



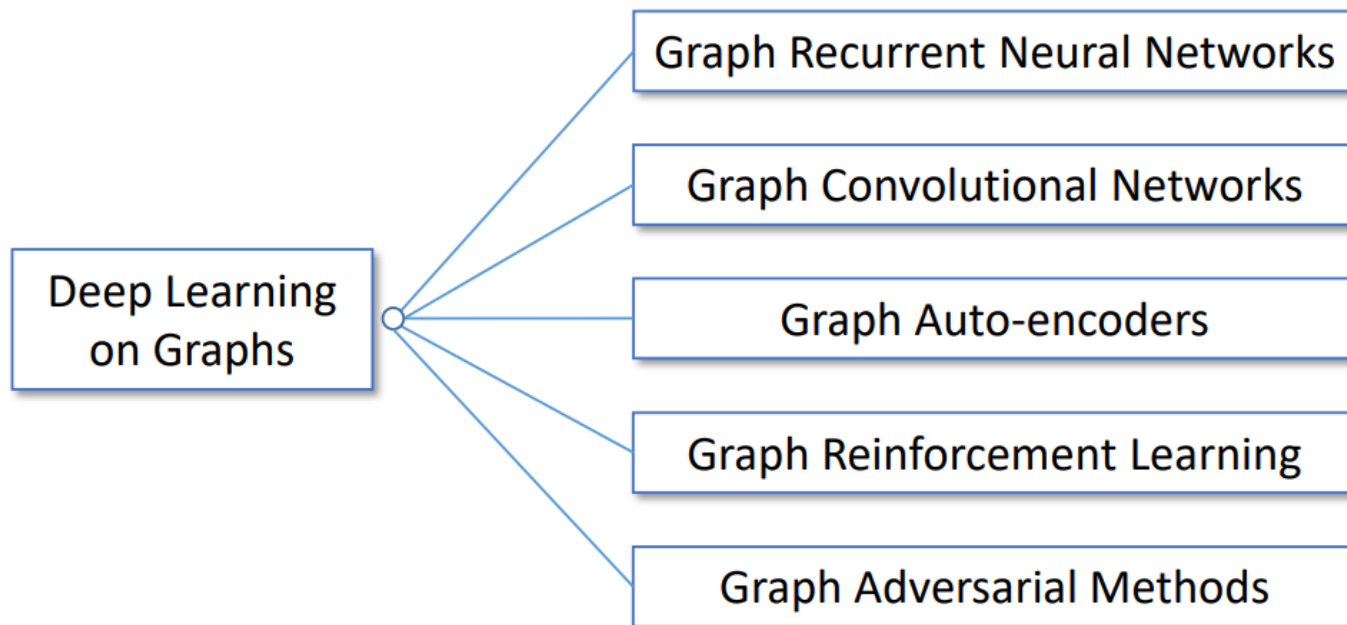
# DEEP-LEARNING НА ГРАФАХ

---

Вепренцев Иван

# ЧТО ТАКОЕ DEEP-LEARNING НА ГРАФАХ?

Применяем различные типы нейросетей для предсказания на графе (узлов, или всей структуры)



# ДВЕ ГЛАВНЫХ КАТЕГОРИИ GNN

Задачи на  
узлах

Классификация  
вершин

Предсказание  
связей

Рекомендация  
вершин

Задачи на  
графе

Классификация графов

Генерация графов

Category	Method	Recursive/sequential patterns of graphs	Scalability	Other Improvements
Node-level	GNN [23]	Recursive definition of node states	No	-
	NN4G [24]		No	-
	GGs-NNs [25]		Yes	Sequence output
	SSE [26]		Yes	-
Graph-level	You et al. [27]	Generate nodes and edges in an autoregressive manner	No	-
	DGNN [28]	Capture the time dynamics of the formation of nodes and edges	Yes	-
	RMGCNN [29]	Recursively reconstruct the graph	Yes	Convolutional layers
	Dynamic GCN [30]	Gather node representations in different time slices	Yes	Convolutional layers

