

## Методы построения графовых нейронных сетей

М. В. Мурзин<sup>1✉</sup>, И. А. Куликов<sup>2</sup>, Н. А. Жукова<sup>3</sup>

<sup>1</sup> Национальный исследовательский университет ИТМО, Санкт-Петербург, Россия

<sup>2</sup> Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

<sup>3</sup> Санкт-Петербургский Федеральный исследовательский центр Российской академии наук,  
СПИИРАН, Санкт-Петербург, Россия

✉ michael.murzin@metalab.ifmo.ru

**Аннотация.** Рассматривается один из способов классификации графовых нейронных сетей на основе базовых концепций, исследуются основы построения сверточных графовых нейронных сетей, графовых нейронных сетей внимания (Graph attentional neural network), рекуррентных графовых нейронных сетей, графовых автоматических кодировщиков, пространственно-временных графовых нейронных сетей. На примере набора данных «Cora» проведено сравнение графовых нейросетевых моделей, представленных в библиотеках TensorFlow, PyTorch, а также модель графовой нейронной сети внимания для задачи классификации узлов графа знаний. Показана эффективность использования графовых сетей внимания для классификации узлов графа.

**Ключевые слова:** графы знаний, графовые нейронные сети, классификация узлов графа

**Для цитирования:** Мурзин М. В., Куликов И. А., Жукова Н. А. Методы построения графовых нейронных сетей // Изв. СПбГЭТУ «ЛЭТИ». 2024. Т. 17, № 10. С. 40–48. doi: 10.32603/2071-8985-2024-17-10-40-48.

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

Review article

## Methods for Constructing Graph Neural Networks

M. V. Murzin<sup>1✉</sup>, I. A. Kulikov<sup>2</sup>, N. A. Zhukova<sup>3</sup>

<sup>1</sup> ITMO University, Saint Petersburg, Russia

<sup>2</sup> Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

<sup>3</sup> SPIIRAS of St. Petersburg Federal Research Center of the Russian Academy of Sciences,  
Saint Petersburg, Russia

✉ michael.murzin@metalab.ifmo.ru

**Abstract.** Discusses an approach to classifying graph neural networks in terms of basic concepts. In addition, the fundamentals of convolutional graph neural networks, Graph attentional neural networks, recurrent graph neural networks, graph automatic encoders, and spatial-temporal graph neural networks are discussed. On the example of Cora dataset, a comparison of neural network models presented in TensorFlow, PyTorch libraries, as well as the model of graph neural network of attention for the task of classification of nodes of the knowledge graph, is carried out. The efficiency of using graph attention networks to solve the problem of graph node classification is shown.

**Keywords:** knowledge graphs, graph neural networks, graph node classification

**For citation:** Murzin M. V., Kulikov I. A., Zhukova N. A. Methods for Constructing Graph Neural Networks // LETI Transactions on Electrical Engineering & Computer Science. 2024. Vol. 17, no. 10. P. 40–48. doi: 10.32603/2071-8985-2024-17-10-40-48.

**Conflict of interest.** The authors declare no conflicts of interest.

**Введение.** Построение графовых нейросетей и их использование – мощный метод дедуктивно-го анализа графов знаний. Дедуктивные методы позволяют на основе априорных сведений и общих правил дополнить граф новыми данными. На сегодняшний день сфера применения графовых нейросетевых моделей обширна, на их основе удается построить инструменты предиктивной аналитики, вопросно-ответные системы. Кроме того, удается успешно интегрировать решения на основе графовых нейросетей, например в медицинские системы, системы интернета вещей для удовлетворения потребностей пользователей. Многие научные работы предлагают новые методы для повышения точности нейросетевых моделей, в связи с чем количество моделей стремительно возрастает и становится сложно выявить принципиальные отличия методов для построения эффективной модели.

Графы знаний – это гибкая структура представления данных, и за счет этого они находят широкое применение в различных предметных областях. Так, например, графы знаний применяются в биологических системах для представления связей между молекулами. Одним из методов дедуктивного анализа данных в графах знаний являются графовые нейронные сети – их использование, например позволяет выявить совместимость лекарств, что значительно упрощает рекомендации для пациентов. Кроме того, выстраивание цепи посещения веб-сайтов пользователями позволяет настроить рекламу на сайте для повышения продаж смежных категорий товаров на основе пользовательских запросов за счет использования графовых нейронных сетей.

