

# Глубинное обучение в анализе графовых данных

## 0. Введение

## О курсе. Организационное.

- 12 лекций и семинаров
- 5 домашних заданий (3 теоретических, 2 практических)
- Лекции по четвергам 14:40
- Семинары по четвергам 16:10

## О курсе. Оценивание

- Итог =  $0.7 O_n + 0.3 O_{\text{э}}$ , округление арифметическое
- $O_n = (\text{Кол-во баллов}) / 7$
- 3 практ. дз (10, 15, 20), 2 теор.дз (10, 15)
- Экзамен - защита доп. дз (5) + устный ответ (5)
- Автоматы с 6.5 за накоп (при условии всех сданных\* домашних заданий)
- Блоков нет

## О курсе. Дедлайны

- Дедлайны мягкие, за каждый день просрока -10%, после трех дней приходит жесткий
- Два раза за курс сдать можно после жесткого (дается еще 4 дня)
- Сдавать после 7 дней нельзя
- Для сдачи\* учитывается оценка без штрафа

## О курсе. Выполнение домашних заданий

- Домашние задания выполняются индивидуально
- В случае подозрения на списывание студент вызывается на защиту

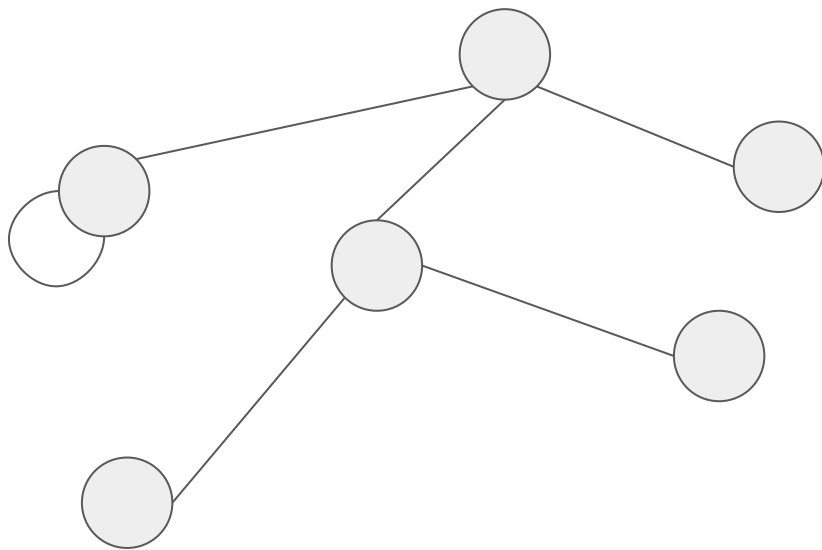
## О курсе. Преподаватели

- Лектор - Валитов Эльдар (AliExpress Russia)
- Семинарист - Павел Федоров (ВТБ, Whalemap)

# Глубинное обучение в анализе графовых данных

1. Графы и анализ данных.