# КЛАССИЧЕСКИЙ GNN НА УЗЛАХ

1) Каждая вершина графа представляется как вектор

$$s_i = \sum_{j \in \mathfrak{N}(i)} \mathfrak{F}(s_i, s_j, F_i^v, F_j^v, F_{i,j}^E)$$

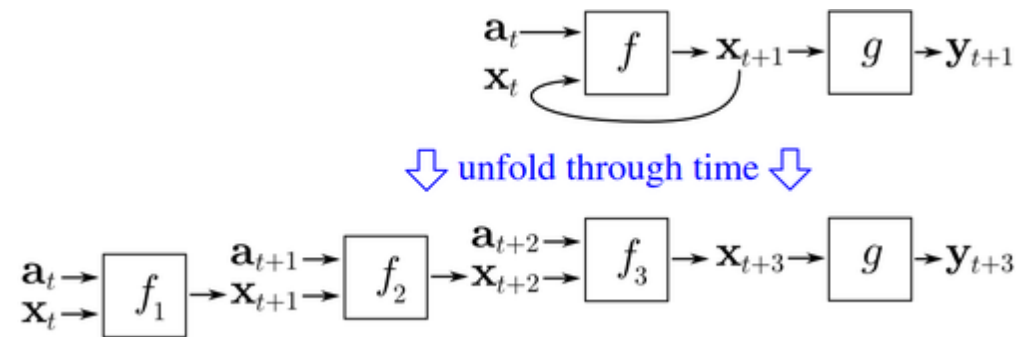
$\mathfrak{F}$  – параметрическая функция

$F_i^v$  - признаки вершины  $i$

$F_{i,j}^E$  - признаки ребра из  $i$  в  $j$

2) Выход считается как  $\hat{y}_i = \mathcal{O}(s_i, F_i^v)$

3) Обучение методом сквозного обратного распространения ошибки



# МОДИФИКАЦИИ И ПРИМЕНЕНИЕ GNN НА УЗЛАХ

## Gated Graph Sequence Neural Network

Использует управляемый рекуррентный блок в качестве модификации, тогда узлы

считаются как 
$$\mathbf{s}_i^{(t)} = (1 - \mathbf{z}_i^{(t)}) \odot \mathbf{s}_i^{(t-1)} + \mathbf{z}_i^{(t)} \odot \tilde{\mathbf{s}}_i^{(t)}$$

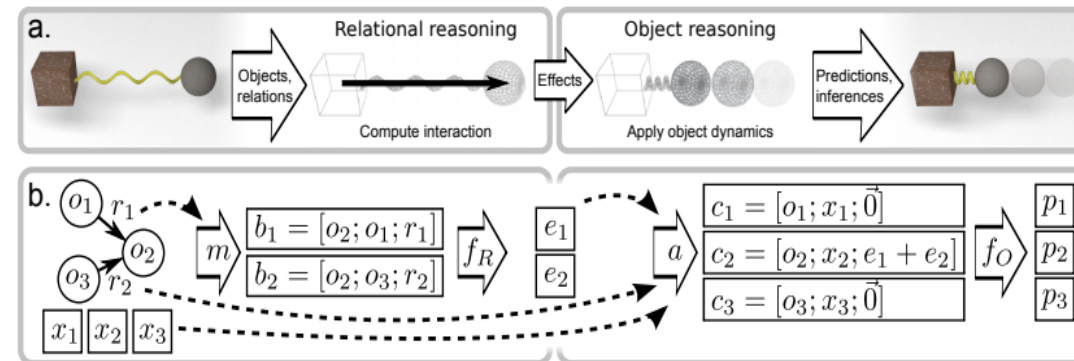
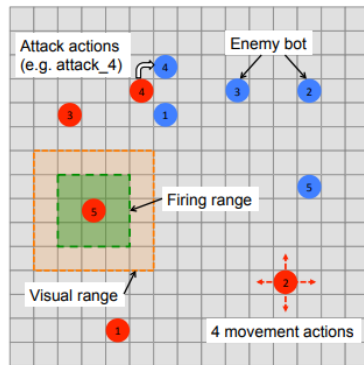
$\mathbf{z}$  - вычисляется через систему вентилей