Актуальность данного исследования связана с возрастающим в течение последнего десятилетия интереса к графовым нейросетям, определяемого высокой точностью моделей при разработке систем принятия решений, вопросно-ответных систем [1], в инструментах предиктивной аналитики, а также рекомендательных системах [2]. При этом можно выделить определенные классы задач, решаемых с помощью графовых нейронных сетей: предсказание отношений в графе, классификация вершин графа, классификация самих графов, а также предсказание эволюции графа на основании данных о предыдущих состояниях графа. Развитие графовых нейросетей повлекло за собой возникновение принципиально новых нейросетевых моделей. На сегодняшний день недостаточно проработан обоснованный выбор ин-

струментария при решении конкретного класса задач для достижения максимальной точности нейросетевых моделей, поскольку от выбора архитектуры графовой нейросети зависит точность итоговой модели и, как следствие, материально-технические показатели системы, которая содержит графовую нейронную сеть.

В рамках данной статьи проведен анализ различных архитектур графовых нейронных сетей на основе базовых принципов их построения. Кроме того, были построены различные модели графовых нейронных сетей, проведено сравнение точности моделей для решения задачи классификации вершин графа.

**Постановка задач.** Графовые нейронные сети успешно применяются для решения задач предсказания связей между вершинами графа, классификации графов, а также для классификации вершин графа. Особый интерес представляет задача классификации вершин графа, поскольку ее решение имеет перспективы для применения в пользовательских, а также промышленных системах интернета вещей [3]. Для решения рассматриваемой задачи с различной эффективностью применяются разнообразные архитектуры графовых нейронных сетей: рекуррентные, сверточные, графовые нейросети внимания. Актуальным остается выбор архитектуры графовых нейронных сетей для достижения наибольшей точности [4], [5] при решении задачи классификации вершин графа. Исходя из существующей проблемы, данная статья нацелена на анализ точности различных структур графовых нейронных сетей для решения задачи классификации вершин графа. Объектом исследования служат архитектуры графовых нейронных сетей.

**Архитектуры графовых нейронных сетей.** Авторами [6] был предложен подход к классификации графовых нейронных сетей. Согласно данной классификации, можно выделить следующие категории графовых нейронных сетей:

- 1) рекуррентные графовые нейронные сети;
- 2) сверточные графовые нейронные сети;
- 3) графовые автоматические кодировщики;
- 4) пространственно-временные графовые нейронные сети.

Кроме того, в [7] указывается нейросетевой метод графовых сетей внимания, обладающий высокой точностью при решении задач классификации узлов графа. Рассмотрим подробнее каждый метод.

Рекуррентные графовые нейросети предназначены для описания систем, удовлетворяющих условиям Марковского процесса. Процесс называется Марковским, если выполняется условие

$$P(x_{t+1} | x_{1:t}) = P(x_{t+1} | x_t), \quad (1)$$

где  $P(x_{t+1} | x_{1:t})$  – вероятность события в момент времени  $t + 1$  при условии, что известны предыдущие события от начала до момента времени  $t$ ;  $P(x_{t+1} | x_t)$  – вероятность события в момент времени  $t + 1$  при условии, что известно состояние системы в момент времени  $t$ .

При этом, если процесс не удовлетворяет условию (1), то он может быть рассмотрен в качестве скрытой Марковской модели. Стохастический процесс есть скрытая Марковская модель, если существует такой процесс  $z_t$ , для которого выполняется

$$P(z_{t+1} | z_{1:t}) = P(z_{t+1} | z_t) \text{ и } P(x_t | z_t) = P(x_t | z_{0:t}).$$

Скрытое состояние  $z_t$  – это Марковский процесс. Состояние  $x_t$  при этом называется условно независимым, поскольку зависит только от текущего состояния  $z_t$ . Результат предсказания траектории  $y_t$  также условно независим, т. е. зависит только от состояния скрытого состояния  $z_t$ .

Скрытое состояние  $z_{t+1}$  определяется состоянием  $z_t$ , плотность распределения вероятности  $y_t$  полностью определяется состоянием  $z_t$ . Однако состояния  $z_t$  не относятся к наблюдаемым.

Рекуррентные нейронные сети получают информацию о состояниях  $z_t$  через наблюдаемые состояния  $x_t$ ; далее, зная состояние  $z_t$ , можно полностью определить состояние  $y_t$ . На рис. 1 представлена диаграмма состояний рекуррентных нейронных сетей. На нем введены два параметра:  $\phi_1$  и  $\phi_2$ . Параметр  $\phi_1$  есть отображение множеств  $z_{t-1}$  и  $x_t$  на множество  $z_t$ , параметр  $\phi_2$  – отображение  $z_t$  на  $\hat{y}_t$ . Нейросеть называется рекуррентной,