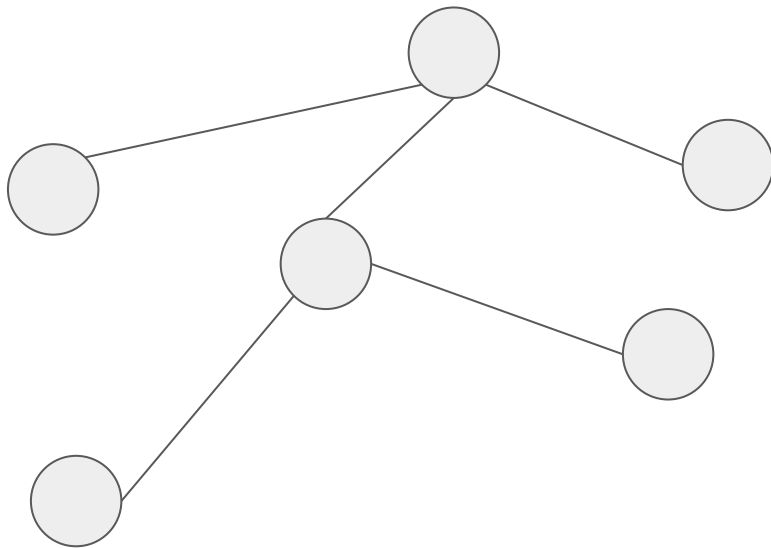
Что такое графы





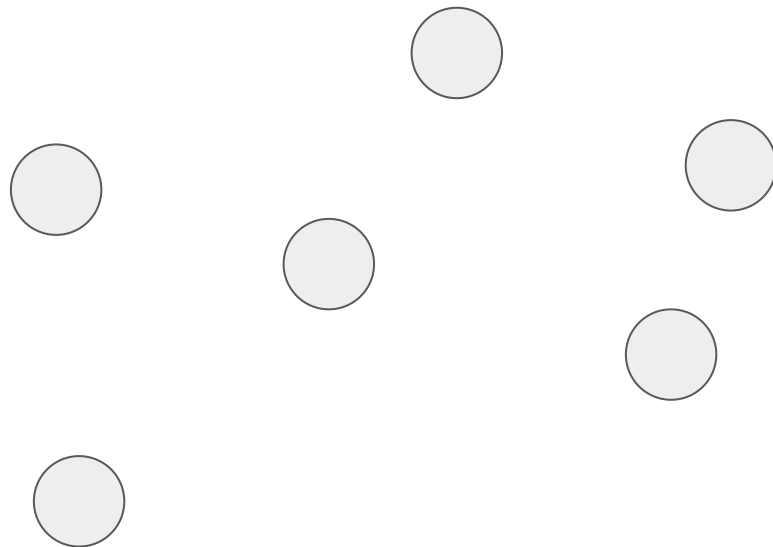
# Что такое графы

$G = (V, E)$ , где  $V$  - множество вершин,  $E$  - множество ребер



# Что такое графы

Граф?



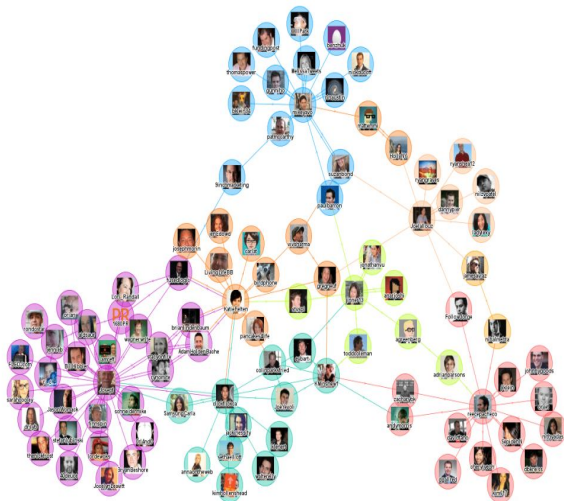
## Зачем нужны графы в контексте данных

- Графы являются обобщенным способом представления любой информации связанной с наличием взаимодействия между объектами
- Существует большое количество различных методов анализа графов, поэтому удобно приводить данные из различных областей к одному формату

