Домашнее задание 2

Глубинное обучение в анализе графовых данных.

Владимир Лузин

9 декабря 2024 г.

1 Задание 1. Построение графа вычислений.

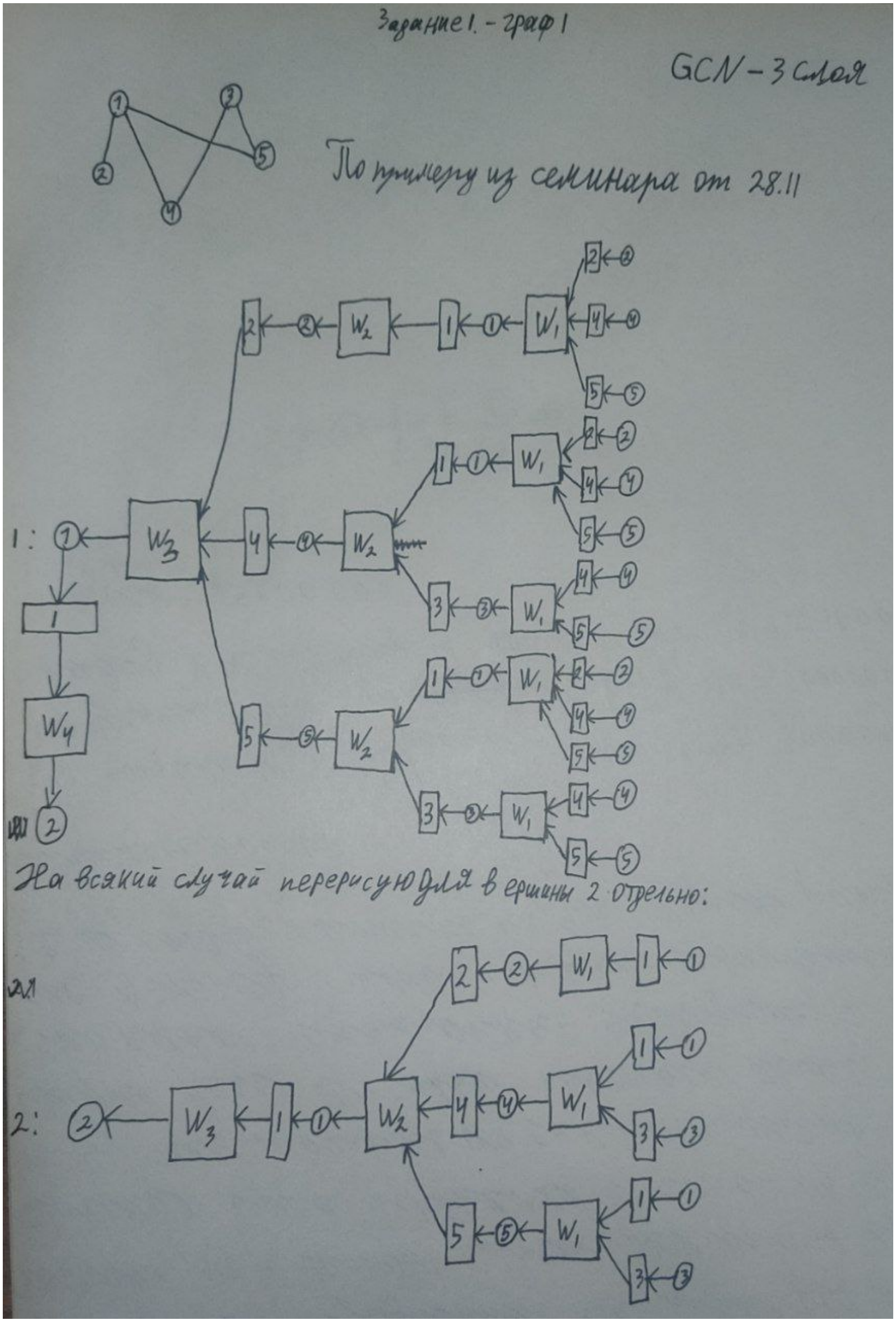


Рис. 1: Граф 1

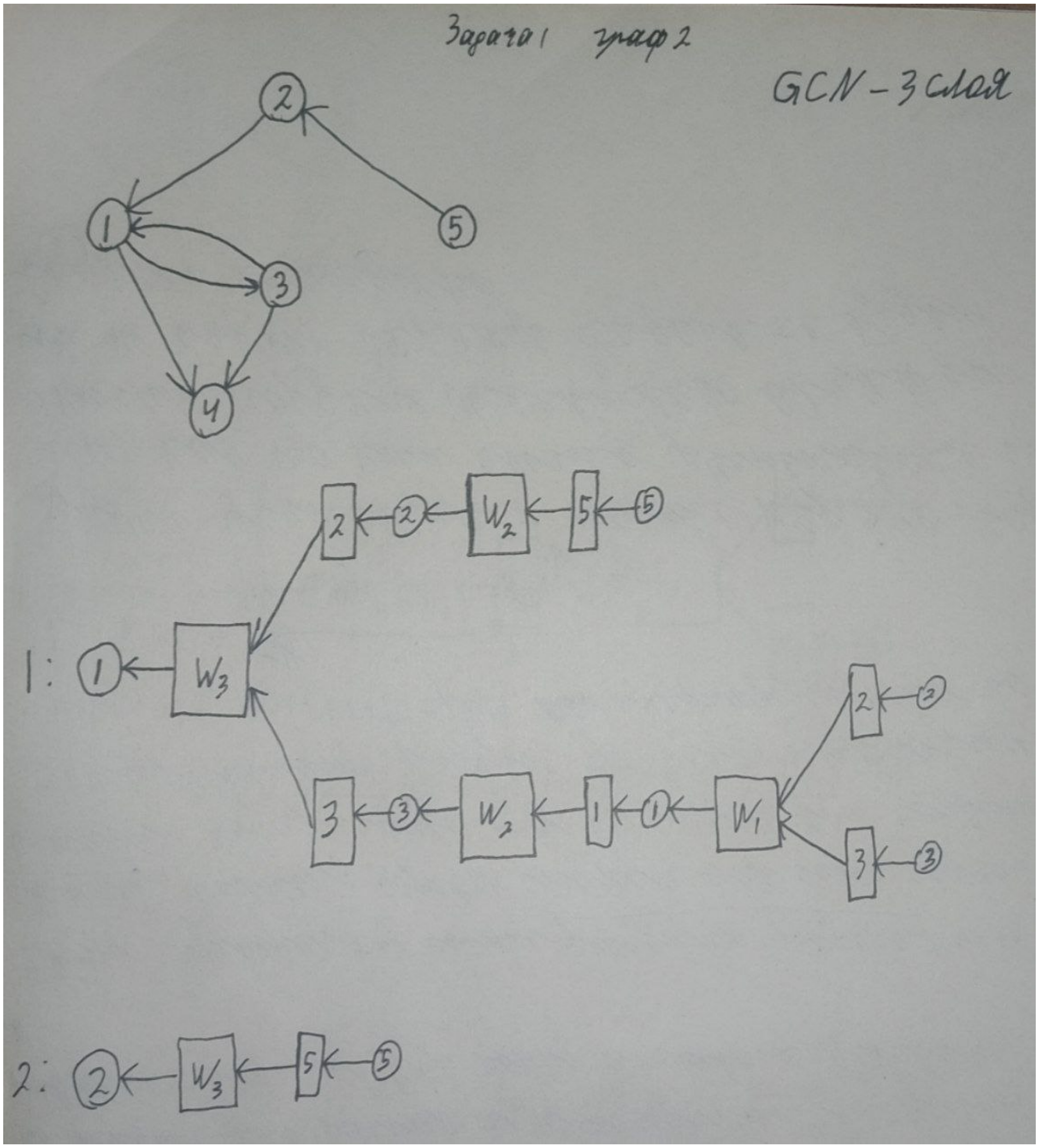