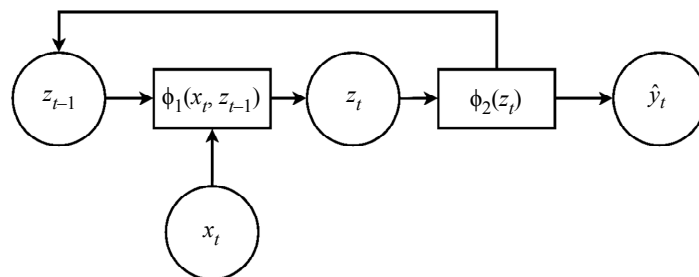


Рис. 1. Диаграмма состояний рекуррентной нейронной сети  
Fig. 1. State diagram of recurrent neural network

поскольку при рассмотрении нового наблюдаемого состояния  $z_t$  переходит в  $z_{t-1}$  и снова поступает на вход  $\phi_1$ . Параметры  $\phi_1$  могут быть представлены узлами полносвязной нейронной сети, значение которых определяется линейной комбинацией  $z_{t-1}$  и  $x_t$ . Параметры  $\phi_2$  также могут быть представлены перцептронами, состояние которых полностью определяется  $z_t$ .

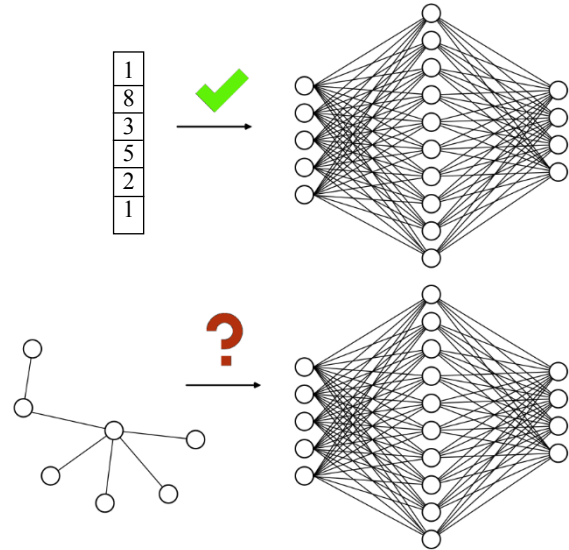


Рис. 2. Структурированное представление графа знаний  
Fig. 2. Structed knowledge graph representation

Сверточные нейронные сети успешно применяются для решения задач классификации и кластеризации изображений. Сами изображения можно представить как трехмерную матрицу, размеры которой определяются размерами изображения, а также тремя каналами цветов: красным, зеленым и синим, с помощью значений которых можно закодировать любой цвет. Изображение можно легко представить в виде графа, отобразив массив пикселей в узлы ненаправленного графа, связи в котором присутствуют между соседними пикселями. Нейронные сети оперируют данными предопределенного размера, в случае с изображениями не возникает сложностей

при составлении такой структуры. Графы знаний представляют собой гибкую структуру хранения данных и не гарантируют жесткой привязки к количеству узлов, связей (иллюстрация проблемы представлена на рис. 2), поэтому особенно значима предобработка данных перед построением нейросетевых моделей, определение узлов, для которых требуется построение модели, строгое определение количества свойств для каждого узла.

Сверточные графовые нейронные сети – это многослойные нейронные сети, которые обрабатывают вершины графа и связи между ними, находящиеся в векторном представлении [7]. Введем обозначения:  $G = (V, E)$ , где  $V$  и  $E$  – вершины и ребра графа. Количество вершин графа обозначим  $n$ . Для каждой вершины считаем, что она имеет соединение сама с собой. Пусть  $\mathbf{X} \in R^{n \times m}$  – матрица, состоящая из вершин графа и свойств вершин, где  $m$  – размерность вектора свойств; каждая строка  $\mathbf{x}_v$  – вектор свойств для вершины  $v$ . Пусть  $\mathbf{A}$  – матрица смежности графа знаний,  $\mathbf{D}$  – матрица степеней, где  $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ ,  $i$  –

узел графа. Диагональные элементы матрицы смежности равны 1, поскольку было обозначено, что каждая вершина имеет соединение сама с собой. С помощью одного слоя свертки можно получить информацию лишь о ближайших вершинах графа, для извлечения информации о большем числе узлов необходимо использовать большее число сверточных слоев. Для однослойной нейронной сети новая  $k$ -мерная матрица признаков узлов может быть вычислена следующим образом:

$$\mathbf{L}^{(1)} = f(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}_0),$$

где  $\tilde{\mathbf{A}} = D^{-1/2}\mathbf{A}D^{1/2}$  – нормализованная матрица смежности;  $\mathbf{W}_0 \in R^{m \times k}$  – матрица весов;  $f$  – функция активации.

Как было отмечено ранее, для сбора информации о большем количестве узлов необходимо использовать большее количество слоев свертки. Матрица признаков в таком случае может быть определена следующим образом:

$$\mathbf{L}^{(j+1)} = f(\tilde{\mathbf{A}}\mathbf{L}^{(j)}\mathbf{W}_j),$$

где  $j$  – количество слоев;  $\mathbf{L}^{(0)} = \mathbf{X}$ .