# Примеры графовых данных



Схема метрополитена



Социальный граф

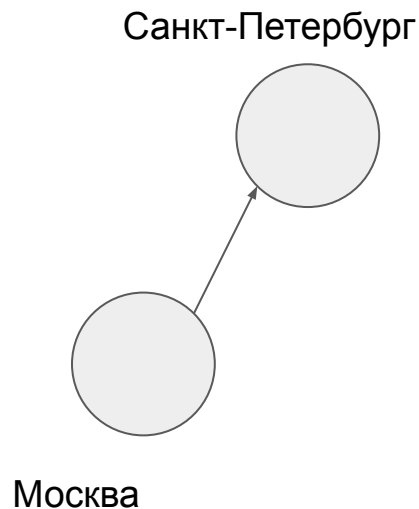
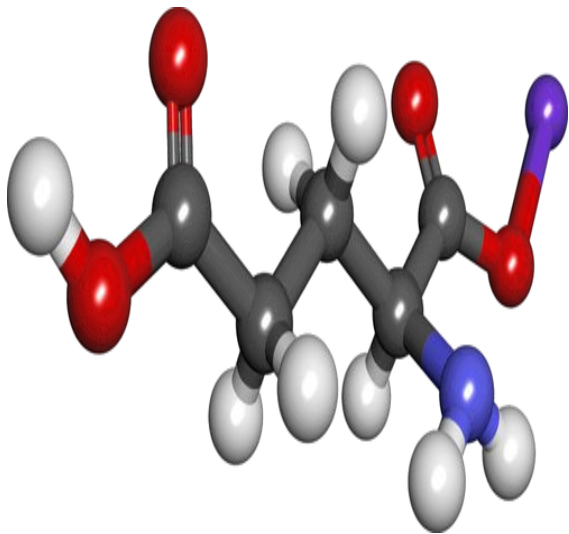
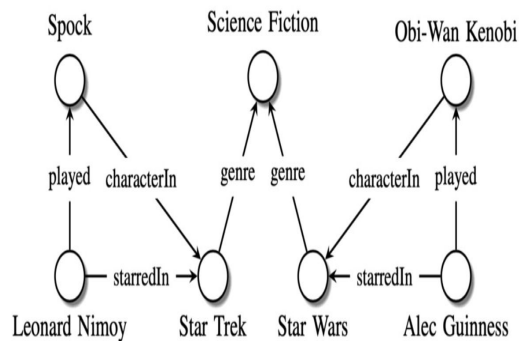


схема трафика

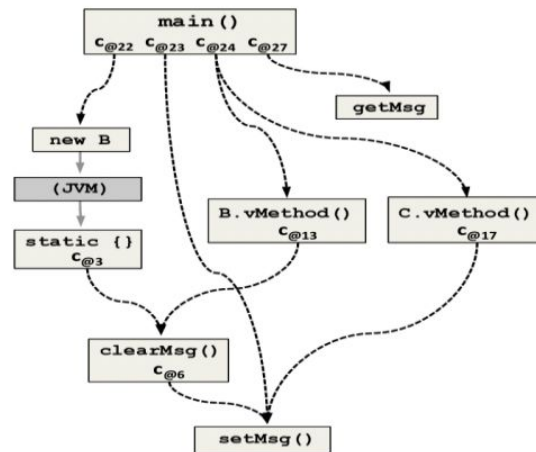
# Примеры графовых данных



молекулы



графы знаний



код

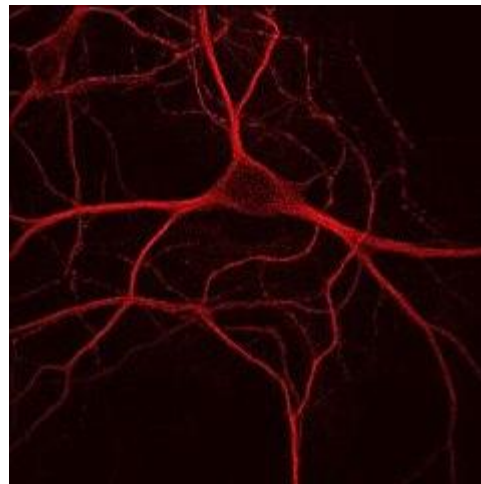
# Примеры графовых данных



компьютерная сеть



интернет



сети нейронов

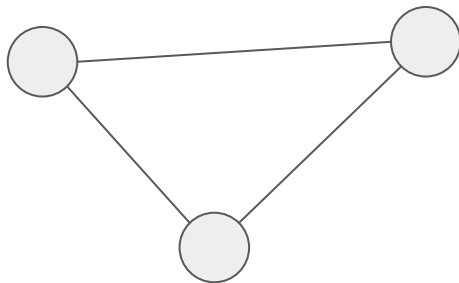
# Мотивация

Основные применения глубинного обучения сегодня - последовательные данные:

- NLP : “askfnkksnf” - последовательность текста
- CV : структурированная из пикселей картинка
- Audio : звуковые волны