$t$  - время

## SSE

Использует стохастический градиентный спуск вместе с GRU

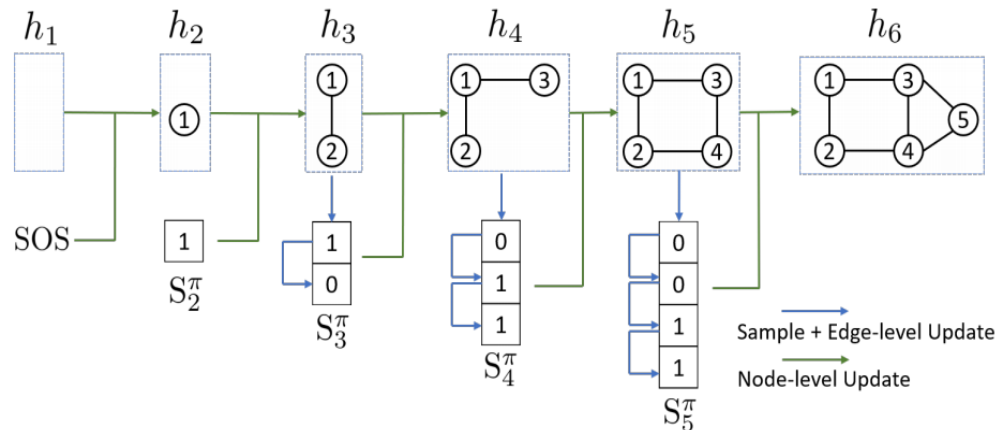
## Применения:



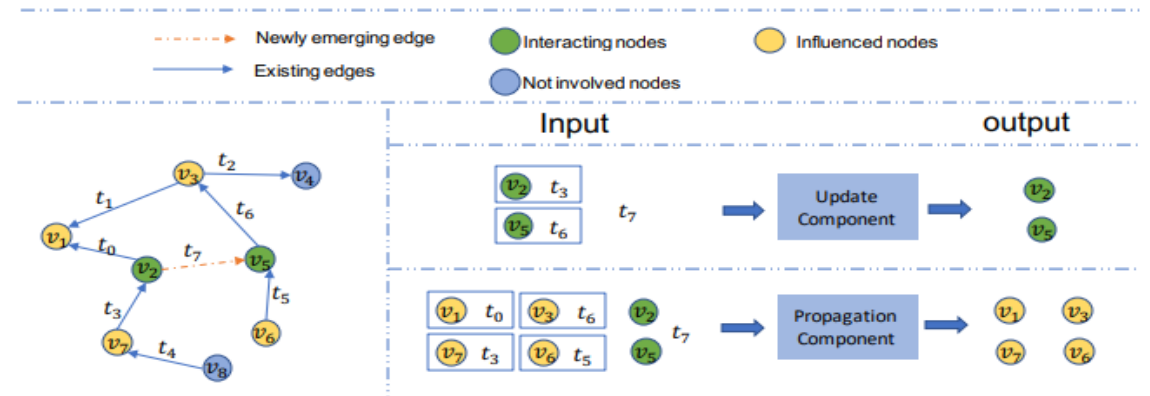
# GNN НА ГРАФАХ

Основное отличие – теперь все признаки находятся с помощью RNN, применяемой сразу ко всему графу, то есть вершины кодируются не по отдельности, а все вместе, а основная цель – предсказание на графе

## Примеры



You et. al 2 RNN



Dynamic Graph Neural Networks

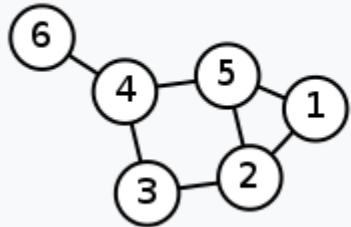
# СВЁРТОЧНЫЕ НЕЙРОСЕТИ НА ГРАФАХ (GCN)