Успешное применение сверточных нейронных сетей на основе графов в задачах классификации изображений, а также в задачах обработки естественного языка обусловлено способностью

сверточных нейронных сетей использовать свойства стационарности и композиционности данных [8]. Однако неевклидовы характеристики графов делают свертку не таким удобным инструментом по сравнению с ее применением для анализа изображений. Основными направлениями развития в решении данной проблемы служат переход к построению графов по спектральному представлению данных по аналогии с одномерным преобразованием Фурье, а также использование вейвлет-преобразования.

Для проведения преобразования Фурье над графом необходимо определить для графа оператор Лапласа. Авторами [8] отмечается, что в качестве оператора Лапласа можно использовать матрицу случайного хождения по графу. Собственное значение оператора Лапласа в таком случае можно представить как:

$$\tilde{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T,$$

где  $\mathbf{U}$  – матрица, столбцы которой  $u_l$  – это собственные векторы оператора Лапласа;  $\mathbf{\Lambda}$  – диагональная матрица, состоящая из собственных значений  $\lambda_l$  оператора Лапласа;  $l$  – порядковый номер элемента главной диагонали матрицы  $\mathbf{\Lambda}$ .

Результат преобразования Фурье можно записать как

$$\hat{x}(\lambda_l) = \langle x, u_l \rangle = \sum_{i=1}^n x(i)u_l^*(i),$$

где  $u_l^*(i)$  – комплексное сопряжение  $i$ -го элемента собственного вектора оператора Лапласа.

Для проведения обратного преобразования Фурье можно воспользоваться выражением

$$x_i = \sum_{l=1}^n \hat{x}(\lambda_l)u_l(i).$$

Использование преобразования Фурье над графом расширяет область применения сверточных нейронных сетей на основе графов.

Кластеризация представляет собой задачу машинного обучения без учителя, в которой изначально неизвестно, на какое количество классов необходимо разделить обучающую выборку. Графовые автоматические кодировщики могут применяться как совместно со сверточными нейронными сетями для решения задач кластеризации [9], так и отдельно от них при замене сверточной нейронной сети на линейную модель, представленную в виде матрицы смежности [10]. Графовые автоматические кодировщики также успешно применяются для задач эмбединга, метода снижения размерно-

сти векторов узлов и отношений графа. За счет использования графовых автоматических кодировщиков оказывается возможным восстановить (декодировать) структуру исходного графа уже после проведения процесса эмбединга.

Графовые автоматические кодировщики – это семейство моделей, нацеленных на создание отображения каждого узла графа  $i \in V$  в вектор  $\mathbf{z}_i \in R^d$  при условии  $d \ll n$ , из которого должна быть возможность восстановить с определенной степенью точности матрицу смежности [10]. Если такие условия выполняются, то векторы  $\mathbf{z}_i$  должны содержать в себе важные особенности структуры исходного графа. Матрица  $\mathbf{Z}$  размером  $n \times m$  для всех векторов  $\mathbf{z}_i$  есть результат работы графовой нейронной сети, обрабатывающей матрицу смежности  $\mathbf{A}$ . Для восстановления исходного графа к графовой нейронной сети добавляется декодер внутреннего произведения [11]. Восстановленная матрица смежности может быть записана следующим образом:

$$\hat{\mathbf{A}}_{ij} = \sigma(\mathbf{z}_i^T \mathbf{z}_j),$$

где  $\sigma = \frac{1}{1 + e^{-x}}$  – функция-сигмоид.

Чем больше произведение  $\mathbf{z}_i^T \mathbf{z}_j$ , тем ближе друг к другу находятся узлы  $i$  и  $j$  в графе. Матрица  $\mathbf{Z}$  находится с помощью графовой нейронной сети при подаче на вход матрицы смежности  $\mathbf{A}$ . Веса графовой нейронной сети находятся методом градиентного спуска для минимизации потерь при восстановлении исходной матрицы смежности (взвешенные перекрестные потери энтропии).

Расширением функциональности графовых автоматических кодировщиков служат вариативные графовые автоматические кодировщики [11]. Авторам удалось разработать вероятностную модель, включающую скрытые слои векторов  $\mathbf{z}_i$  длиной  $d \ll n$  для каждого узла  $i \in V$ , интерпретируемых как представления узлов в пространстве эмбединга. Кодировочная часть вариативных автоматических графовых кодировщиков может быть определена как

$$q(\mathbf{Z} | \mathbf{A}) = \prod_{i=1}^n q(\mathbf{z}_i | \mathbf{A}),$$

где  $q(\mathbf{z}_i | \mathbf{A}) = N[\mathbf{z}_i | \mu_i, \text{diag}(\sigma_i^2)]$ .