Рис. 2: Граф 2

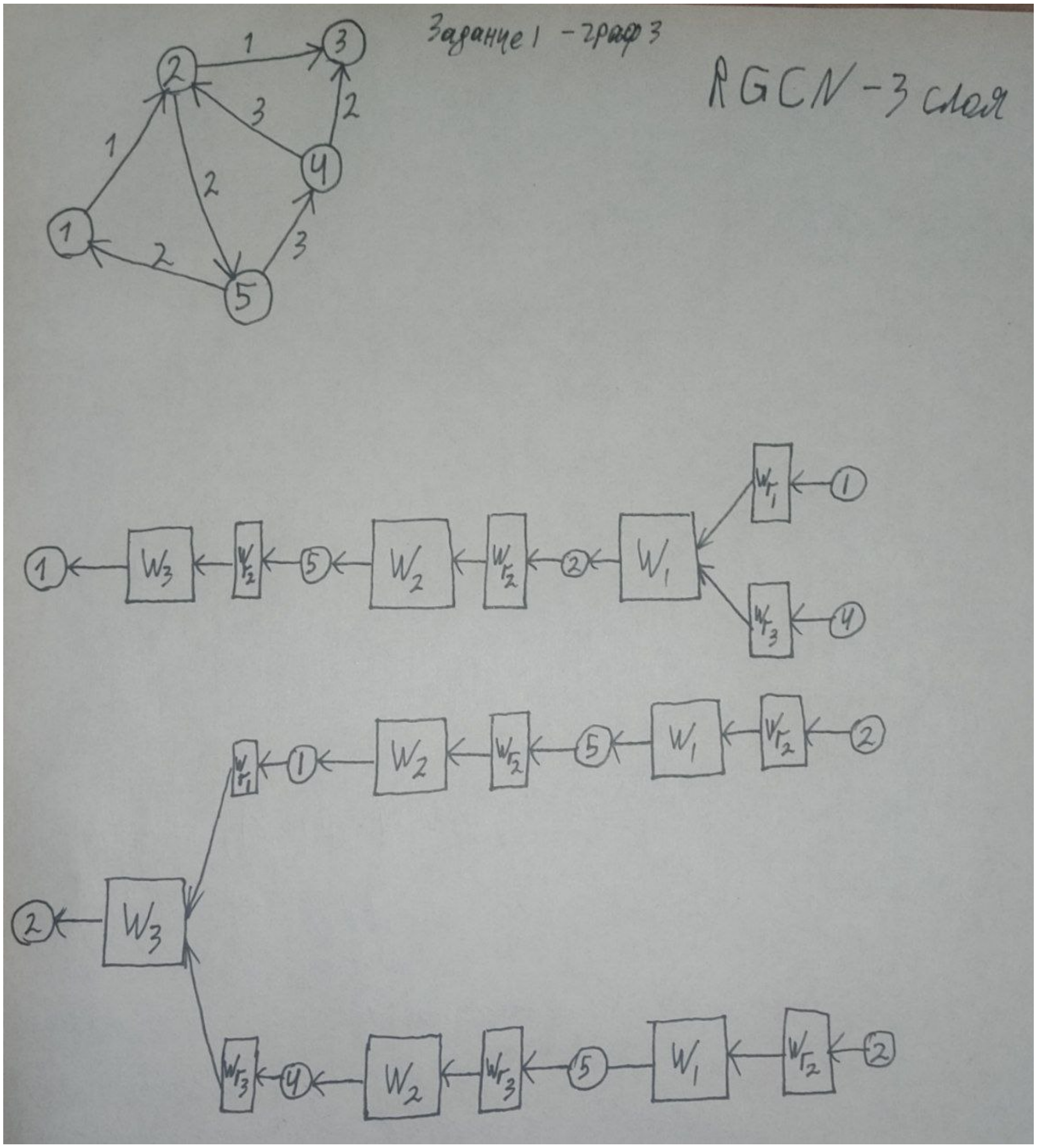


Рис. 3: Граф 3

2 Задание 2. Когда WL не работает.