# Мотивация

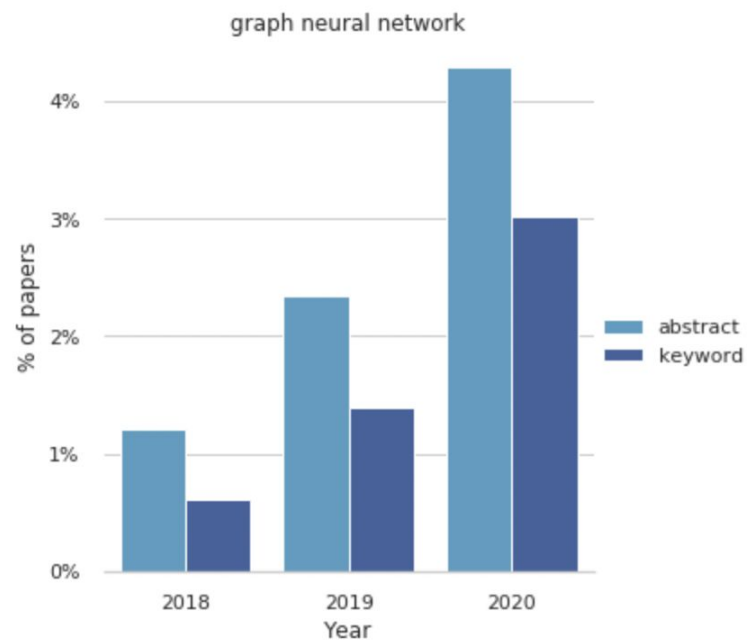
Отличительная особенность графов - связи между объектами, следовательно информацию можно обрабатывать шире и иначе



Где начало и конец?



# Мотивация



[Тренд GNN на ICLR](#)

# Подходы к анализу графовых данных без нейросетей

Многие задачи, решаемые с помощью глубинного обучения существуют давно для графовых данных, соответственно существовали какие-то методы решения этих задач с помощью обычных алгоритмов на графах

# Подходы к анализу графовых данных без нейросетей

[ссылка с примерами](#)

Например:

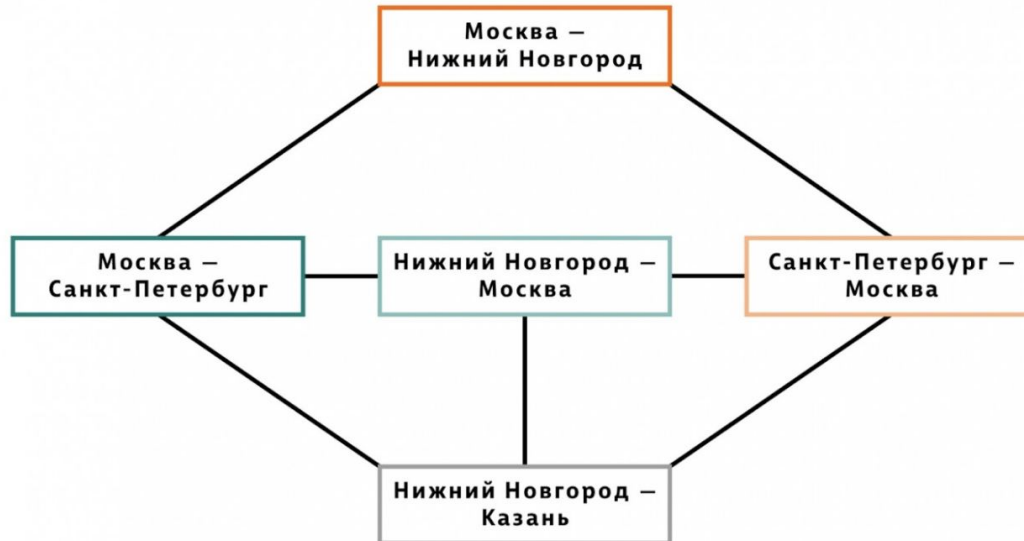
- Рекомендательные системы - обходы



- ❑ Групповые вкусы пользователя - **начальные вершины обхода графа**
- ❑ Делаем 3 случайных шага по графу
- ❑ В каждой вершине есть вероятность телепортироваться в начало обхода
- ❑ Конечные вершины графа - **кандидаты для рекомендации**

# Подходы к анализу графовых данных без нейросетей

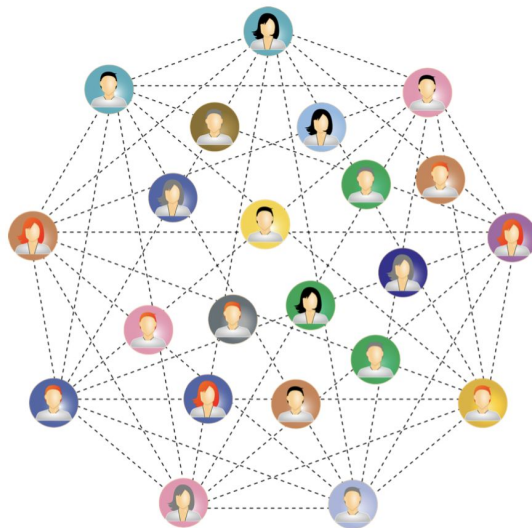
- Vehicle Route Problem - оптимизация маршрутов



# Типы задач на графах

1. Edge-level - задачи связанные с ребрами (классификация связей, регрессия, дополнение графа)
2. Node-level - задачи связанные с вершинами (классификация вершин, регрессия, кластеризация)
3. Graph-level - специфичный для задач на графах тип задач (та же классификация, регрессия, генерация, эволюция)

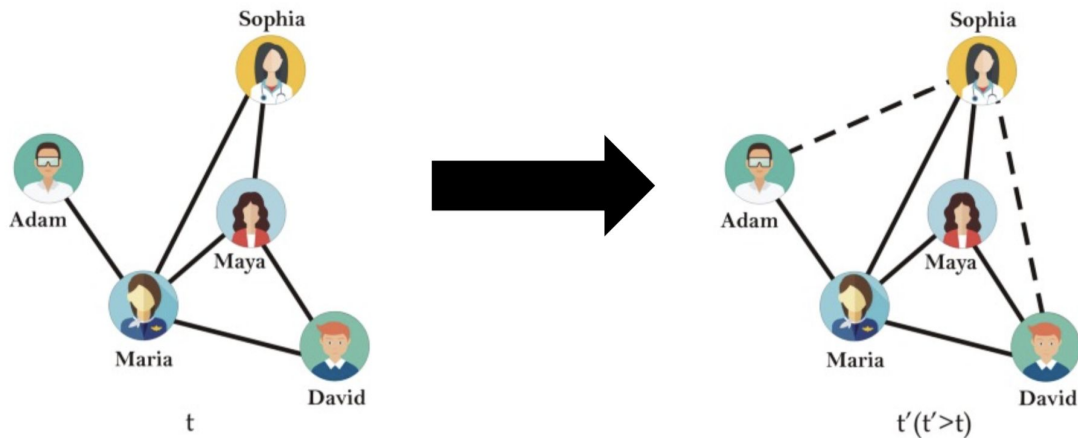
# Node-level задачи



Определение интересов человека по его окружению

Хотим выучить  $f: (V, E) \rightarrow \mathbb{R}^{|V|}$

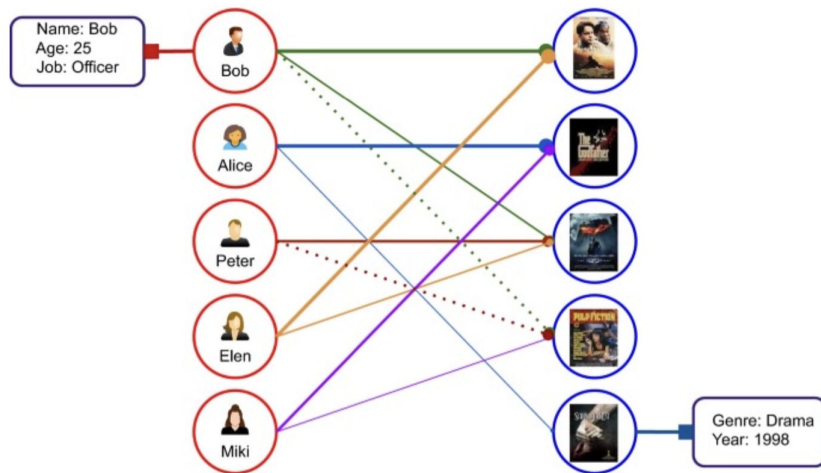
# Edge-level задачи



Сделать рекомендацию на друзей в социальной сети

Для всех пар  $v_i, v_j$  хотим сделать предсказание  $f(v_i, v_j)$

# Edge-level задачи

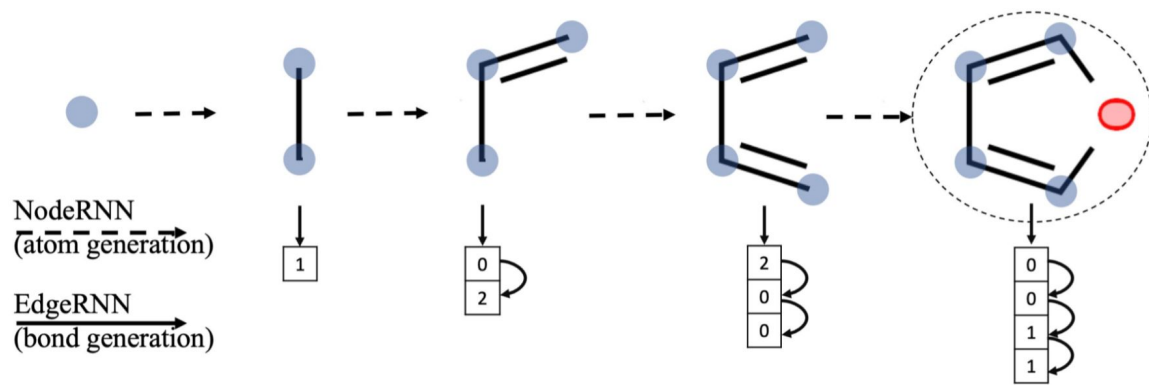


Сделать рекомендацию фильма или музыки

Для всех пар  $v_i, v_j$  хотим сделать предсказание  $f(v_i, v_j)$



# Graph-level задачи



Хотим научиться создавать графы с определенными характеристиками

## И subgraph-level задачи

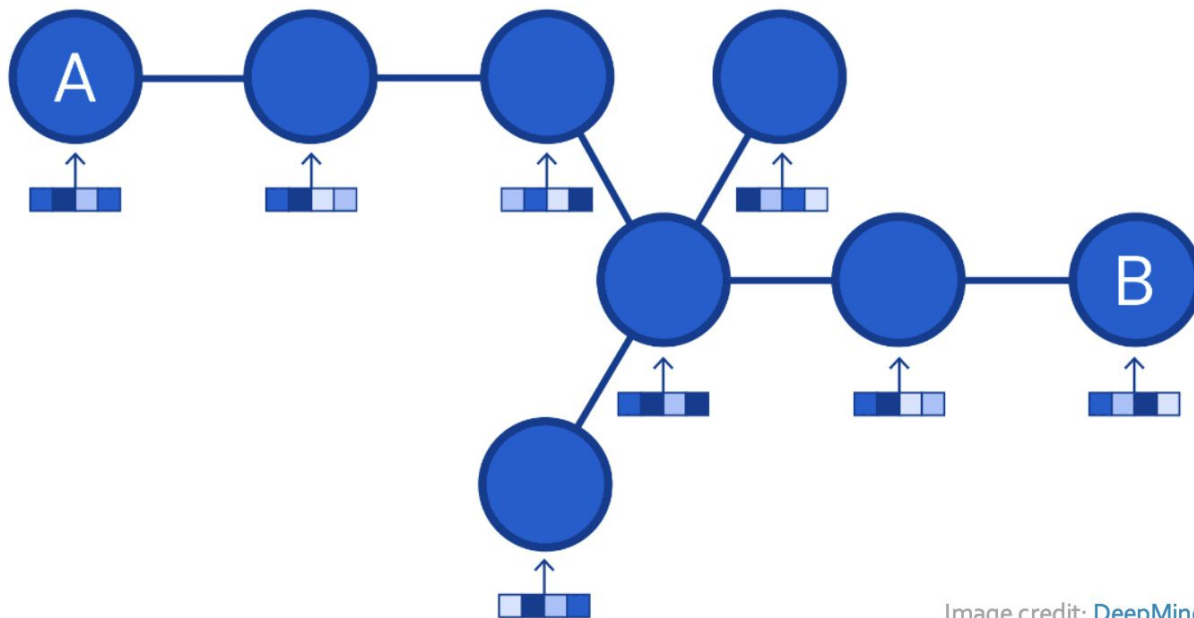


Image credit: [DeepMind](#)

Предсказание времени прибытия машины (разбиение дороги на сегменты (вершины) и доступность из одного сегмента в другой (ребра))

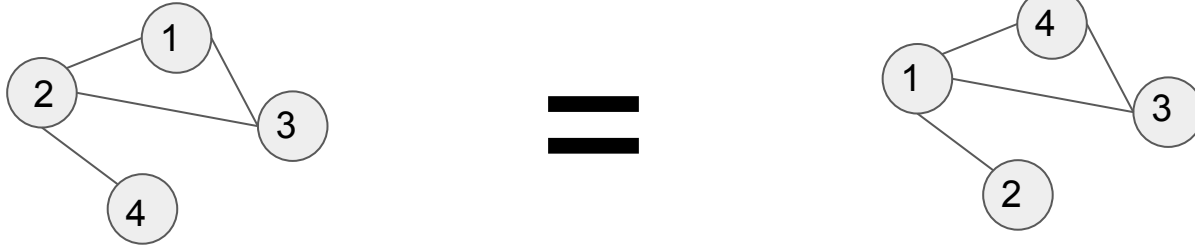
# Особенности алгоритмов на графах

Для работы со сложными системами возникает определенная проблема - выбор представления для использования методов анализа

Существуют некоторые базовые требования к алгоритмам на графах

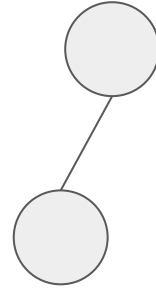
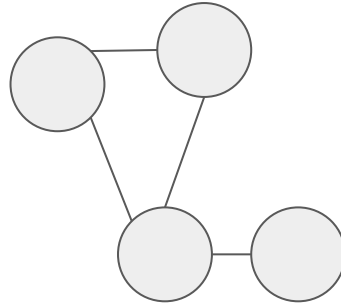
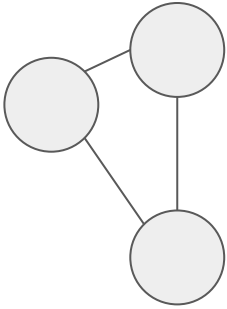
# Особенности алгоритмов на графах

Алгоритмы на графах должны работать независимо от перестановки индексов вершин



# Особенности алгоритмов на графах

Алгоритмы на графах должны уметь принимать на вход графы разных размеров



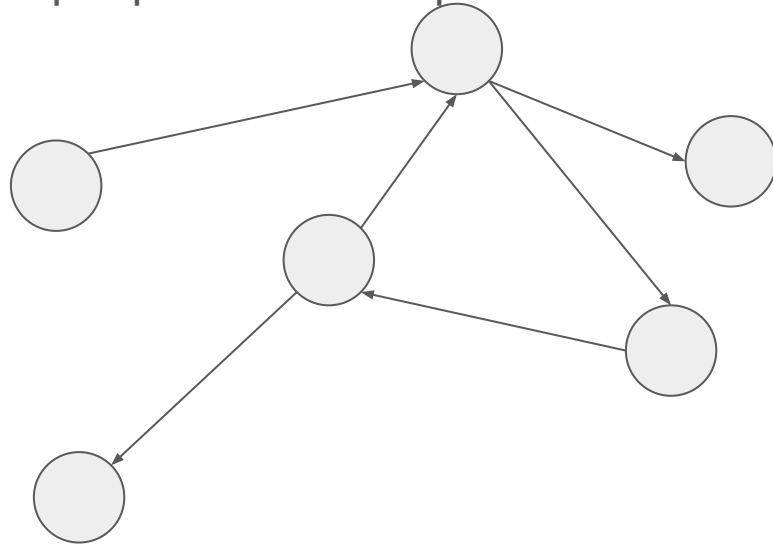
# Выбор представления данных

Важным этапом перед применением глубинного обучения является выбор представления

Нужно разобраться какими свойствами должен обладать граф для представления данных

# Свойства представлений

Ориентированность - ребра имеют направление



# Свойства представлений

Связность - наличие пути между всеми вершинами