Параметры гауссиана получаются на выходе графовой нейронной сети  $\mu_i = GNN_{\mu}(\mathbf{A})$ , где  $\mu$  – матрица, содержащая суммы векторов  $\mu_i$ . Скрытые векторы  $\mathbf{z}_i$  – выборки данного распределения. Из этих векторов генеративная модель стремится восстановить матрицу смежности, вычисляя внутренние произведения:

$$q(\mathbf{A} | \mathbf{Z}) = \prod_{i=1}^n \prod_{j=1}^n P(\mathbf{A}_{ij} | \mathbf{z}_i, \mathbf{z}_j),$$

где  $P(\mathbf{A}_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^T \mathbf{z}_j)$ .

Во время обучения веса графовой нейронной сети подбираются для максимизации вариационной нижней границы правдоподобия [10] с использованием метода градиентного спуска, а также преимущественно используя распределение Гаусса при установлении распределения скрытых векторов. Автоматические графовые кодировщики успешно применяются для задач кластеризации, задач генерации молекул, в рекомендательных системах, а также в системах предсказания отношений между узлами графа.

Рассмотренные ранее сверточные, рекуррентные нейронные сети и автоматические кодировщики успешно применяются для решения большинства нейросетевых задач. Однако существует ряд задач, для которых применение данных методов неоптимально. Одна из них – это задача предсказания трафика, для решения которой необходимо применение пространственно-временных графовых нейросетей [12].

Введем некоторые определения. Пусть  $\{x_i^1, \dots, x_i^T\}$  – временной ряд в точке пространства  $i$ , в момент времени  $T$ . Набор данных трафика в определенной области – это набор из  $N$  временных рядов  $\mathbf{X} = \{X_1, \dots, X_N\} \in R^{N \times T}$ . Задача предсказания трафика заключается в предсказании следующих временных рядов на основании данных о предыдущих  $S$  временных рядах. Формально задачу можно поставить следующим образом [12]:

$$\Theta^* = \arg \min \sum_{i=1}^T L_0 [X^{(t-S):t}, X^{(t+1):(t+T)}],$$

где  $X^{(t-S):t}$  и  $X^{(t+1):(t+T)}$  – входное и выходное значения нейронной сети соответственно;  $\Theta^*$  – оптимальный параметр, определяющий веса модели для нахождения нелинейной функции потерь  $L_0$ .

В последние годы пространственно-временные графовые нейронные сети показали высокую эффективность в обучении неевклидовых структур для предсказания дорожного трафика, которые состоят из статичного графа и динамично меняющихся входных данных. Пусть  $G = (V, E, A)$  – граф, наполненный пространственно-временными данными дорожного трафика, задачу нейросетевого обучения можно представить следующим образом:

$$\left[ X^{(t-S):t}, G \right] \xrightarrow{f} X^{(t+1):(t+T)},$$

где  $f$  – функция, которая получается в результате обучения.

Большинство пространственно-временных моделей нейросетей основанных на определенном графе для применения процесса распространения не могут быть напрямую расширены для применения многомерных задач. Авторами [13] была разработана модель MTGNN для обучения не предопределенных графовых структур. Обучаемый слой графов был разработан для изучения смежности графов, которая является входом для модулей свертки. Последовательно адаптивное обучение смежности графов можно сформулировать следующим образом:

$$\tilde{A} = \text{ReLU} \left\{ \text{tg} \left[ \omega \left( \mathbf{E}_1 \mathbf{E}_2^T - \mathbf{E}_2 \mathbf{E}_1^T \right) \right] \right\},$$

где  $\mathbf{E}_1 = \text{tg}(\omega E_1^0 W_1)$ ,  $\mathbf{E}_2 = \text{tg}(\omega E_2^0 W_2)$  – случайно проинициализированные эмбединги узлов с параметром модели  $W$ ;  $\omega$  – гиперпараметр для контроля насыщения функции активации.

В [12] было изучено применение модели MTGNN для решения задачи предсказания автомобильного трафика. Точность модели оказалась более 90 %, что говорит о возможности успешного применения пространственно-временных нейронных сетей для обучения неевклидовых структур.

Рассмотрим архитектуру графовых сетей внимания, представленную в [7]. Особенность данной архитектуры заключается в сложении слоев, в которых узлы могут наблюдать характеристики своих соседей. За счет данного подхода удается избежать трудозатратных матричных операций (например, инверсии) за счет априорного знания структуры графа.