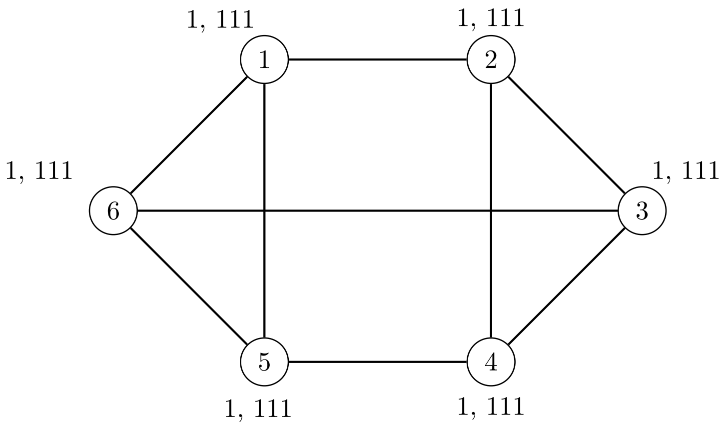
Насколько я понял, требуется привести конкретный пример двух таких графов.

Я смог придумать два неизоморфных графа. Неизоморфизм можно доказать тем, что в первом графе есть треугольник-подграф, а во втором — его нет. В остальном у обоих графов одинаковое количество вершин и ребер, одинаковое количество компонент и степени вершин в каждом графе также одинаковые.

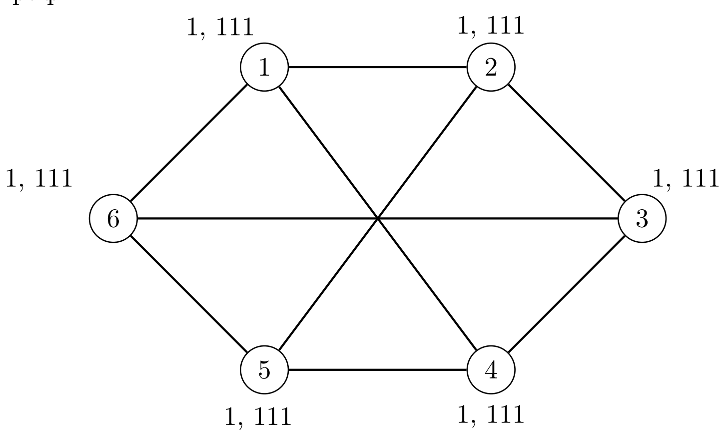
Оба графа имеют одинаковые WL ядра, поскольку они оба симметричны и имеют одинаковое число вершин, степеней вершин (3 для каждой вершины) и структуру соседей.

Но на всякий случай вручную проведу 3 итерации.