Проблема: очевидно, что нельзя использовать свёрточные операции, обычно применяемые для изображений, из-за отсутствия сеточной структуры (для большинства графов)

Пусть  $h_v^t$  состояние вершины «v» в момент времени «t». Аналогично,  $e_{vw}^t$  — состояние ребра между вершинами «v» и «w» в момент времени «t». Тогда назовём сообщением  $m_v^{t+1} = (\sum h_w^t, \sum e_{vw}^t)$

Тогда определим **свёрточную операцию** как  $\mathbf{u}_1 *_G \mathbf{u}_2 = \mathbf{Q} \left( \left( \mathbf{Q}^T \mathbf{u}_1 \right) \odot \left( \mathbf{Q}^T \mathbf{u}_2 \right) \right)$

Где  $\mathbf{u}_1, \mathbf{u}_2$  — сообщения между соответствующими вершинами,  $\mathbf{Q}$  — собственные вектора матрицы лапласиана графа, которая вычисляется как  $\mathbf{L} = \mathbf{D} - \mathbf{A}$

Labelled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$



# СВЁРТОЧНЫЕ ОПЕРАЦИИ (ПРОДОЛЖЕНИЕ)

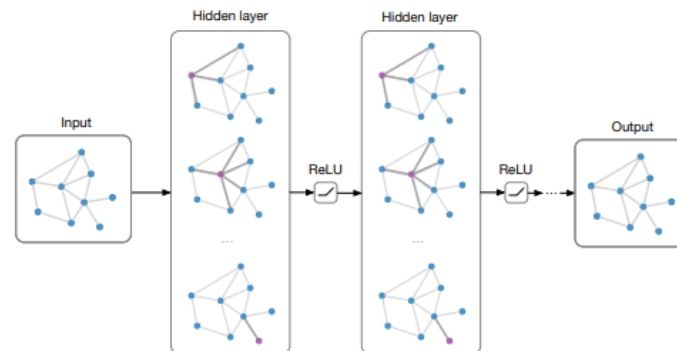
Определив свёрточные операции теперь введём свёрточный слой:

$$\mathbf{u}_j^{l+1} = \rho \left( \sum_{i=1}^{f_l} \mathbf{Q} \Theta_{i,j}^l \mathbf{Q}^T \mathbf{u}_i^l \right) \quad j = 1, \dots, f_{l+1},$$

$\mathbf{u}'$  - ВЫХОД

$\Theta = \Theta(\Lambda) \in \mathbb{R}^{N \times N}$  - матрица, полученная наложением фильтров на матрицу из собственных векторов матрицы Лапласа

Проблема: вычисление собственных векторов очень ресурсозатратно и требует  $O(N)^2$  операций. Недавно было предложено решение этой проблемы с использованием Чебышёвского разложения





# READOUT OPERATIONS

Мы вывели свёрточные слои. Каким образом теперь следует соединить выходы, чтобы решать задачу на графе целиком, а не на узлах?

подсчёт статистик  $\mathbf{h}_G = \text{mean/max/sum}(\mathbf{h}_1^{(K)}, \mathbf{h}_2^{(K)}, \dots, \mathbf{h}_n^{(K)})$ . (самый простой, но не эффективный способ: не учитывается структура и размер графа)

Set2Set алгоритм, создающий память и применяющий LSTM

«осмысленная» нумерация узлов и объединение их в бинарное дерево в зависимости от номера

DiffPool – учитывает и структуру графа, но гораздо более ресурсозатратен (для каждого слоя строится кластеровая матрица  $\mathbf{S}^{(k)} = \text{softmax}(\text{ConvGNN}_k(\mathbf{A}^{(k)}, \mathbf{H}^{(k)}))$  )

# ПРИМЕНЕНИЕ GCN

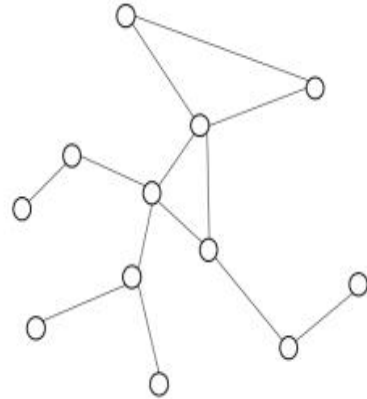
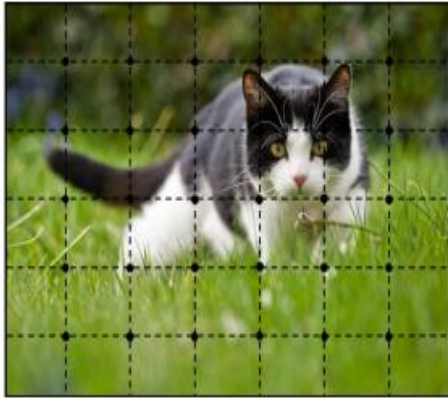
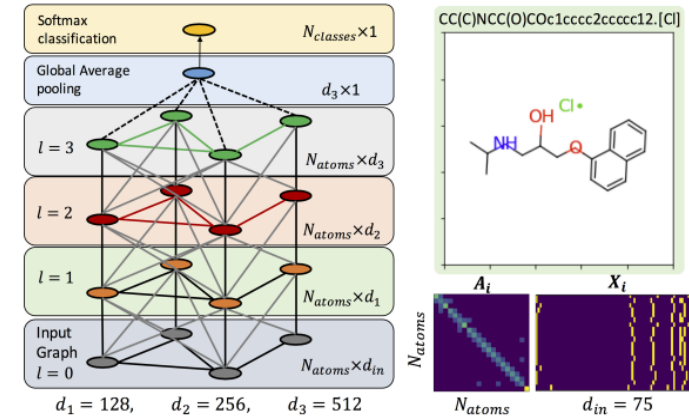
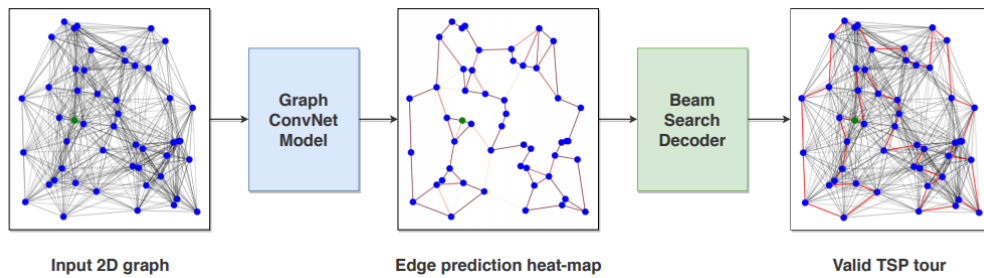


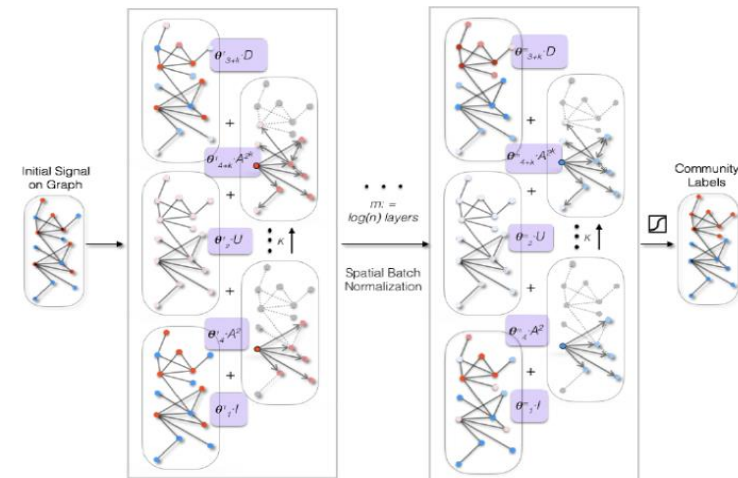
Image recognition



Molecular Structure Analysis



travelling salesman problem



Community Prediction