**Методика исследования.** Для исследования задачи классификации вершин графа рассмотрим открытый набор данных Cora, содержащий сведения о цитировании научных публикаций. Выбранный набор данных характеризуется полнотой, валидностью и уникальностью. Для набора

данных Cora построены модели графовых нейронных сетей внимания, а также сверточных графовых нейросетей, представленных в библиотеках Tensorflow и PyTorch. Для построения нейросетевых моделей использовались вычислительные ресурсы Google Colab. Оценена точность нейросетевых моделей для решения задач классификации вершин графа, в качестве критерия использовалась доля верно определенных тематик научных статей. На основании точности моделей сделаны выводы об оптимальном выборе конфигурации нейросетевых моделей для решения задачи классификации вершин графа.

Набор данных Cora содержит уникальные идентификаторы научных публикаций, из которых производится цитирование, идентификаторы научных публикаций [10], которые цитировали, а также в формате one-hot кодирования представлены цитаты, которые были процитированы. В ходе работы требуется разделение научных публикаций на области науки, к которым они относятся.

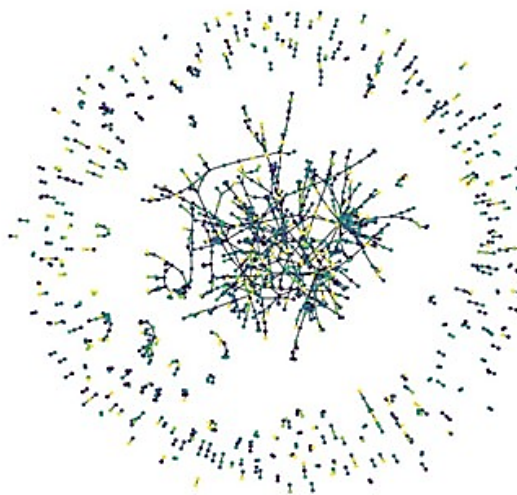


Рис. 3. Граф на основе данных Cora  
Fig. 3. Graph based on Cora database

Существует возможность построения графа знаний из представленного набора данных средствами языка Python. Так, с использованием библиотеки matplotlib можно получить изображение графа знаний (рис. 3), каждый узел которого отражает научную публикацию. Для более удобного восприятия можно обозначить различными цветами предметные области, к которым относится научная публикация. Кроме того, такое представление помогает оценить расстояния между различными узлами графа в векторном представлении. Данный граф знаний – разреженный, что характерно для графов, основанных на реальных данных.



Полная реализация нейросетевых моделей представлена на языке Python в репозитории [14]. При реализации моделей использовалось разделение исходного набора данных на тренировочную и тестовую выборки в отношении 50/50. В нейросетевой модели PyTorch использовалась модель оптимизации Adam со скоростью обучения 0.01, в нейросетевой модели GAT – оптимизатор SGD со скоростью обучения 0.3.

**Результаты исследований.** Сравнение точностей нейросетевых моделей для предсказания тематики научных статей в зависимости от используемых слов в цитировании представлено на рис. 4.

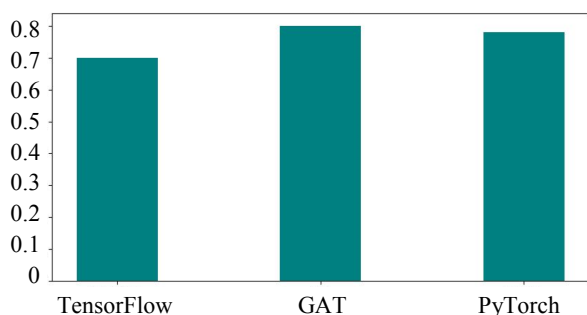


Рис. 4. Сравнение точности нейросетевых моделей  
Fig. 4. Comparison of accuracy of neural network models

При решении задачи классификации узлов графа знаний наибольшей точностью обладает модель графовой сети внимания (GAT); модели сверточных нейросетей, представленные в библиотеках TensorFlow и PyTorch, обладают точностью выше 70 %.

**Обсуждение.** В результате исследования точности различных нейросетевых моделей для задачи классификации вершин графа большей точ-

ностью обладает графовая сеть внимания, поскольку данная архитектура позволяет повышать вес от соседних вершин графа, учитывает близость векторов соседних вершин графа в векторном пространстве. Кроме того, более высокая точность графовых нейросетей внимания может быть обусловлена разреженностью набора данных Coqa. Также графовая нейронная сеть внимания по сравнению со сверточной нейронной сетью может лучше справляться с проблемой затухания градиентов и в результате иметь более высокую точность. Представленные результаты имеют прикладной характер и могут применяться при построении графовых моделей с учетом специфики решаемой задачи.