Граф 1



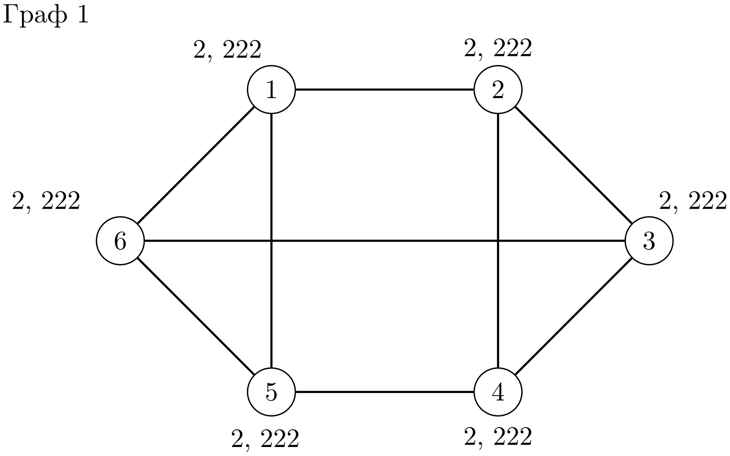
Граф 2



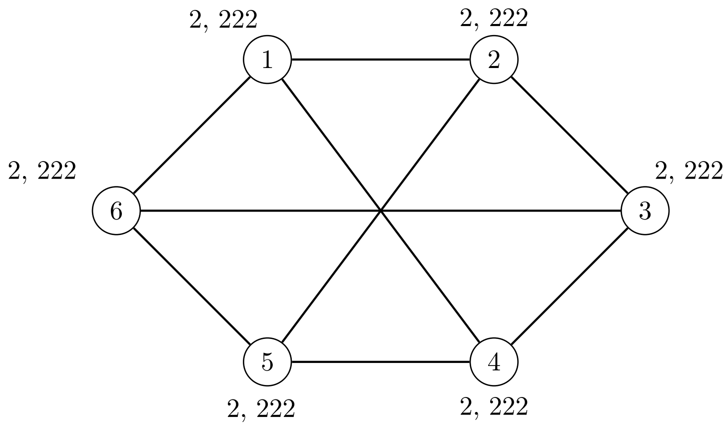
агрегация	хеши	количество
1, 111	2	6

Таблица 1: хеш таблица - граф 1 - итерация 1

Граф 1



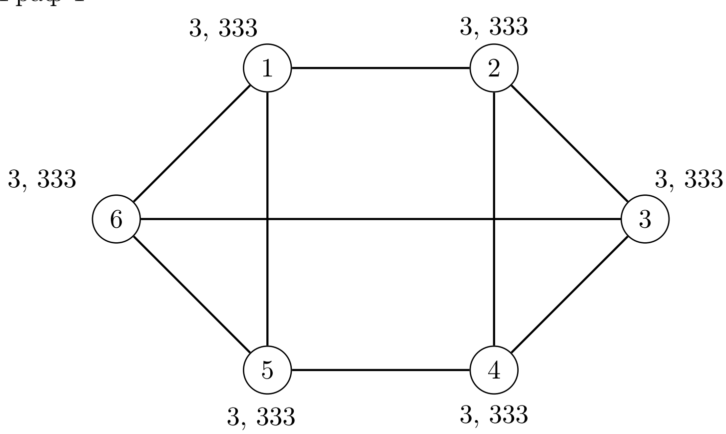
Граф 2



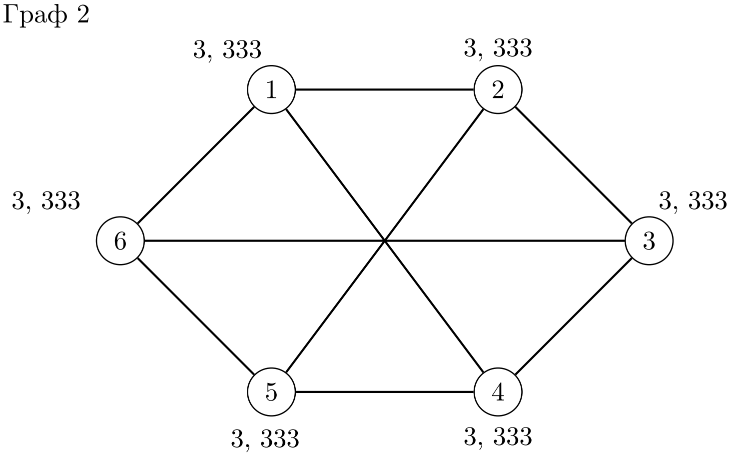
агрегация	хеши	количество
2, 222	3	6

Таблица 2: хеш таблица - граф 1 - итерация 2

Граф 1



Граф 2



агрегация	хеши	количество
3, 333	3	6

Таблица 3: хеш таблица - граф 1 - итерация 2

Итоговые вектора графов

$$f_1 = \langle 6, 6, 6 \rangle$$

$$f_2 = \langle 6, 6, 6 \rangle$$

При k итераций вектор будет выглядеть как вектор из $()^k$. То есть два графа, несмотря на их неизоморфичность, будут иметь одинаковые wl ядра.

Для улучшение алгоритма, я придумал несколько вариантов. Сложно сказать какие из них являются тривиальными, а какие нет, поэтому приведу несколько вариантов:

1. Добавление в графы информацию о кратчайшем узле. То есть минимальное расстояние по любому из рёбер вершины, приводящее к этой же вершине (по одному ребру нельзя ходить дважды). Например для первой вершины первого графа это расстояние будет 3, а для первой вершины второго графа это расстояние будет 4. (у остальных вершин второго графа также не будет вершин с расстоянием меньше 4) То есть графы уже будут отличаться по улучшенному wl ядру на первом шагу.

