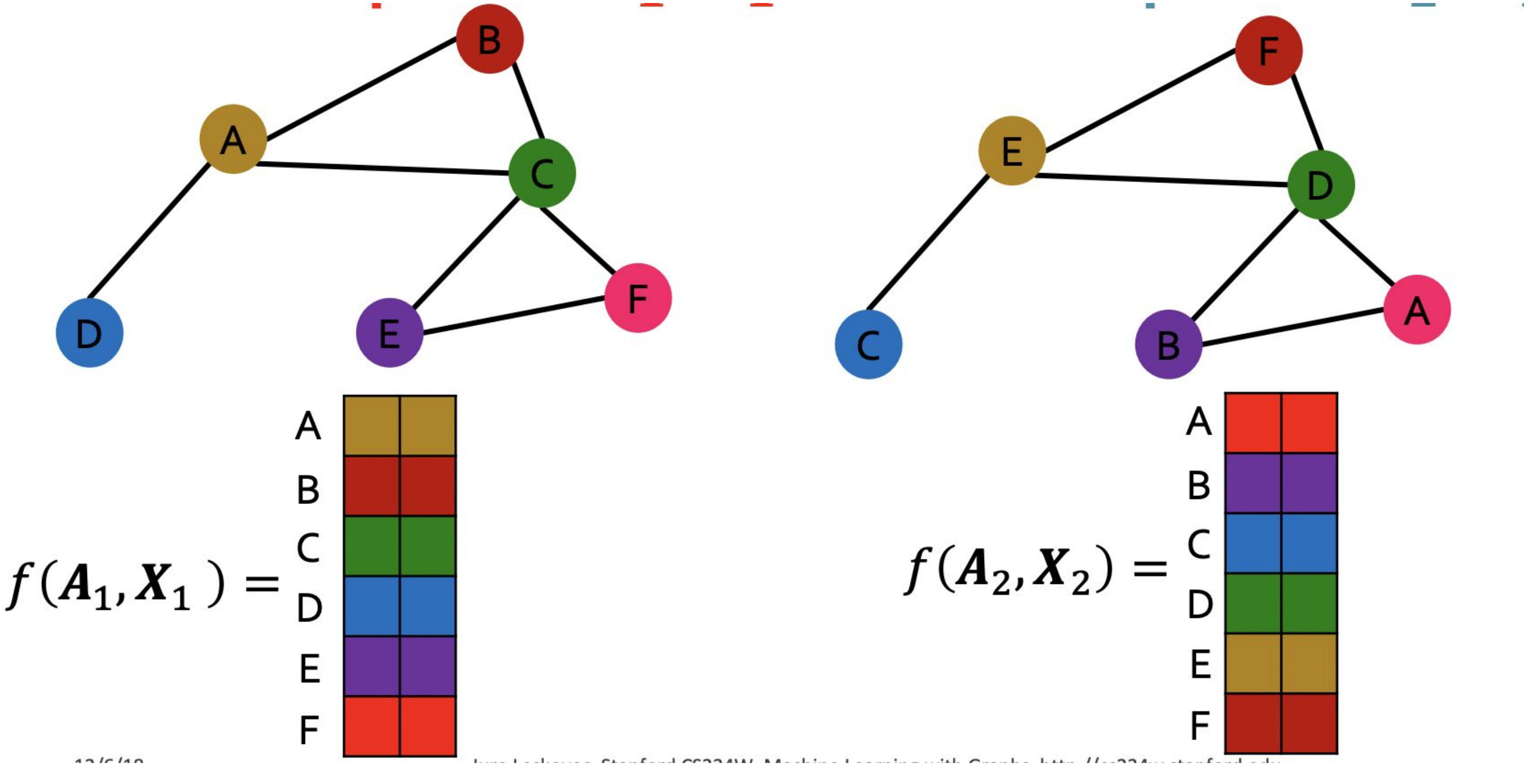
or this:



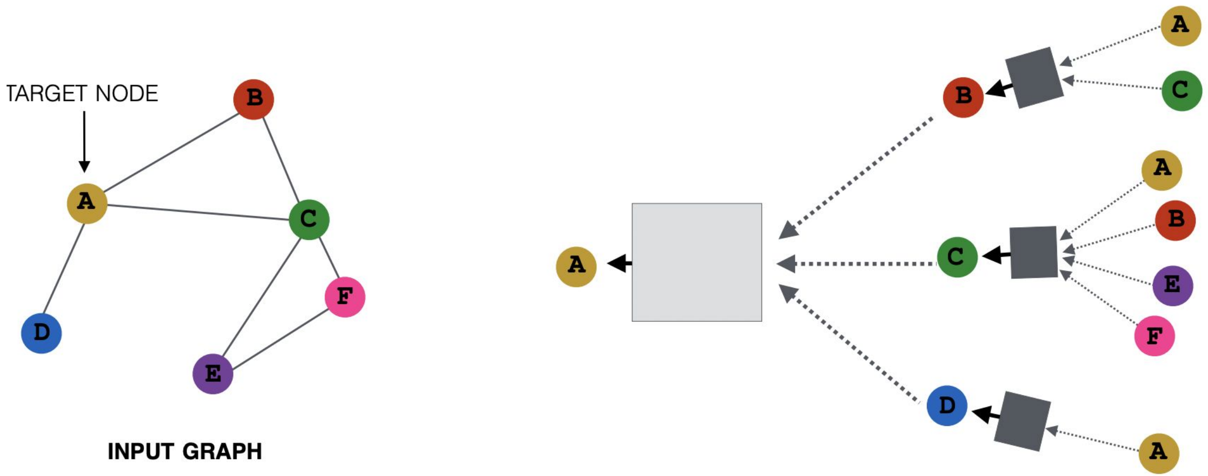
Инвариант перестановок



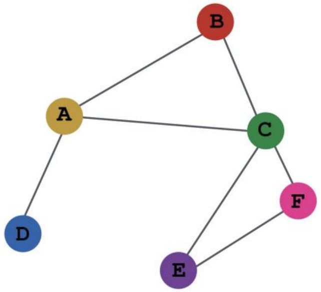
Вершины

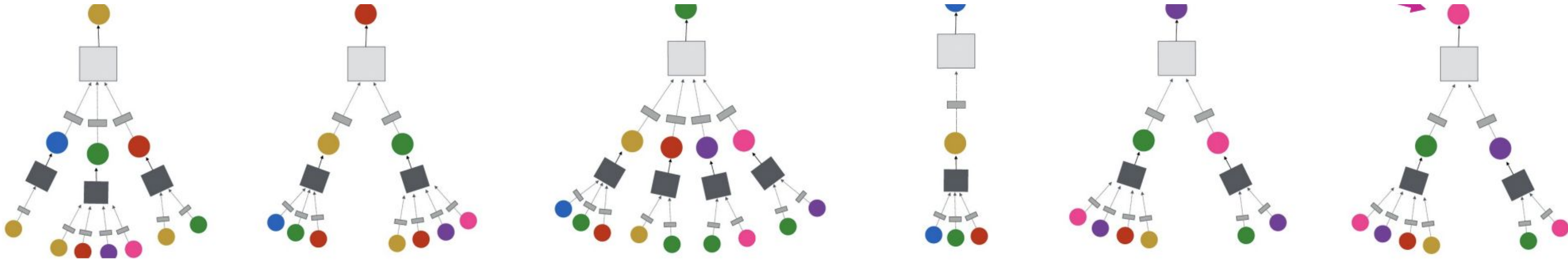

B
A
C
D
E
F
F
E
D
C
B
A
A
B
C
D
E
F
$f(\boldsymbol{A}_1, \boldsymbol{X}_1) =$
A
B
C
D
E
F
$f(\boldsymbol{A}_2, \boldsymbol{X}_2) =$

Основная идея - аггрегация

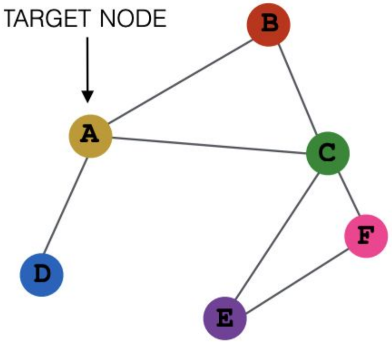


Агрегация соседей

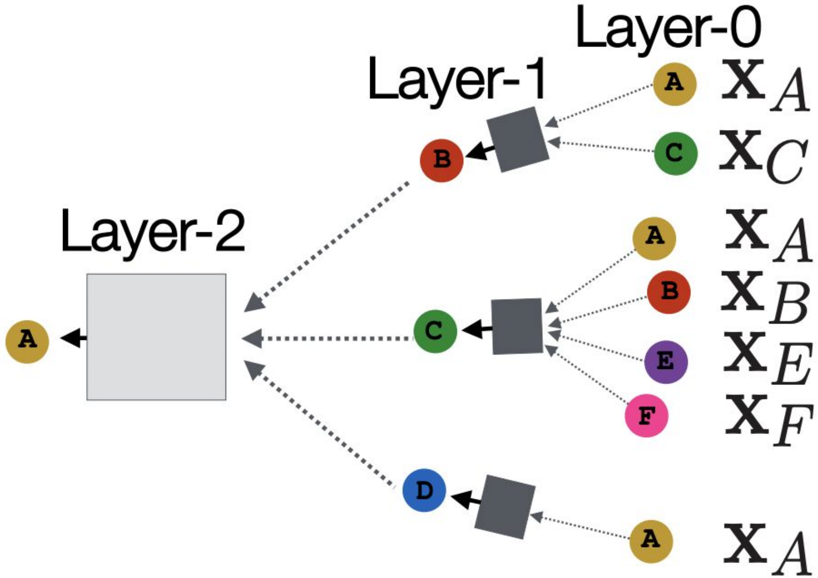




Как считать



INPUT GRAPH



Как считать

$$\boxed{h_v^0 = x_v}$$

Initial 0-th layer embeddings are equal to node features

embedding of v at layer k

$$h_v^{(k+1)} = \boxed{\sigma}\left(W_k \boxed{\sum_{u \in N(v)} \frac{h_u^{(k)}}{|N(v)|}} + B_k \boxed{h_v^{(k)}}\right), \forall k \in \{0, \dots, \boxed{K} - 1\}$$

Total number of layers

$$\boxed{z_v} = h_v^{(K)}$$

Average of neighbor's previous layer embeddings

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Notice summation is a permutation invariant pooling/aggregation.