**Заключение.** В данной статье проведен анализ различных архитектур графовых нейронных сетей. Кроме того, в среде разработки Google Colab реализованы модели сверточных графовых нейронных сетей, представленных в библиотеках TensorFlow и PyTorch, а также модель графовой сети внимания для решения задачи классификации вершин графа на примере открытого набора данных Coqa. Полученные модели с точностью выше 70 % классифицируют научные статьи по темам, что представляет хороший результат для решения задачи многоклассовой классификации. Для задачи классификации вершин графа более высокую точность показала графовая сеть внимания. В дальнейшем планируется применить полученные результаты при решении задачи классификации вершин графа для системы интернета вещей.

#### Список литературы

1. Jiang Z., Chi C., Zhan Y. Research on medical question answering system based on knowledge graph // IEEE Access. 2021. Vol. 9. P. 21094–21101. doi: 10.1109/ACCESS.2021.3055371.
2. Graph neural networks in recommender systems: A survey / Shi. Wu, F. Sun, W. Zhang, X. Xie, B. Cui // ACM Computing Surveys. 2022. Vol. 55, iss. 5. P. 1–37. doi: 10.1145/3535101.
3. Graph neural networks in IoT: A survey / G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbell, L. E. Barnes, M. Boukhechba. URL: <https://doi.org/10.48550/arXiv.2203.15935> (дата обращения: 30.05.2024).
4. Vignac C., Ortiz-Jiménez G., Frossard P. On the choice of graph neural network architectures // IEEE Intern. Conf. on Acoustics, Speech and Signal Proc. (ICASSP 2020). Barcelona, Spain: IEEE, 2020. P. 8489–8493. doi: 10.1109/ICASSP40776.2020.9054357.
5. Benchmarking graph neural networks / V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, X. Bres-

- son. URL: <https://doi.org/10.48550/arXiv.2003.00982> (дата обращения: 30.05.2024).

6. A Comprehensive survey on graph neural networks / Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu // IEEE Trans. on Neural Networks and Learning Systems. 2021. Vol. 32, iss. 1. P. 4–24. doi: 10.1109/TNNLS.2020.2978386.

7. Graph attention networks / P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio. URL: <https://doi.org/10.48550/arXiv.1710.10903> (дата обращения: 30.05.2024).

8. Graph convolutional networks: a comprehensive review / Si Zhang, H. Tong, J. Xu, R. Maciejewski // Computational Social Networks. 2019. Vol. 6, iss. 1. Art. 11. 23 p. doi: 10.1186/s40649-019-0069-y.

9. Li X., Zhang H., Zhang R. Adaptive graph auto-encoder for general data clustering // IEEE Trans. on Pattern Analysis and Machine Intelligence. 2022. Vol. 44, iss. 12. P. 9725–9732. doi: 10.1109/TPAMI.2021.3125687.

10. Salha G., Hennequin R., Vazirgiannis M. Keep it simple: Graph autoencoders without graph convolutional networks. URL: <https://doi.org/10.48550/arXiv.1910.00942> (дата обращения: 30.05.2024).

11. Kipf T. N., Welling M. Variational Graph Auto-Encoders URL: <https://doi.org/10.48550/arXiv.1611.07308> (дата обращения: 30.05.2024).

12. Bui K H. N., Cho J., Yi H. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues // Appl. Intell. 2022. Vol. 52(4). P. 2763–2774. doi: 10.1007/s10489-021-02587-w.

13. Connecting the dots: Multivariate time series forecasting with graph neural networks / Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang // Proc. of the 26<sup>th</sup> ACM SIGKDD Intern. Conf. on knowledge discovery & data mining (KDD'20). New York, NY, USA: ACM, 2020. P. 753–763. doi: 10.1145/3394486.3403118.

14. Graph neural network models. URL: [https://github.com/murzin66/Graph\\_neural\\_network\\_models](https://github.com/murzin66/Graph_neural_network_models) (дата обращения: 30.05.2024).

### Информация об авторах

**Мурзин Михаил Владимирович** – магистрант, факультет программной инженерии и компьютерной техники Национального исследовательского университета ИТМО, Кронверкский пр., д. 49, лит. А, Санкт-Петербург, 197101, Россия.

E-mail: [michael.murzin@metalab.ifmo.ru](mailto:michael.murzin@metalab.ifmo.ru)

<https://orcid.org/0009-0001-0040-2195>

**Куликов Игорь Александрович** – соискатель кафедры математического обеспечения и применения ЭВМ (МОЭВМ) СПбГЭТУ «ЛЭТИ».

E-mail: [i.a.kulikov@gmail.com](mailto:i.a.kulikov@gmail.com)

<https://orcid.org/0000-0002-2532-5579>

**Жукова Наталия Александровна** – д-р техн. наук, доцент, вед. научный сотрудник СПИИРАН СПб ФИЦ РАН, 14-я линия ВО, 39, Санкт-Петербург, 199178, Россия.

E-mail: [nazhukova@mail.ru](mailto:nazhukova@mail.ru)

<https://orcid.org/0000-0001-5877-4461>

### References

1. Jiang Z., Chi C., Zhan Y. Research on medical question answering system based on knowledge graph // IEEE Access. 2021. Vol. 9. P. 21094–21101. doi: 10.1109/ACCESS.2021.3055371.

2. Graph neural networks in recommender systems: A survey / Shi. Wu, F. Sun, W. Zhang, X. Xie, B. Cui // ACM Computing Surveys. 2022. Vol. 55, iss. 5. P. 1–37. doi: 10.1145/3535101.

3. Graph neural networks in IoT: A survey / G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbell, L. E. Barnes, M. Boukhechba. URL: <https://doi.org/10.48550/arXiv.2203.15935> (дата обращения: 30.05.2024).

4. Vignac C., Ortiz-Jiménez G., Frossard P. On the choice of graph neural network architectures // IEEE Intern. Conf. on Acoustics, Speech and Signal Proc. (ICASSP 2020). Barcelona, Spain: IEEE, 2020. P. 8489–8493. doi: 10.1109/ICASSP40776.2020.9054357.

5. Benchmarking graph neural networks / V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, X. Bresson. URL: <https://doi.org/10.48550/arXiv.2003.00982> (дата обращения: 30.05.2024).

6. A Comprehensive survey on graph neural networks / Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu // IEEE Trans. on Neural Networks and Learning Systems. 2021. Vol. 32, iss. 1. P. 4–24. doi: 10.1109/TNNLS.2020.2978386.

7. Graph attention networks / P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio. URL: <https://doi.org/10.48550/arXiv.1710.10903> (дата обращения: 30.05.2024).

8. Graph convolutional networks: a comprehensive review / Si Zhang, H. Tong, J. Xu, R. Maciejewski // Computational Social Networks. 2019. Vol. 6, iss. 1. Art. 11. 23 p. doi: 10.1186/s40649-019-0069-y.

9. Li X., Zhang H., Zhang R. Adaptive graph auto-encoder for general data clustering // IEEE Trans. on Pattern Analysis and Machine Intelligence. 2022. Vol. 44, iss. 12. P. 9725–9732. doi: 10.1109/TPAMI.2021.3125687.

10. Salha G., Hennequin R., Vazirgiannis M. Keep it simple: Graph autoencoders without graph convolutional networks. URL: <https://doi.org/10.48550/arXiv.1910.00942> (дата обращения: 30.05.2024).

11. Kipf T. N., Welling M. Variational Graph Auto-Encoders URL: <https://doi.org/10.48550/arXiv.1611.07308> (дата обращения: 30.05.2024).

12. Bui K H. N., Cho J., Yi H. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues // Appl. Intell. 2022. Vol. 52(4). P. 2763–2774. doi: 10.1007/s10489-021-02587-w.

13. Connecting the dots: Multivariate time series forecasting with graph neural networks / Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang // Proc. of the 26<sup>th</sup>



ACM SIGKDD Intern. Conf. on knowledge discovery & data mining (KDD'20). New York, NY, USA: ACM, 2020. P. 753–763. doi: 10.1145/3394486.3403118.

14. Graph neural network models. URL: [https://github.com/murzin66/Graph\\_neural\\_network\\_models](https://github.com/murzin66/Graph_neural_network_models) (data obrashhenija: 30.05.2024).

---

Information about the authors

**Mikhail V. Murzin** – master student, Faculty of Software Engineering and Computer Technology, ITMO University, Kronverkskiy av., 49, lit. A, St. Petersburg, 197101, Russia.

E-mail: [michael.murzin@metalab.ifmo.ru](mailto:michael.murzin@metalab.ifmo.ru)

<https://orcid.org/0009-0001-0040-2195>

**Igor A. Kulikov** – Candidate of the Department of Mathematical Support and Computer Applications, Saint Petersburg Electrotechnical University.

E-mail: [i.a.kulikov@gmail.com](mailto:i.a.kulikov@gmail.com)

<https://orcid.org/0000-0002-2532-5579>

**Natalia A. Zhukova** – Dr Sci. (Eng.), Associate Professor, leading researcher, SPIIRAS of St. Petersburg Federal Research Center of the Russian Academy of Sciences, 14<sup>st</sup> line, 39, Vasilievsky Island, St. Petersburg, 199178, Russia.

E-mail: [nazhukova@mail.ru](mailto:nazhukova@mail.ru)

<https://orcid.org/0000-0001-5877-4461>

Статья поступила в редакцию 17.09.2024; принята к публикации после рецензирования 22.10.2024; опубликована онлайн 25.12.2024.

Submitted 17.09.2024; accepted 22.10.2024; published online 25.12.2024.

---