2. Также можно добавлять не минимальный цикл, а количество циклов заданной длины. Например, можно добавить параметр t=4. Тогда для каждой вершины будут добавляться циклы длины 3 и 4 (циклы длины 2 быть не может, так как опять нельзя ходить дважды по одному ребру). Также лучше не учитывать повторы (иначе циклы будут всегда повторятся дважды). Тогда для того же примера, у вершины 1 графа 1 будет дополнительный вектор (1, 2), а у вершины 1 графа будет дополнительный вектор (0, 3). Причёв у второго графа не будет вершин с дополнительным вектором (1, 2). То есть два графа из примера опять будут различимы после первой итерации

3. Ещё можно раскрасить граф и добавить в итоговый вектор всего графа хроматическое число графа (наименьшее количество цветов для раскраски). У первого графа из примера r=3, а второго r=2. То есть итоговые вектора для 3 итераций будут $\langle 6, 6, 6, 3 \rangle$ и $\langle 6, 6, 6, 2 \rangle$ соответственно, что позволит различить два графа из моего примера. (Выглядит как наименее тривиальный вариант)

Надеюсь хотя бы один из моих вариантов оказался "не тривиальным".

3 Задание 3. Общая форма для передачи сообщений.

Изначальная формула

$$\mathbf{h}_u^{(k)} = \sigma \left(\mathbf{W}_{\text{self}}^{(k)} \mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)} \right)$$

Где

- $\mathbf{h}_u^{(k)}$ — выход на k-м слое для вершины u
- σ — нелинейная функция активации
- $\mathbf{W}_{\text{self}}^{(k)}$ — матрица весов для самих вершин
- $\mathbf{W}_{\text{neigh}}^{(k)}$ — матрица весов для соседей
- $\mathcal{N}(u)$ — множество соседей вершины u
- $\mathbf{b}^{(k)}$ — bias

Опущу b

$$\mathbf{h}_u^{(k)} = \sigma \left(\mathbf{W}_{\text{self}}^{(k)} \mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} \right)$$

Вектор выходных значений для всех вершин можно записать как матрицу $\mathbf{H}^{(k)}$ (Цикл по всем строкам выходных значений каждой вершины. Только теперь строка - выходные значения для одной вершины). Сумму по соседям в свою очередь можно заменить на матричное произведение матрицы смежности A

То есть

$$\sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} \quad = \quad A\mathbf{H}^{(k-1)}.$$

Итоговая формула:

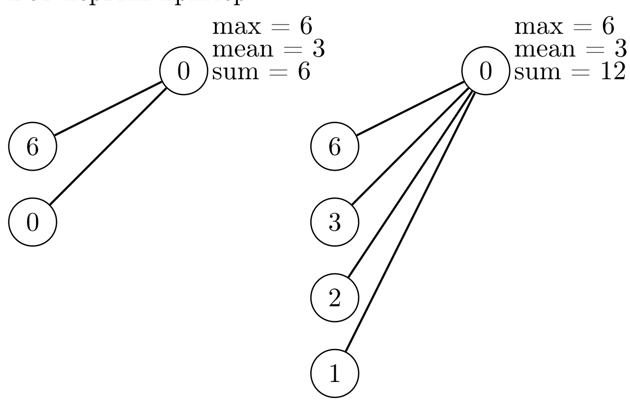
$$\mathbf{H}^{(k)} = \sigma \left(\mathbf{W}_{\text{self}}^{(k)} \mathbf{H}^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} A\mathbf{H}^{(k-1)} \right).$$

- $\mathbf{H}^{(k)}$ — матрица скрытых состояний на слое k
- A — матрица смежности

4 Задание 4. Особенности функции аггрегации

Придумал даже два примера. Значения внутри вершин обозначает h.

Вот первый пример:



А вот второй: